

# System Requirements Specification

## Chatboard

---

Justin Michel, Jeff Willebrand, Felix Yuan

## ABSTRACT

Version 1.0, 9/23/09

## Table of Contents

<b>Revision History .....</b>	<b>2</b>
<b>1. Introduction .....</b>	<b>3</b>
1.1 Purpose .....	3
1.2 Scope and Features .....	3
<b>2. Overall Description .....</b>	<b>3</b>
2.1 Product Perspective.....	3
2.3 Operating environment.....	5
2.4 User Documentation .....	5
2.5 Design/implementation constraints.....	5
2.6 Assumptions and dependencies .....	5
<b>3. System Features .....</b>	<b>5</b>
<b>4. External Interface Requirements .....</b>	<b>10</b>
4.1 User Interfaces.....	10
4.2 Hardware Interfaces.....	10
4.3 Software Interfaces .....	11
4.4 Communications Interfaces .....	11
<b>5. Other Nonfunctional Requirements .....</b>	<b>11</b>
5.1 Performance Requirements.....	11
5.2 Safety Requirements.....	11
5.3 Security Requirements .....	11
5.4 Software Quality Attributes.....	11

## Revision History

Preparer	Date	Reason For Change	Version
Justin Michel Jeff Willebrand Felix Yuan	9/23/09	Initial Draft	Version 1.0 Draft

# 1. Introduction

## 1.1 Purpose

This requirements document is intended to describe preliminary functional and nonfunctional requirements for the first full release of IM Whiteboard client software that was described in the vision and scope document (herein referred to as Chatboard). This document is intended for use by developers on the team, specifically those listed in the preparers headers on the title page (that is also the full extent of the team), to implement and verify the correct functioning of the system. Unless otherwise noted, all requirements are high priority and committed for the final release (version 1.0) though requirements are subject to change.

## 1.2 Scope and Features

Chatboard will allow users of all IM persuasions to communicate to one another, as well as leave messages and pictures on a whiteboard for later access. It will also mimic the chatting service provided by most IM clients but expanded to be multi-protocol. Chats will also allow users a board to draw upon. A more specific vision and scope can be found in the Vision and Scope Document.

# 2. Overall Description

## 2.1 Product Perspective

Chatboard is a chatting client that is meant to enhance a chatting experience between to users by providing new chatting functionality on top of pre-existing chatting services. Chatboard is therefore part of a larger system in that it will provide interaction between other chatting clients and itself. The final goal is to make Chatboard's regular and new features enticing enough so that Chatboard can serve as a replacement to current multi-protocol chatting clients without any feature loss.

