**Instant Messaging Client**Final Project Report – Team 11

Sharon Cao  
Will Emmanuel  
Minwoo Kim  
David Melnick

Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Story ID | Story | Acceptance Criteria | Sprint | Status |
| 1 | As a user, I would like to be able to login to the google talk IM network | 1) Entering the correct username and password allows user to login to system | 1 | completed |
| 2 | As a user, I would like to be able to see my buddy list | 1) Upon logging into system, correct buddy list for user shows up | 1 | completed |
| 3 | As a user, I would like to be able to add and remove buddies from my buddy list | 1) System allows user to add buddy to list  2) System allows user to delete buddy from list if he/she is already on the list | 2 | completed |
| 4 | As a user, I would like to be able to send and receive chat messages | 1) Chat will begin when user selects buddy  2) Chat will close upon user closing chat window  3) Chat will begin when buddy sends message to user | 1 | completed |
| 5 | As a user, I would like to be able to post my current status | 1) User should be able to add/remove status  2) Buddies of user should be able to see user's updated status | 2 | completed |
| 6 | As a user, I would like to be able to see which buddies are online and offline | 1) Buddy list distinguishes between online and offline buddies | 2 | completed |
| 7 | As a user, I would like for the IM client to have additional features | 1) Chat history | 2 | completed |
| 8 | As a user, I would like to be able to logout after my session | 1) Buddylist and existing chats close when user selects logout | 1 | completed |

Project Description

To log in, the user is required to enter a valid GoogleTalk username—including the domain of the username (ex: “@gmail.com”)—and a valid password. Invalid login information will cause the login window to display “incorrect login” at the bottom. Checking the box “Remember User” will keep the username and password stored for the next log-in. A buddylist appears upon connection. Two separate lists of online and offline buddies are displayed. Double clicking on an online buddy will open up a new chat with him/her. All previous GChat Client messages with this user are displayed in the chat window. The “Status” Button allows the user to display a status. Users are also able to see other buddies’ statuses. To add a new buddy, the user clicks the “+” button on the bottom of the screen. The buddy you are adding must accept your request before you are able to chat with them. To remove a buddy, the user clicks the “-“ button on the bottom of the screen upon selecting a buddy to delete.

Project Testing  
**Story:** 1 - As a user, I would like to be able to login to the google talk IM network  
**Criteria:** Entering the correct username and password allows user to login to system  
**When this was tested:** Throughout sprint 1 and 2  
**Result:** Correct username and password results in a successful login, otherwise login screen displays “Incorrect login”

**Story:** 2 - As a user, I would like to be able to see my buddy list  
**Criteria:** Upon logging into system, correct buddy list for user shows up  
**When this was tested:** Throughout sprint 1 and 2  
**Result:** The correct buddy list displays for the username that is logged in.

**Story:** 3 - As a user, I would like to be able to add and remove buddies from my buddy list  
**Criteria:** System allows user to add buddy to list  
**When this was tested:** Middle of sprint 2  
**Result:** When user clicks the “+” button, a pop-up asking for the email address and name of the buddy the user would like to add. After the user adds a valid email address and clicks “Add Buddy” and buddy accepts the request, buddy appears in the user’s buddy list.

**Story:** 3 - As a user, I would like to be able to see my buddy list  
**Criteria:** System allows user to delete buddy from list if he/she is already on the list  
**When this was tested:** End of Sprint 2  
**Result:** When user enters a buddy name and clicks the “-“ button, the buddy is removed.

**Story:** 4 - As a user, I would like to be able to open a chat window with a buddy and begin a chat  
**Criteria:** Chat will begin when user selects buddy  
**When this was tested:** End of Sprint 1  
**Result:** When the user clicks on an online buddy, a chat window pops up

**Story:** 4 - As a user, I would like to be able to open a chat window with a buddy and begin a chat  
**Criteria:** Chat will close upon user closing chat window  
**When this was tested:** End of Sprint 1  
**Result:** When the user has a chat window up already, user can click the “x” button on the chat to close it.

**Story:** 5 - As a user, I would like to be able to post my current status  
**Criteria:** User should be able to add/remove status **When this was tested:** End of Sprint 2  
**Result:** When the user hits status, user can add status

**Story:** 5 - As a user, I would like to be able to post my current status  
**Criteria:** Buddies of user should be able to see user's updated status **When this was tested:** End of Sprint 2  
**Result:** When user updates status, buddies are able to see that new status.

**Story:** 6 - As a user, I would like to be able to see which buddies are online and offline  
**Criteria:** Buddy list distinguishes between online and offline buddies  
**When this was tested:** End of Sprint 1  
**Result:** Buddy list displays online users and offline users in two different groups. Offline users are grayed out, and online users are available for chat.

**Story:** 7 - As a user, I would like for the IM client to have additional features  
**Criteria:** Chat history is displayed  
**When this was tested:** End of Sprint 2  
**Result:** When a user, clicks on a buddy that they have chatted with previously, their chat history is also displayed.

