|  |  |
| --- | --- |
| Mini Project  **OXY CHAT V1.0.01** | **Abstract**  Oxy Chat is a LAN based Instant Messaging Application. To communicate with another system in the Local Area Network this project is sufficient and very useful.  Instant messengers have become a popular medium for providing awareness of others and supporting casual interaction. To smoothly move into and out of interaction, it is necessary to afford an awareness of who is around. We have developed a peripheral visualization for an instant messenger designed to utilize people’s natural cognitive abilities.  Student Name: Yakubu Lute  INDEX NUMBER: 2569914  SUPERVISER: Mr. PANFORD  MINI PROJECT IN JAVA 2017 |

**OXY CHAT**

**A Mini Project**

Submitted to the Faculty of Science of

**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI**

Department Of

**Computer Science**

By

**Mr. Yakubu Lute**

**(2569914)**

**Supervisors: Mr. Panford**

**ABSTRACT**

Oxy Chat is a LAN based Instant Messaging Application. To communicate with another system in the Local Area Network this project is sufficient and very useful.

Instant messengers have become a popular medium for providing awareness of others and supporting casual interaction. To smoothly move into and out of interaction, it is necessary to afford an awareness of who is around. We have developed a peripheral visualization for an instant messenger designed to utilize people’s natural cognitive abilities.

The main objective of this project is to facilitate communication with another person within the network without using the INTERNET. Users can sign in with their User id and Password. They can chat with their friends, if they are currently online. And also, users can chat with their friend in Telugu. This makes the users feel comfort in chatting with their friends in Telugu.

**INDEX**

Topic Page No

**1. Motivation ------------------------------------------------- 1**

**2. Introduction** **------------------------------------------------- 3**

**2.1** Registration **------------------------------------------------- 4**

**2.2** Login **------------------------------------------------- 4**

**2.3** User window **------------------------------------------------- 5**

**2.4** Chat **------------------------------------------------- 5**

**2.5** Forgot password **------------------------------------------------- 5**

**3. Software Requirements Specification -------------------------- 6**

**3.1** Objective --------------------------------------------------------- 7

**3.2** Scope --------------------------------------------------------- 7

**3.3** User interfaces --------------------------------------------------------- 7

**3.4** Functional Requirements ----------------------------------------------------- 9

**3.5** Non-Functional Requirements --------------------------------------- 9

**4. System Design ------------------------------------------------- 10**

**4.1** Class diagram -------------------------------------------------------- 11

**4.2** Sequence diagram for Login ------------------------------------------------ 12

**4.3** Sequence diagram for Forgot password ----------------------------------- 12

**4.4** Sequence diagram for Chat ----------------------------------------------- 13

**4.5** Activity diagram -------------------------------------------------------- 14

**5. Java overview ------------------------------------------------ 15**

5**.1** Introduction **----------------------------------------------------------------- 16**

**5.2** Java is Secure **----------------------------------------------------------------- 16**

**5.3** Java is Platform Independent **----------------------------------------------- 17**

**5.4** JDK **----------------------------------------------------------------- 17**

**6. Sockets** **------------------------------------------------ 18**

**6.1** Introduction **----------------------------------------------------------------- 19**

**6.2** Socket types **----------------------------------------------------------------- 21**

**7. Rice Transliteration scheme ------------------------------------- 22**

**8. Snapshots ----------------------------------------------- 25**

**9. System Testing --------------------------------------------------- 33**

**9.1** Introduction **---------------------------------------------------------------- 34**

**9.2** Software Testing Techniques **---------------------------------------------- 34**

**9.3** Test cases **---------------------------------------------------------------- 36**

**10. Conclusion and Future Scope ------------------------------- 38**

**11. Bibliography ----------------------------------------------- 40**

# **1.MOTIVATION**

Instant messengers have become a popular medium for providing awareness of others and supporting casual interaction. To smoothly move into and out of interaction, it is necessary to afford an awareness of who is around.

Today we have many instant messengers providing many features to users. But we found that many of the users chat with their friends in their regional languages, making use of English characters.

People belonging to Telugu nativity wish to chat in Telugu with their friends. But they do frame the Telugu words in English. To make them communicate in exact Telugu words, we have developed this *OxyChat*. Our motive through this project is to add the feel of Telugu in chat.

**2. INTRODUCTION**

Oxy Chat is a LAN based instant messaging service. It allows users to chat in Telugu too. Any user can login only at the client system. New user has to register for a user id. Existing users can login using their user id and password protected. Whenever the login is successful, user window appears.

