# SunlabIM

**Specifications Document**

By Tim Neal (tneal)

# 1 Introduction

## 1.1 Problem

In the Sunlab, there is not a user-friendly messaging system that all students can commonly use to communicate with each other. Also, there is no client on the Sunlab machines that allows for chat room capabilities, which would be extremely useful to programming groups.

## 1.2 Overview

While some people may find that zwrite is the easiest way to communicate with friends and class members while working in the Sunlab, most will agree that it is not a very user-friendly program. Zwrite is simple to use, yet it can be quite frustrating at times, such as when trying to delete unwanted characters in a message. It also does not include the ability to have a chat with several people all in one conversation. The goal of this program is to set up a graphical, user-friendly instant messaging system that will replace zwrite as the preferred method of communication in the Sunlab.

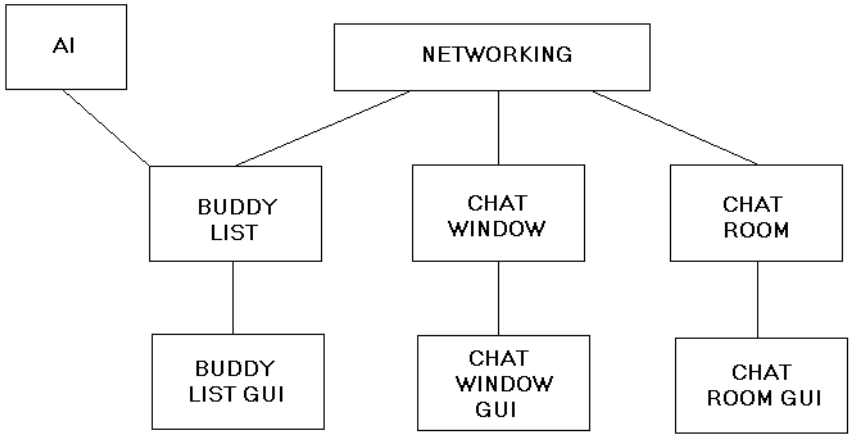
Setting up communication with buddy lists (including add user, drop user, and block user) and a user-friendly graphical interface is a must, and the ability to host chat rooms should be of high priority, as this would be quite useful to groups of programmers in the Sunlab that just could not sit together. Optional features can include buddy info, buddy icons, emoticons, warnings, an AI information buddy, and many other possible features.

## 1.3 Target Users

Target users for this program would include the following:

- Programmers working together in groups in the Sunlab (this is especially useful during busy hours in the Sunlab, when project members may not be able to sit near one another)
- Casual Sunlab users who just want a simple way to communicate with their friends in the lab

# 2 System Model Diagram




Networking

This component is connected to the buddy list, chat window, and chat room because all of those components need to send messages and get information via the network. The network component will need to take commands from one of those components on one computer and respond by sending or getting information from another computer.

Buddy List

The Buddy List is attached to Networking because it will need to get information about which buddies are online via the network. I figured AI was best connected to the buddy list since the AI messenger buddy is reachable via the buddy list.

AI (Artificial Intelligence)

This component represents the possible addition of an AI messenger buddy that can answer questions that anyone might want to ask it. It is best connected to the Buddy List because that is how it is accessed.

Chat Window

The Chat Window needs to send messages to the network and so is connected to Networking.