**Story:** 8 - As a user, I would like to be able to logout after my session  
**Criteria:** Buddy list and existing chats close when user selects logout  
**When this was tested:** End of Sprint 1  
**Result:** Upon clicking the “logout” button, the buddylist and any chat windows are automatically closed.

All team members periodically tested the product during every sprint of the project as well as at the end to ensure that all features were working correctly. A test plan with acceptance criteria was created so that all system requirements would be met. Additionally, a task list was updated regularly to document all product bugs that needed to be fixed. The product was constantly tested on both macs and pcs to ensure that the design was consistent on different types of computers. Every time new code was implemented, the IM client was run again to check that newly added code did not affect code that was already written. These quality assurance methods and processes proved to be helpful in the development of our product because all requirements were met, and the product was finished on time.

Project Design  
**Class: Login**

***Description:*** Upon running the program, a new login screen is displayed that allows for user-input, then sets up a connection provided a valid password/username. The username and password are stored for the next login if the remember box is checked off.

***Important Fields*:** JFrame frame, JtextField login, JPasswordField pwdPass, JCheckBox remember

***Important Methods/Classes:***

main(String [] args): calls run to set up a new login window.

initialize(): creates a new GChat JFrame which allows the user to input information

class SwingAction: upon clicking “login”, sets up a new connection using username and password. Also saves username and password to a file if remember box is selected

**Class: buddyList**

***Description:*** Upon login, sets up a Buddy List that separates online/offline users, and updates information during session. Also defines what occurs when clicking “login”, “+”, and “-“.

***Important Fields:*** XMPPConnection connection, Vector<String> onlineListData, Vector<String> offlineListData, Roster roster

***Important Methods/Classes:***

buddyList (XMPPConnection, String): goes through user’s roster, stores each buddy in offlineListData or onlineListData based on if they are online/offline, calls initialize() at the end, which creates a new buddyList window.

updateBuddyList(): updates the buddylist for each time after login that a buddy changes between online/offline.

class LogoutAction: upon clicking “logout”, a window appears which exits program and ends connection upon confirmation.

class AddFriendAction: upon clicking “+”, the methods defined in the addBuddy class begin

class DeleteFriendAction: upon clicking “-“, removes the currently selected buddy from the buddylist

**Class: addBuddy**

***Description:*** A new window appears upon clicking “+” which allows the user to input an email and a name for the buddy. This new buddy is added to the user’s roster once the new buddy confirms.

***Important Fields:*** JFrame frame, JTextField textfield, Roster roster, JTextField name

***Important Methods/Classes:***

addBuddy(Roster, XMPPConnection): calls initialize() which sets up a new window which allows for user input and updates the user’s roster and connection

class BuddyAdd: upon clicking “Add Buddy”, creates a new buddy to add to the user’s roster based on the inputted email and name.

**Class: chatWindow**

***Description:*** Allows the user to select a buddy and begin a chat. All sent and received messages are displayed in addition to message history.

***Important Fields:*** RosterEntry chatBuddy, ChatManager chatManager, JTextField message, Chat chat, File file

***Important Methods/Classes:***

chatWindow(RosterEntry, chatManager): creates a new chat, accesses previous messages between users, calls initialize()

initialize() : creates a new window upon clicking a buddy and calls sendMessage(String) upon clicking “Send” in the chat window

sendMessage(String): sends a message from the user to the buddy, displays it, and stores the information on the file of message history

**Class: changeStatus**

***Description:***  Allows user to create a status upon clicking “Status”

***Important Fields:*** JFrame frame, Roster roster, XMPP Connection connection

***Important Methods/Classes:***

changeStatus(Roster, XMPPConnection): calls initialize which sets up a window that allows user to input a status, then updates the roster and connection based on the user’s new presence, which is defined in the BuddyAdd class

class BuddyAdd: defines what occurs upon clicking to update status, updating presense

Project Responsibilities

Sharon (Project Manager) –

* Updated product backlog and task list
* Weekly project reports
* Researched, tested, implemented code for receiving IMs
* Added design elements

Will (Software Architect) –

* Designed IM client
  + Connect to google IM network
  + Buddylist with all buddies
  + Start chat with buddies
  + Chat history
  + Logout
  + Status updates

Minwoo (GUI Designer) –

* Managed user interface
* Implemented various features
  + Add/Remove buddy
  + Status updates
  + Buddylist online/offline groups

David (QA Manager) –

* Ensured that elements of the project work correctly
* Researched code for receiving IMs
* Checked if product met project requirements

Project Retrospective

Creation of the project backlog was especially useful in developing our IM client. Listing out from the very beginning exactly what the team planned to do ensured that the IM client would include all the necessary requirements. By adding stories and the acceptance criteria, it was easy to check if the requirements were being met. Additionally, including specific dates that each task should be finished helped to keep the project on schedule so that it could be completed by the deadline.

The task list for each story was a good way for the team to determine which things still needed to be done, and to add tasks if current features needed to be fixed or additional features should be added to the client. The total hours was difficult to estimate, and as a result wasn’t used much in the planning and completion of our project.