Once he has logon, he has the list of online users. He can chat with them by sending a connect request. If the receiver accepts, then they can chat.

To chat in Telugu, the user has to install Telugu Lipi Editor and Telugu Lipi font.

All the services required by the logon user will be sent to the server for processing, and he gets the response for the requests. The LAN connection provides the communication medium between the client and the server.

All the records regarding the users are stored at the server. The server manages the records of all the users.

2.1 Registration

The very first step for using this application is to get registered. Anyone who wishes to chat has to get a unique id. The user has to click on *Get Your Id*, and the registration form appears. The registration form contains the fields - username, password, confirm password, first name, last name, designation, city, hint question & hint answer. The fields accept all characters, blank spaces, special characters & numbers.

Now he has to fill all the fields in the registration form. And one restriction in registering is that the field in the registration form can have maximum of 20 characters only.

2.2 Login

Once the registration has completed, users can login using their login id and password protected. Users have to enter correct login id and password. The user id is case-insensitive.

2.3 User window

When the login is success, user window appears. The user window has the list of all online users. The user can chat with any number of users who are in online. The user has to select the contact and click *Connect* to send a chat request to that contact. If the contact wishes to signout, he has to click on *Signout.* Then the user is removed from the online users list.

2.4 Chat

If the contact at the other side accepts the chat request, then they can chat. Here a new window opens for both the users to chat. In this chat window, there are certain controls like – BOLD, ITALIC, SIZE, 2 text areas, a SEND button to send the chat message and a CLOSE button to stop chatting with that contact. The message to be sent is typed in lower text area field. When the user wishes to send the chat message, he has to click on *Send*. The messages sent and received will appear in the upper text area.

If the contact wishes to send message in Telugu, he has to open Telugu Lipi Editor and type the message in the editor to get the message in Telugu. The message so obtained in Telugu is to be copied to chat window text area and he can send it. This message appears in exact Telugu to the contact at the other side.

If any of the contact who is chatting to each other wishes to close the chat, then he has to click on *Close.* Then the chat window closes at both the sides with prior intimation to user at the other side.

2.5 Forgot password

In case the registered users forget their password; they can set a new password using the hint question. All they need to do is to answer the hint question correctly which they have given at the time of registration. If the user answers it correctly, they he will be given a chance to set a new password. Once the password has been successfully changed, the new password will be his password.

SOFTWARE

REQUIREMENTS

SPECIFICATION

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Objective

Oxy Chat provides messenger services to all the users with in a LAN network.

3.2 Scope

Our project provides access rights to all the registered users in the LAN. Users with in the LAN can communicate to one another using their login. We connect all the users at the Client systems to a Central Server which contain the records of the registered users. Communication between the clients is made using their system ip address. Client requests are processed at the server and are directed to the receiver.

Users with in the LAN can chat to one another, and also in Telugu.

3.3 User interfaces

Login screen

User id Textbox

Password Textbox

Sign in Button

Get new id Button

Forgot Password Button

Help Button

User window

Contact list Tree structure

Connect Button

Signout Button

Chat window

Send Button

Bold Button

Italic Button

Size Combo box

Message Text Area

Received messages Text Area

Registration window

User name Textbox

Password Textbox

Confirm password Textbox

First Name Textbox

Last Name Textbox

Designation Textbox

City Textbox

Hint Question Combo box

Hint Answer Textbox

Register me Button

Clear Button

Cancel Button

Forgot password window

Login id Textbox

Hint Question Combo box

Hint Answer Textbox

New password Textbox

Confirm Password Textbox

Submit Button

Cancel Button

3.4 Functional Requirements

**1. Registration for new users**

1. Access rights to the registered users.

2. Registration is based on a predefined form.

3. Predefined form has the following fields

i) Username

ii) Password

iii) Confirm password

iv) First Name

v) Last Name

vi) Designation

vii) City

viii) Hint Question

ix) Hint Answer

4. Facility to modify/cancel any field before submitting the form.

5. Facility to cancel the registration.

**2. Chat Requests**

1. Send/receive chat request to/from the contact.

2. Accept/Deny the request.

3.5 Non functional requirements

Reliability

It requires continuous availability of server. Users can connect only if server is ON.

Scalability

The number of users does not affect the scalable issue if the LAN speed is made to cope up with the traffic of service requests.

Transparency

The user is not aware of the kind of authentication taking place at the server. The messages sent by the sender are transparent of the messages received at the receiver.