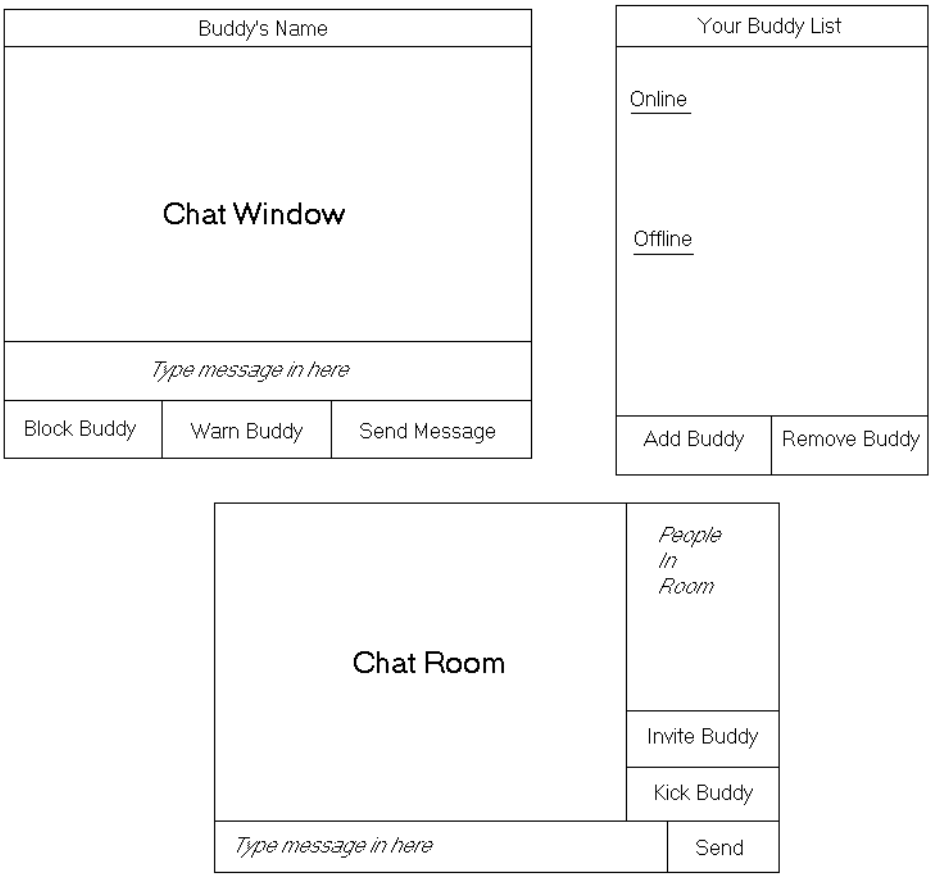
Chat Room

The Chat Room needs to send messages to the network and so is connected to Networking.

GUI

The GUI for each component is connected to its respective component only.

# 3 User Interface Diagram




Chat Window

The picture in the upper left is a sketch of what the chat window in the instant messenger program would look like. The chat text would appear in the middle of the window, while simple buttons appear at the bottom of the window to allow the user to block that buddy, warn that buddy, or send a message. The buddy's name is displayed at the top of the window.

Buddy List

The upper right picture is a sketch of the buddy list. It will have a list of online and offline buddies, and buttons at the bottom to add or remove buddies.

Chat Room

The bottom picture is a sketch of what the chat room will look like. It will have a list of people in the room, and the host will have the invite and kick buddy buttons enabled. The conversation will take place in the big window in the upper left portion of the picture.

# 4 Non-Functional Requirements

## 4.1 Performance

It is important that this program have a quick message transmission time so that users can communicate effectively between one another. The Buddy List should also be updated regularly to note who is online and who is offline.

## 4.2 Testing

Going through the list of requirements and checking to make sure that each item in the list is working correctly is a good way to test this program. At the very least, all the features in the must-have category should be implemented and working correctly in order for the project to be complete.

## 4.3 Reliability

A stress test should be done to measure performance to make sure that this program could handle a large number of users (no larger than the number of nodes in the Sunlab).

## 4.4 Ease of Use

The most important feature of this program would be for it to have a simple GUI with buttons and clear, simple menus that allow access to all of the functions of this program.

# 5 Updated Requirements

### a) Must-Have Features

**- Networking (HIGH PRIORITY)**
- Should be able to host or connect to chat rooms
- Buddy List should be updated with who is online/offline
- Should be able to message anyone who is online

**- Buddy List including the following features: (HIGH PRIORITY)**
- Add buddy - Adds a buddy to the buddy list by typing in name or searching through matching criteria
- Remove buddy - Removes a buddy from buddy list
- Should show the location of each buddy in the Sunlab
- Should be able to group buddies into different groups

- Must have the ability to set different away messages when the user is not available

**- User-friendly GUI (HIGH PRIORITY)**
- Simple logon screen (just logon and password)
- Simple, one-click buttons to send messages, add buddies, remove buddies, and block buddies
- Several two-click menus to allow easy access to any of the implemented features

**- Chat Windows (HIGH PRIORITY)**
- Separate windows for separate conversations
- Block buddy - Ignores all instant messages from a buddy
- Should be allowed to configure window to the user's preference such as:
    - Window size
    - Background color
    - Text font and size
- Either have hot-keys or simple buttons that allow a user to insert emoticons such as the classic smilie faces into a message
- Add the ability to warn an annoying user
- Users with a warn percentage over 100% should be knocked off the instant messaging system for a set period of time

**- Chat room (HIGH PRIORITY)**
- Must have the ability to invite other users into the chat room, and to boot unwanted users
- Should have list of all users in the chat room.

## b) Optional Features

**- AI Instant Messenger Buddy (MEDIUM PRIORITY)**
- Should be able to answer simple questions the user might ask such as
- What the weather is like
- When a project is due
- Any special events coming up

**- Sounds (MEDIUM PRIORITY)**
- Have sounds alerting you to when a buddy signs on or signs off
- Have sounds for sending and receiving instant messages

**- Buddy information (MEDIUM PRIORITY)**
- Each user can be allowed to enter in personal information about him or herself so that other users can read about them.

**- Sending files (MEDIUM PRIORITY)**

- Allow user to send and receive files

**- Buddy Icons (LOW/MEDIUM)**
- Users can have the ability to have a picture of their choice appear when they message a buddy

# 6 Risks

1) This project may not be big enough to split among ten people.
2) If the AI messenger buddy is included in the program, the program may actually become too big.
3) Getting the information for the AI messenger buddy (such as project deadlines and the current weather) may not be possible.