## 2.2 User Classes and characteristics

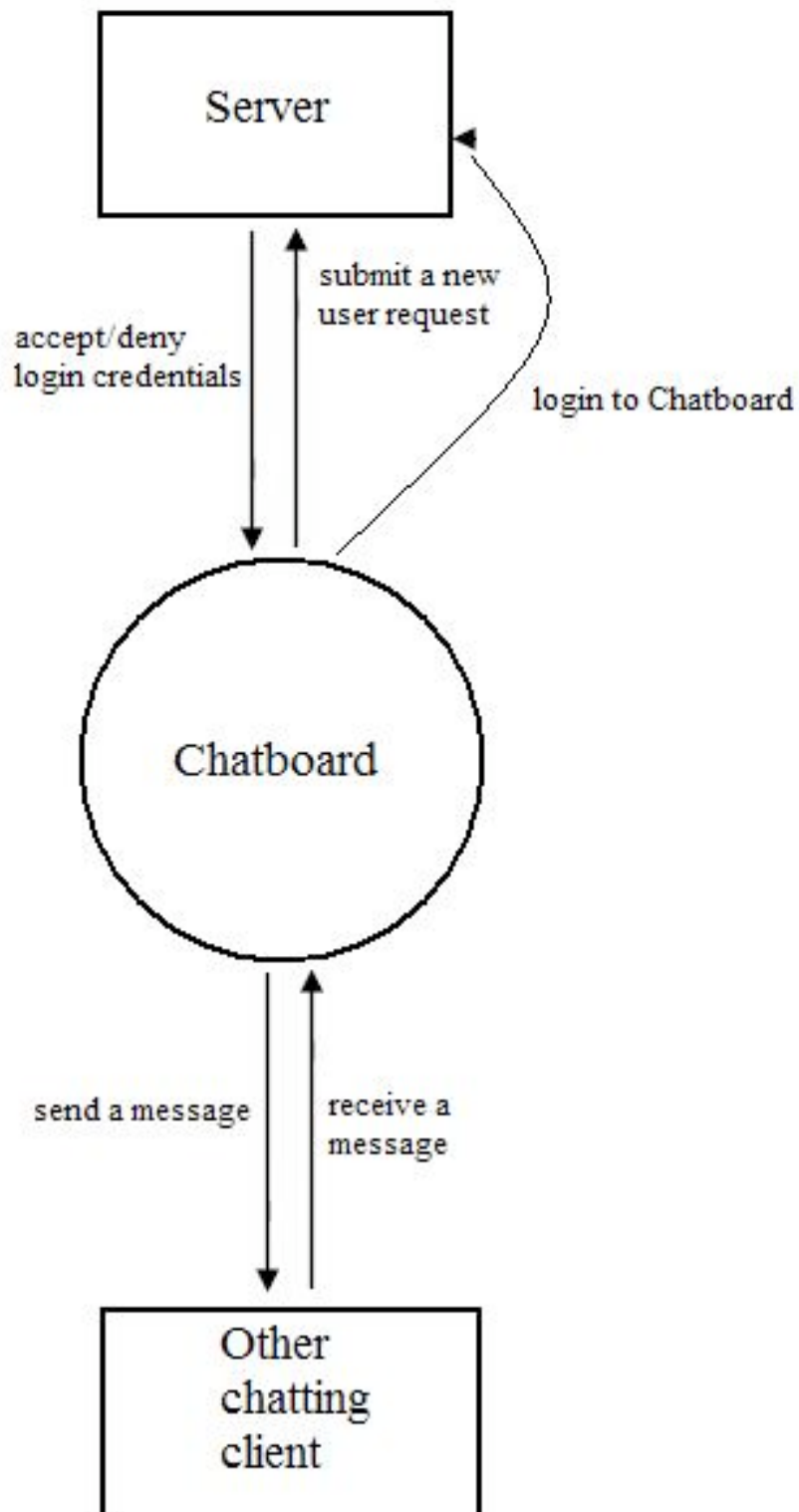


Figure 1: Context Diagram for Chatboard

#### Outside Chatting Client

An outside chatting client is any sort of instant-messaging system that supports one of the protocols used by Chatboard but is not Chatboard. Chatboard will need to be able to communicate seamlessly over its supported protocols.

#### Server

The server will be used mainly for new user registration and login purposes. It will contain a database for user login/registration purposes, and support webservice requests from Chatboard software.

### 2.3 Operating environment

OE-1: Chatboard shall operate on all modern operating systems that support Java.

### 2.4 User Documentation

UD-1: The software will provide a Help section accessible via the main menu which will provide descriptions and instructions of how to use all the major features of Chatboard.

### 2.5 Design/implementation constraints

CO-1: Chatboard shall be written in Java.

CO-2: Server data requests shall occur using the XML-RPC standard.

### 2.6 Assumptions and dependencies

AS-1: A Chatboard user has access to the Internet when attempting to use any chat-related services

DE-1 The server of Chatboard will depend upon the Internet capabilities of its hosting location.

## 3. System Features

### 3.1 Register or Login

#### 3.1.1 Description and priority

Allows users to register for the service provided by chatboard whether through the client itself or through the website. Chatboard uses a webservice and database to keep track of accounts by users. All passwords are encrypted. Priority = high

#### 3.1.2 Stimulus/Response

Stimulus: User registers for an account on the website or through client

Response: Page queries database to check if username is taken. If it is return error. If it isn't then add to database

Stimulus: User login via client

Response: Client encrypts password and sends the user information via XMLRPC to the webservice where it is decrypted. Sends error message if the login info is bad

Stimulus: User forgets password and clicks the forgot password feature

Response: Client calls password reset method via XMLRPC and an email is sent with a password

Stimulus: User wants to change password

Response: User is directed to new client page that will allow him to enter his old password and then his new password. New password is then sent to webservice.

### 3.1.3 Functional Requirements

Webservices.Register: Registers the user with the website. If the username specified is already taken, then return an error. Otherwise add the user to a database, and encrypt the password.

Webservices.Login: Login an already registered user. If credentials are wrong, inform the user of his or her error and then prompt for relogin.