SYSTEM DESIGN

4.1 Class Diagram:



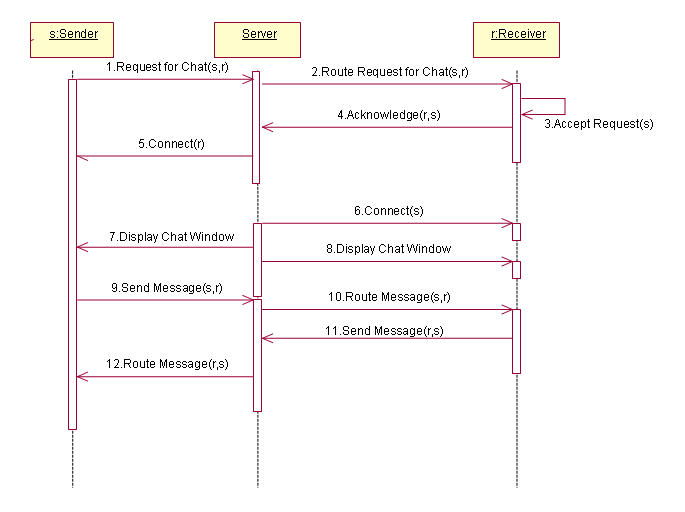
4.2 Sequence Diagram for Login:



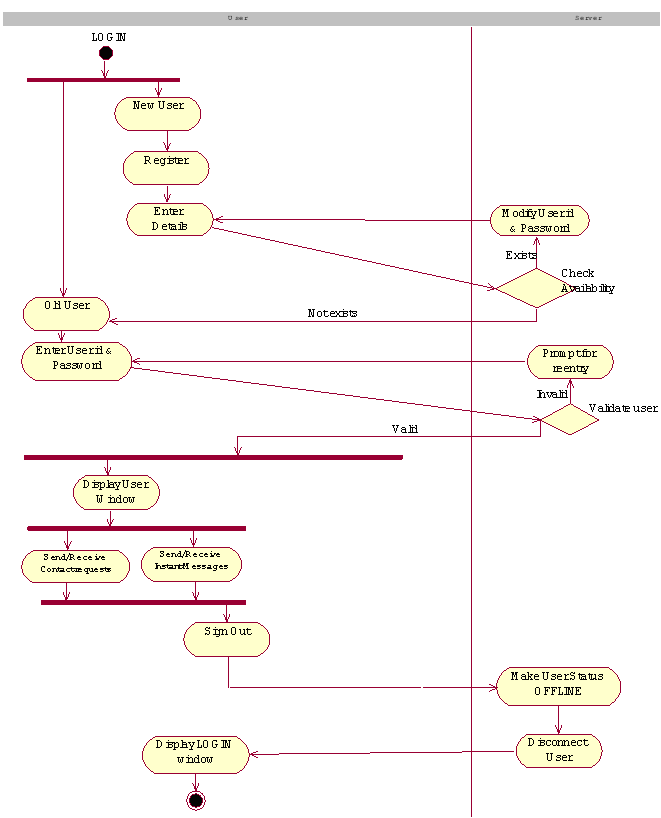
4.3 Sequence Diagram for Forgot Password:



4.4 Sequence Diagram for Chat:



4.5 Activity Diagram:



**5. JAVA OVERVIEW**

# 5.1 Introduction:

Java is loosely based on C++ syntax, and is meant to be Object oriented. It however differs from C++ in many ways. Structure of Java is widely between an interpreted and compiled language. Java programs are compiled by the Java compiler into ‘Byte code’, which are secure and portable across different platforms. The technique of having intermediate code i.e., Byte code solves both the security and the portability problems. Byte code is highly optimized set of instructions designed to be executed by virtual machine that the Java run-time system emulates. That is the Java run time system is an interpreter for Byte code.

These Byte codes are essentially instructions encapsulated in a single byte, to what is known as Java Virtual Machine (JVM), which resides in standard browser, any application can have JVM in-built. JVM, which verifies these Byte codes into machine specific instruction at run time.

# 5.2 Java is Secure:

It is important that a programmer not be able to write subversive code for applications or applets. This is especially true with the Internet being used more and more extensively for services such as electronic commerce and electronic distribution of software and multimedia content.

The way memory is allocated and laid out. In java an object’s location in memory is not determined until the runtime, as opposed to C and C++. As the result, a programmer cannot look at a class definition and figure out how it might be laid out in memory. Also since, Java has no pointers, a programmer cannot forge pointers to memory.

The Java Virtual Machine (JVM) doesn’t trust any incoming code and subjects it to what is called Byte Code Verification. The byte code verifier, part if the virtual machine, checks that

bd14580_ The format of incoming code is correct

bd14580_ Incoming code doesn’t forge pointers.

bd14580_ It doesn’t violate access restrictions.

bd14580_ It accesses objects as what they are

The Java byte code loader, another part of the JVM, checks whether classes loaded during program execution are of local from across a network. Imported classes cannot be substituted for built in classes, and built in classes cannot accidentally reference classes bring in over a network.

The Java Security manager allows user to restrict entrusted Java applets so that they cannot access the local network, local files and other resources.

5.3 Java is Platform Independent:

Platform independence is a way of saying that java is architectural neutral. Java programs do not care on what system they are running on. Most computer software is developed for a specific operating system. Platform independence is the ability of the same program to work on different operating systems. As shown by the use of the applets on the web, a java. Class file of byte code instructions can execute on any platform without any alterations.

5.4 The JDK:

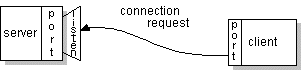
The java Development Kit is a set of command line tools that can be used to create. Java programs. The java version 1.2 includes the following tools; a compiler, an interpreter to run compiled java standalone applications, an applet viewer to run java applets, an archive to create compressed archives and other utilities.

6. SOCKETS

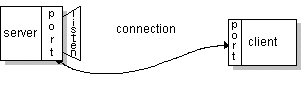
6.1 Introduction

A socket is one end-point of a two-way communication link between two programs running on the network. Socket classes are used to represent the connection between a client program and a server program. The java.net package provides two classes--Socket and ServerSocket--that implement the client side of the connection and the server side of the connection, respectively. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

**On the client-side**: The client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.



If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.



On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server.

The client and server can now communicate by writing to or reading from their sockets.

An endpoint is a combination of an IP address and a port number. Every TCP connection can be uniquely identified by its two endpoints. That way you can have multiple connections between your host and the server.

The java.net package in the Java platform provides a class, Socket, that implements one side of a two-way connection between your Java program and another program on the network. The Socket class sits on top of a platform-dependent implementation, hiding the details of any particular system from your Java program. By using the java.net.Socket class instead of relying on native code, your Java programs can communicate over the network in a platform-independent fashion.

Additionally, java.net includes the ServerSocket class, which implements a socket that servers can use to listen for and accept connections to clients. This lesson shows you how to use the Socket and ServerSocket classes.

### The client server model

Most interprocess communication uses the client server model. These terms refer to the two processes which will be communicating with each other. One of the two processes, the client, connects to the other process, the server, typically to make a request for information. A good analogy is a person who makes a phone call to another person.

Notice that the client needs to know of the existence of and the address of the server, but the server does not need to know the address of (or even the existence of) the client prior to the connection being established. Notice also that once a connection is established, both sides can send and receive information.

The system calls for establishing a connection are somewhat different for the client and the server, but both involve the basic construct of a socket. A socket is one end of an interprocess communication channel. The two processes each establish their own socket.

The steps involved in establishing a socket on the client side are as follows:

1. Create a socket with the socket() system call
2. Connect the socket to the address of the server using the connect() system call
3. Send and receive data. There are a number of ways to do this, but the simplest is to use the read() and write() system calls.

The steps involved in establishing a socket on the server side are as follows:

1. Create a socket with the socket() system call
2. Bind the socket to an address using the bind() system call. For a server socket on the Internet, an address consists of a port number on the host machine.
3. Listen for connections with the listen() system call
4. Accept a connection with the accept() system call. This call typically blocks until a client connects with the server.
5. Send and receive data

6.2 Socket Types:

When a socket is created, the program has to specify the address domain and the socket type. Two processes can communicate with each other only if their sockets are of the same type and in the same domain. There are two widely used address domains, the unix domain, in which two processes which share a common file system communicate, and the Internet domain, in which two processes running on any two hosts on the Internet communicate. Each of these has its own address format.

The address of a socket in the Unix domain is a character string which is basically an entry in the file system. The address of a socket in the Internet domain consists of the Internet address of the host machine (every computer on the Internet has a unique 32 bit address, often referred to as its IP address). In addition, each socket needs a port number on that host. Port numbers are 16 bit unsigned integers. The lower numbers are reserved in Unix for standard services. For example, the port number for the FTP server is 21. It is important that standard services be at the same port on all computers so that clients will know their addresses. However, port numbers above 2000 are generally available.