Webservices.Login.Reset: Reset password via email. Password will need to be changed upon reentry into the software

Webservices.Login.Change: Change Password. Password will be asked to conform to the rules of a strong password

## 3.2 Add, Delete, Block operations on account

### 3.2.1 Description and Priority

The add, delete, and block operations are all major operations that are very similar regardless of what category the task belongs. The operations are all done to the main account. One can add and delete an IM service account, which will be done through webservice. One can also add, delete, and block friends, though this will be done for each individual IM account rather than through the master account. Priority = high.

### 3.2.2 Stimulus/Response:

Stimulus: User adds an account.

Response: First the client checks the credentials of the account by attempting to

login to it. If it fails then prompt user for reentry. If it doesn't fail then call the webservice and add the account in a separate table.

- Stimulus: User deletes an account from his account  
Response: Client hits the webservice and the webservice will remove the IM account from the user's Chatboard account
- Stimulus: User turns off an account that he has added  
Response: The IM account that was "turned off" will be sent a sign off message and those friends will no longer show up on the friends list until the user turns the account back on
- Stimulus: User turns on account he has added  
Response: Saved user credentials will be pulled from the computer and the user's IM account will be set a signon message. Friends associated with that account will be added to the friend list
- Stimulus: User adds a friend to an IM account  
Response: Client calls IM account's webservice (AIM, MSN, etc.) and requests an addition. Friend list refreshed
- Stimulus: User deletes a friend from an IM account  
Response: Client calls IM account's webservice and requests a deletion. Friend list refreshed.
- Stimulus: User blocks a friend from an IM account  
Response: Client calls IM account's webservice and requests a block. Friend list refreshed
- Stimulus: User wants to add a folder of a specific category to an IM account  
Response: Client calls IM account's webservice and requests addition of a folder. Friend list refreshed
- Stimulus: User wants to delete a folder of a specific category to an IM account  
Response: Client asks user if he wants to delete all friends as well. If not, then friends are placed in a default folder in the friends list. Client sends delete message to IM account's webservice and requests deletion of folder (and friends if folder deleted)
- Stimulus: User adds friend to folder  
Response: Tag friend with folder name (delete former folder name if any) behind the scenes and update friend list.

### 3.2.3 Functional Requirements:

account.addAccount:	Checks to see if account exists by logging into service. If it does, then hit Chatboard webservice and add account to database. Adds friends to friend list
account.deleteAccount:	Deletes account from user's Chatboard account and removes all friends from friend list
account.accountOff:	Sends that IM account a sign off message and remove friends from friend list
account.accountOn:	Finds credentials stored on HD and sign onto that IM account. Friends are added to friends list.
account.IMAccount.addF:	Adds a friend to the IM account in question by hitting its webservice. Update friend list
account.IMAccount.delF:	Deletes a friend from the IM account in question by hitting its webservice. Update friend list.
account.IMAccount.bloF:	Blocks a friend from the IM account in question by hitting its webservice. Update friend list.
account.IMAccount.addFol:	Add a folder to a specific IM account by hitting its webservice. Update friend list
account.IMAccount.delFol:	Add a folder to a specific IM account by hitting its webservice. Ask if they would like to remove all friends in folder. Update friend list and possibly hit the webservice to delete all friends in folder.
account.IMAccount.movF:	Move Friend to folder by tagging friend with folder name. Remove old foldernames

### 3.3 IM and Chatting

#### 3.3.1 Description and Priority

The IM functionality serves as one of the two main backbones of Chatboard. Very straightforward in that it is like having a conversation except online. It has already been replicated many times by other IM client software. Priority = high.

#### 3.3.2 Stimulus/Response

Stimulus:	User double clicks on a friend he wishes to talk with
Response:	IM window opens up if no IM window is open or "no tabs" is clicked in preferences. Otherwise tab opens in existing window. IM is from the account on which the friend exists
Stimulus:	User types a message and clicks send
Response:	Client checks to see if the message is too long to be sent or the friend is no longer online. If it is then return error. Otherwise send the message and print message inside IM window
Stimulus:	User receives a message from a friend



Response:	If the friend's window is already open then simply print the message and keep focus on the last message. If the window is not open, add new tab or new window to user's screen along with message
Stimulus:	User enters chatroom or accepts invitation to chat
Response:	New chatroom tab or window opens with a panel on the side that allows the user to see all members of the chatroom
Stimulus:	User types a message in chat.
Response:	Message is sent to all chatroom members and message is printed on screen.
Stimulus:	User receives a message in chat
Response:	Message is printed on screen.
Stimulus:	Quit Chatboard application
Response:	Close all windows and tabs and sign off all accounts.