There are two widely used socket types, stream sockets, and datagram sockets. Stream sockets treat communications as a continuous stream of characters, while datagram sockets have to read entire messages at once. Each uses its own communication protocol. Stream sockets use TCP (Transmission Control Protocol), which is a reliable, stream oriented protocol, and datagram sockets use UDP (Unix Datagram Protocol), which is unreliable and message oriented.

RICE

TRANSLITERATION

SCHEME

**7. Rice Transliteration scheme**

The intention of this project is to provide Telugu language support to chat application. But it is difficult for a novice user to frame the Telugu words directly using the keyboard. In order to make it simpler, we are using a RIT tool “Telugu Lipi Editor” that automatically converts the corresponding English alphabets into Telugu words. RIT lets you type in roman characters and outputs a postscript file which can contain Telugu and English characters. It uses the character # to switch between Telugu and English modes.

For example,

maaku ee rOju #school# lEdu

## Here is the transliteration scheme:

a   
aa A a'   
i   
ea ee ii I i' ia  
u   
uu oo U u' ua   
R   
~l   
~L   
e   
ae E e'   
ai ei   
o   
oa oe O o'   
ou au ow   
  
  
k   
kh kH Kh KH K  
g   
gh gH Gh GH G  
~m   
c ch   
C Ch CH c'   
j z   
jh jH Jh JH J   
~n

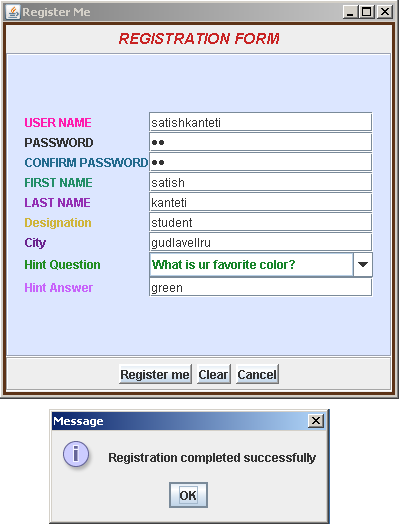
T t'   
Th th'   
D d'   
Dh DH dh' dH'   
N n'   
t   
th   
d   
dh   
n n&   
p   
ph pH Ph PH f P   
b   
bh bH Bh BH B   
m m&   
y   
r   
l   
v w V W   
s' S   
sh sH Sh SH   
s   
h H   
L l'   
ksh x   
~r r''

SNAPSHOTS

Login screen



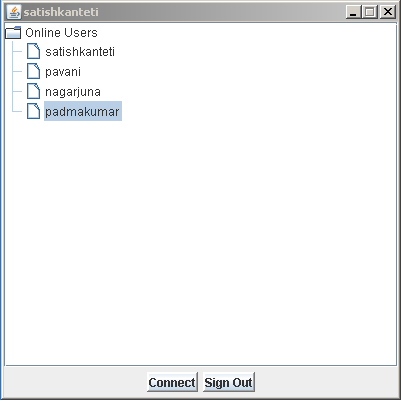
Registration form



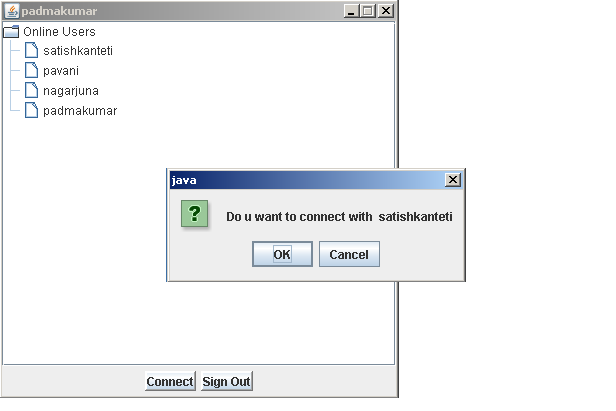
User login –satishkanteti



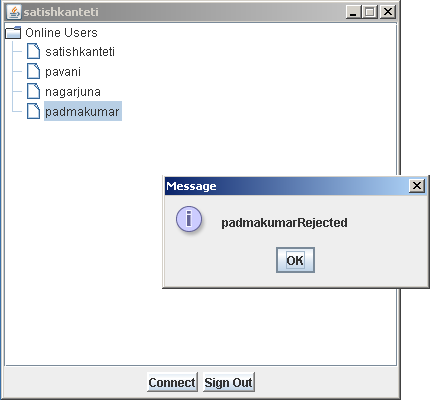
User window



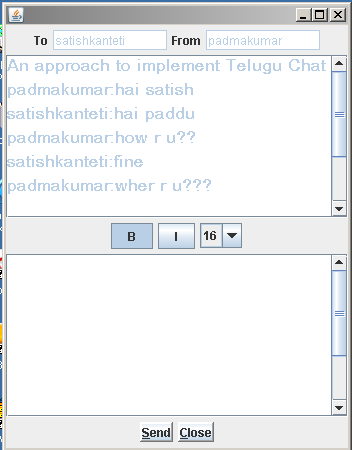
Chat request received from satishkanteti

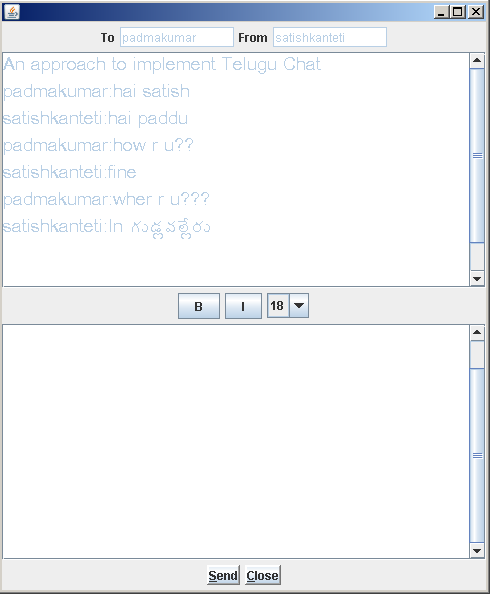


Chat requested rejected by Osman Ali

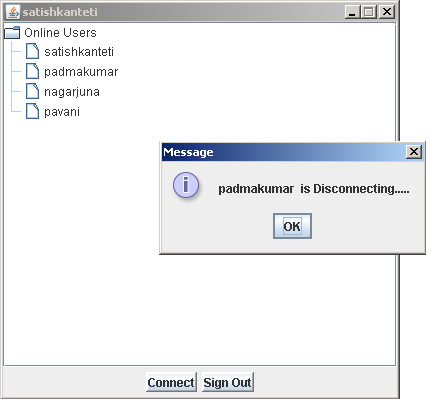


Chat windows when chat request was accepted





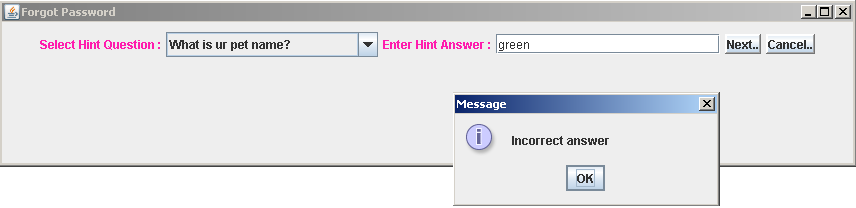
padmakumar is closing the chat with satishkanteti



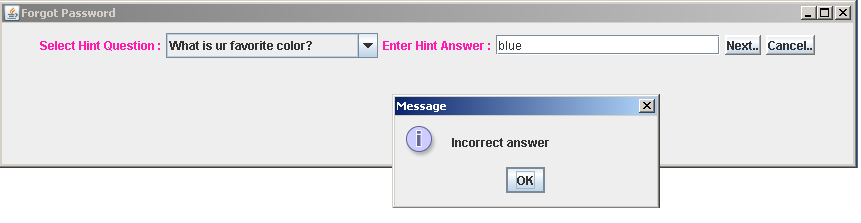
Forgot password – Enter login id



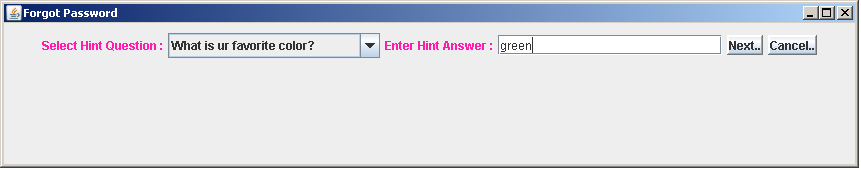
Forgot password – Selected hint question incorrectly and answered correctly