### 3.3.3 Functional Requirements:

IM.talk:	Opens a new IM window or tab depending on the preferences with the friend that was double clicked on
IM.sendMsg:	Client checks to see if friend is online. If the friend is online then proceed to check message length. If message length is too long to be sent, then return error. Else send message to account.
IM.receiveMsg:	Client receives a message and prints it out to the screen.
IM.beginChat:	Client opens a chat window if user accepts a chat invitation or joins a chatroom.
IM.chat.sendMsg:	Client sends a message to all members of the chatroom after the message length is checked.
IM.chat.receiveMsg:	Client receives a message in chat and decodes it before printing it out.

## 3.4 Whiteboard

### 3.4.1 Description and Priority

The killer feature of Chatboard that makes it more usable than other IM clients. It allows a pane that users can draw on that mimic the whiteboard in two situations, ones that are especially familiar to students. First the whiteboard on the dorm room door, where others can leave messages for the student to find. The second is the whiteboard in a study room where students go to do hw. Priority = high

### 3.4.2 Stimulus/Response

Stimulus:	User opens whiteboard pane for an online or offline friend
-----------	--

Response: A new IM window or tab opens if not already talking to said friend, and a pane slides out from the side that allows for free form drawing

Stimulus: User's whiteboard is updated

Response: If user is not logged in then the whiteboard update is saved on the server. When he logs in he is informed that his whiteboard has been updated and the message is updated

Stimulus: Chatboard only chatroom is opened

Response: Whiteboard functionality activated, giving the admin sole control over who can draw on the pane.

Stimulus: Chatroom whiteboard is updated

Response: Client updates the whiteboard in similar fashion

### 3.4.3 Functional Requirements

- whiteboard.writeMsg: Allows user to write a message on another user's whiteboard supposing that they have the Chatboard client. Message is stored in the cloud if the user is not online
- whiteboard.receiveMsg: User's whiteboard is updated by someone else while he is online or off. When user comes online he is told that he has received a new message on his whiteboard and the user's whiteboard is updated as such.
- whiteboard.chat.writeMsg: Same as above
- whiteboard.chat.receiveMsg: Only difference between this and above is that this is updated in real time or a rather close approximation of it
- whiteboard.chat.giveMarker: Admin privileges that allow the creator the chatroom to hand out markers to whoever he pleases. Depending on the permissions of the room, he may or may not be able to take them back

## 4. External Interface Requirements

### 4.1 User Interfaces

- UI-1: The program will display a list of all online "friended" users.
- UI-2: The user shall be able to open a new conversation window for each "friended" user, or to add new conversations to a single tabbed window.
- UI-3: The user should be able to easily navigate between various configuration options and between conversation windows.

### 4.2 Hardware Interfaces

No hardware interfaces have been identified.

### **4.3 Software Interfaces**

No software interfaces have been identified.

### **4.4 Communications Interfaces**

- CI-1: The Chatboard application shall send text and whiteboard drawings to other Chatboard applications.
- CI-2: The Chatboard application shall send a notification to a user's friends list to let their Chatboard application show the user as being signed on.
- CI-3: The Chatboard application should interface with various gateways to communicate with proprietary Instant Messaging standards, such as AIM and MSN.

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

- PE-1: The system shall be verified to accommodate up to 300 users.
- PE-2: The system shall respond to user input within 3 seconds of an action being performed.
- PE-3: A message or whiteboard drawing shall be sent and verified as received within 5 seconds of hitting the "send" button.

### **5.2 Safety Requirements**

No safety requirements have been identified.

### **5.3 Security Requirements**

- SE-1: All user account information shall be encrypted.

### **5.4 Software Quality Attributes**

Availability-1: The Chatboard server that clients will connect to shall be available as long as the Case Western Reserve University network is active in the Village at 115 dormitories, excepting any downtime caused by scheduled maintenance.

Robustness-1: The Chatboard application shall log all user conversations (and possibly whiteboard drawings), so if a user is disconnected, he can look up past conversations on his harddrive.