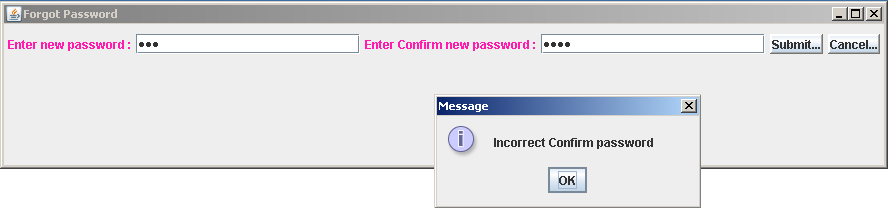
Forgot password – Selected hint question correctly and answered incorrectly



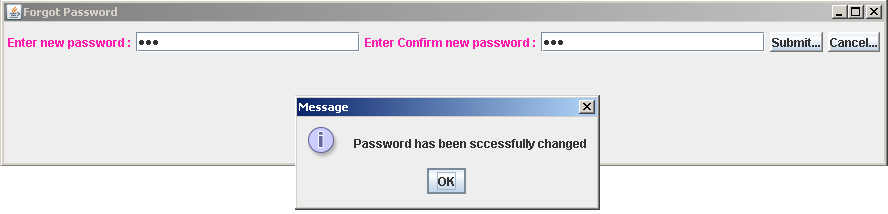
Forgot password – Selected hint question correctly and answered correctly



Forgot password – Incorrect Confirm password



Forgot password –Password successfully changed



SYSTEM TESTING

**9. SYSTEM TESTING**

9.1 Introduction:

The testing phase is an important part of software development. It is the process of finding process and missing operations and also a complete verification to determine whether the objectives are met and the user requirement are satisfied.

9.2 SOFTWARE TESTING TECHNIQUES:

Black Box Testing:

In this testing by knowing the internal operation of a product, tests can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software

White Box Testing:

This testing is also called a Glass Box Testing. In this testing, by knowing the specified function that a product has been designed to perform the test can be conducted that demonstrates each function is fully operation at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Unit Testing:

During the implementation of system module of the system was tested separately to uncover errors with its boundaries*.*

Functionality Testing:

To ensure that each module performs intended functions only those correctness and robustness.

Integration Testing:

The objective of integration testing is to take the unit tested modules and build a program structure that has been defined in the design. We have done a top down integration which is incremental approach which is constructing and testing small segments where errors are easier to isolate and correct. The integration process was performed in three steps. Tests were conducted as each module was integrated.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Name | OxyChat |  |  |  |  |  |
| Version No: | 1 |  |  |  |  |  |
| Test Case Created by | Osman Ali(Comp Sci Std of KNUST) |  |  |  |  |  |
| Reviewed by | Ali Nankpah (Senior programmer at YourIT Solution ,Accra) |  |  |  |  |  |
| Date | 10/05/2017 |  |  |  |  |  |
| **Module Name** | **LOGIN** |  |  |  |  |  |
| Test Case Id | Test Case Name | Steps to execute | Test Data | Expected Value | Actual Value | Result |
|  |  |  |  |  |  |  |
| Oxychat01 | Check Login | Enter User Id | Future | User Id should be accepted | User Id accepted | Pass |
|  |  | Enter Password | \*\* | Password should be displayed in encrypted form | Password displayed in encrypted form |  |
|  |  | Click Login |  | Display user window | User window displayed |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Module Name** | **REGISTRATION** |  |  |  |  |  |
| Test Case Id | Test Case Name | Steps to execute | Test Data | Expected Value | Actual Value | Result |
| Oxychat02 | Check Get Your Id | Click Get Your Id |  | Display Registration window | Registration Window displayed | Pass |
|  |  |  |  |  |  |  |
| Oxychat03 | Check Register me | Enter username | Osman | Username should be accepted | Username accepted | Pass |
|  |  | Enter password | \*\*\*\*\*\*\*\*\*\* | password should be accepted and displayed in encrypted form | password accepted and displayed in encrypted form |  |
|  |  | Enter confirmpassword | \*\*\*\*\*\*\*\*\*\* | confirm password should be accepted and displayed in encrypted form | password accepted and displayed in encrypted form |  |
|  |  | Enter firstname | Ali | First name should be accepted | First name accepted |  |
|  |  | Enter lastname | Osman | Last name should be accepted | Last name accepted |  |
|  |  | Enter Designation | student | Designation should be accepted | Designation accepted |  |
|  |  | Enter city | Kumasi | City should be accepted | City accepted |  |
|  |  | Select Hint Question | What is ur pet name? | Hint Question should be accepted | Hint Question accepted |  |
|  |  | Enter Hint Answer | Nick | Hint Answer should be accepted | Hint Answer accepted |  |
|  |  | Click Register me |  | Display the result | Result displayed |  |
|  |  |  |  |  |  |  |
| Oxychat04 | Check Clear | Click Clear |  | It should clear all the fields | All the fields were cleared | Pass |
|  |  |  |  |  |  |  |
| Oxychat05 | Check Cancel | Click Cancel |  | Current window should be invisible and display login window |  | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Module Name | **CHAT** | | | | | |
| **Test Case Id** | **Test Case Name** | **Steps to execute** | **Test Data** | **Expected Value** | **Actual Value** | **Result** |
| Oxychat06 | Check Connect | Select the user in the contact list |  | Contact node should get selected | Contact node got selected | Pass |
|  |  | Click Connect |  | It should send chat request to the receiver | Receiver received the chat request |  |
| Oxychat07 | Check Signout | Click Signout |  | Current window should be invisible | Current window has been invisible | Pass |
| Oxychat08 | Check Ok | Click OK |  | Display Chat Window | Chat window displayed | Pass |
| Oxychat09 | Check Cancel | Click Cancel |  | Receiver should be intimated as contact rejected | Receiver received the acknowledgement | Pass |
| Oxychat10 | Check Bold | Click Bold |  | Message should be in BOLD | Message appeared in BOLD | Pass |
| Oxychat11 | Check Italic | Click Italic |  | Message should be in ITALIC | Message appeared in ITALIC | Pass |
| Oxychat12 | Check Size | Select size |  | Message should be in selected size | Message displayed in selected size | Pass |
| Oxychat13 | Check Send | Click Send button |  | It should send the message to the receiver | Message was sent to receiver | Pass |
| Oxychat14 | Check Close | Click Close button |  | Chat windows should be invisible to both the users and should acknowledge the receiver | Chat windows got closed with prior intimation to user | Pass |
| Oxychat15 | Check Textarea | Enter message in textarea |  | Message should be displayed in the correponding style | Message got displayed in corresponding style | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module Name** | **FORGOT PASSORD** | | | | | |
| **Test Case Id** | **Test Case Name** | **Steps to execute** | **Test Data** | **Expected Value** | **Actual Value** | **Result** |
| Oxychat16 | Check Submit. | Enter Login id | Youngfutre5050 | Login id should be accepted | Login id accepted | Pass |
|  |  | Click Submit. |  | Display Hint Question window | Displayed Hint Question window |  |
|  |  |  |  |  |  |  |
| Oxychat17 | Check Cancel. | Click Cancel. |  | Current window should be invisible | Current window has been invisible | Pass |
|  |  |  |  |  |  |  |
| Oxychat18 | Check Next.. | Select Hint Question |  | It should get selected | Hint Question got selected | Pass |
|  |  | Enter Hint Answer | nick | Display new password window | Displayed new password window |  |
|  |  |  |  |  |  |  |
| Oxychat19 | Check Cancel.. | Click Cancel.. | Check Cancel.. | Current window should be invisible | Current window has been invisible | Pass |
|  |  |  |  |  |  |  |
| Oxychat20 | Check Submit… | Enter new password | \*\*\*\*\* | password should be accepted and displayed in encrypted form | password accepted and displayed in encrypted form | Pass |
|  |  | Enter Confirm password | \*\*\*\*\* | confirm password should be accepted and displayed in encrypted form | confirm password accepted and displayed in encrypted form |  |
|  |  |  |  |  |  |  |
| Oxychat21 | Check Cancel… | Click Cancel... | Check Cancel… | Current window should be invisible | Current window has been invisible | Pass |

**10. CONCLUSION**

**&**

**FUTURE SCOPE**

* OxyChat does provide online chatting to users within a LAN. This can be extended to work in WAN/MAN/INTERNET.
* This can also be extended to provide chat in many other regional languages.

**11. BIBLIOGRAPHY**

1. Software Engineering - Roger S.Pressman
2. Java Complete Reference - Herbert Schildt
3. Java How to program - Deitel & Deitel
4. Database Management Systems - Ramakrishnan

**WEB-SITES:-**

* [http://java.sun.com](http://java.sun.com/)
* [http://java2s.com](http://java2s.com/)
* <http://javaprogrammersworld.com>
* <http://oracle.org>
* http://telugu-pride-telegu-editor.soft112.com