

# Deliverable III

User Manual and Program Information

**Project:**

Ventilate

**Group Members:**

Ryan Porterfield

Jacob Pebworth

Austin Hoppe

Christopher Hines

**Group Name:**

4444-Chat\_Group-5

**Date**

November 18, 201

## Contents

|                                |   |
|--------------------------------|---|
| User Manual.....               | 1 |
| Create Account.....            | 1 |
| Login to Account .....         | 2 |
| Create Chat .....              | 4 |
| Send Message .....             | 4 |
| Compile and Run.....           | 5 |
| Automated Tests.....           | 5 |
| Features and Limitations.....  | 5 |
| Implemented Functionality..... | 6 |
| Missing Functionality .....    | 6 |
| Team Contributions .....       | 6 |
| Christopher Hines .....        | 6 |
| Austin Hoppe .....             | 7 |
| Jacob Pebworth.....            | 7 |
| Ryan Porterfield .....         | 7 |
| Meeting Minutes.....           | 7 |
| Meeting 1.....                 | 8 |
| Meeting 2.....                 | 8 |
| Meeting 3.....                 | 9 |

## User Manual

Ventilate is a chat application which allows users to:

- Create accounts
- Login to accounts
- Create chats
- Send messages

### Create Account

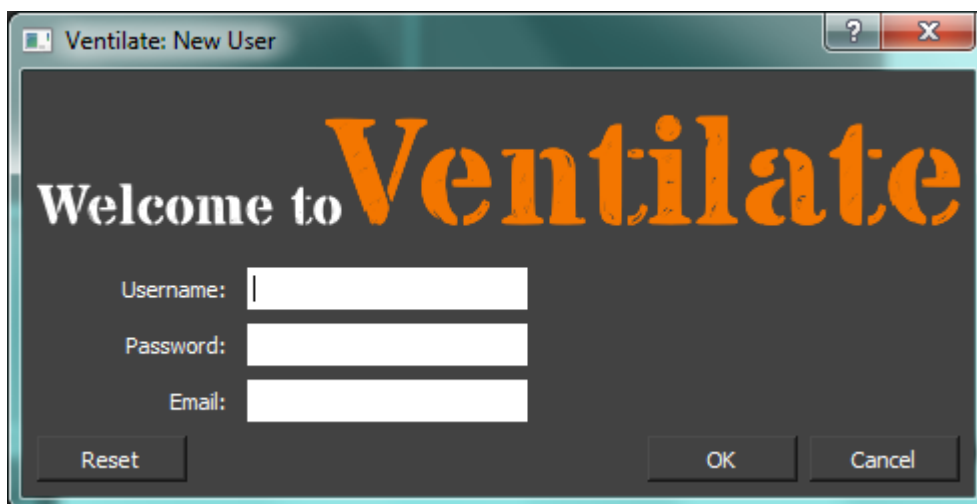
To create a new account, click on the “File” tab in the toolbar and select “Login”. This will open the Login dialog window.



From the Login dialog, click on the “New User” button. This will open the New User dialog window.

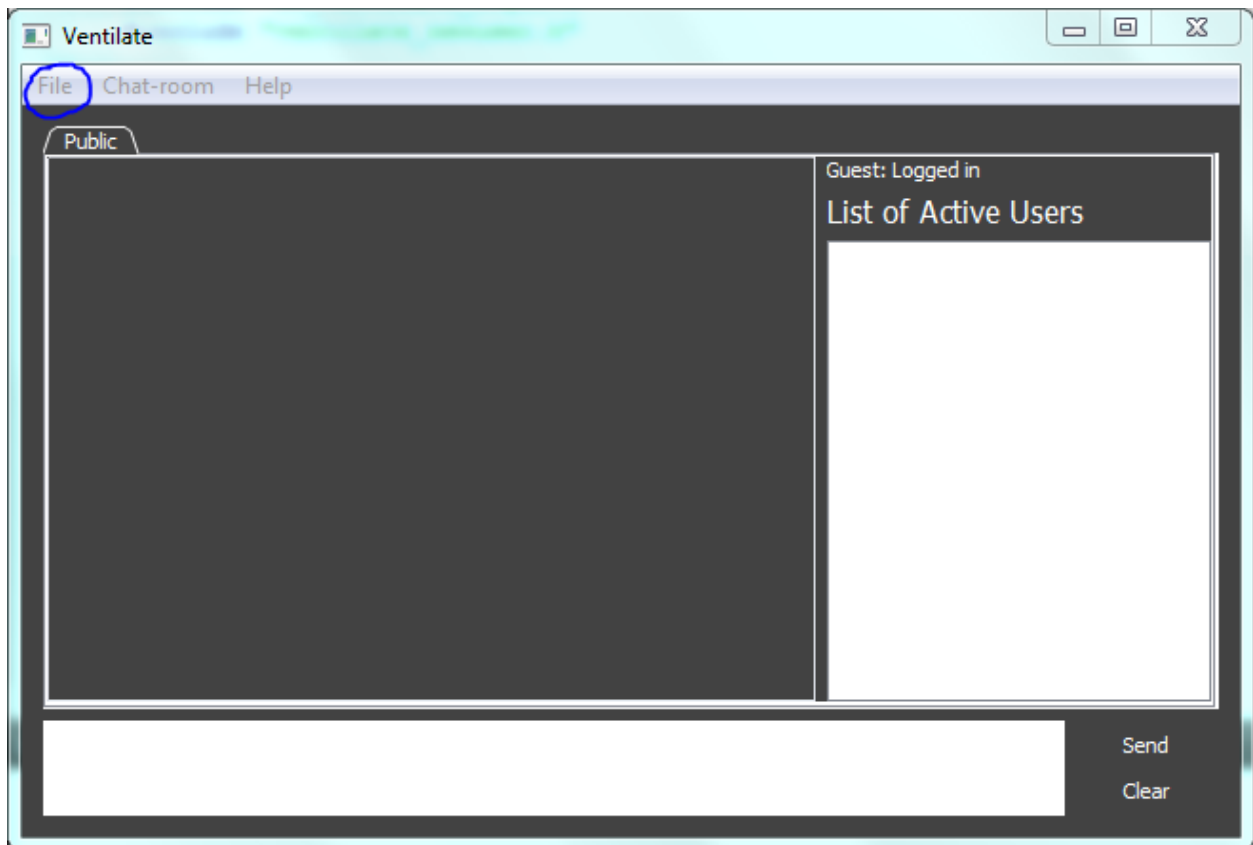


On the New User dialog window type a username, password, and email address into the appropriate text boxes then press “OK”.



### [Login to Account](#)

To login to an account, click on the “File” tab in the toolbar and select “Login”. This will open the Login dialog window.

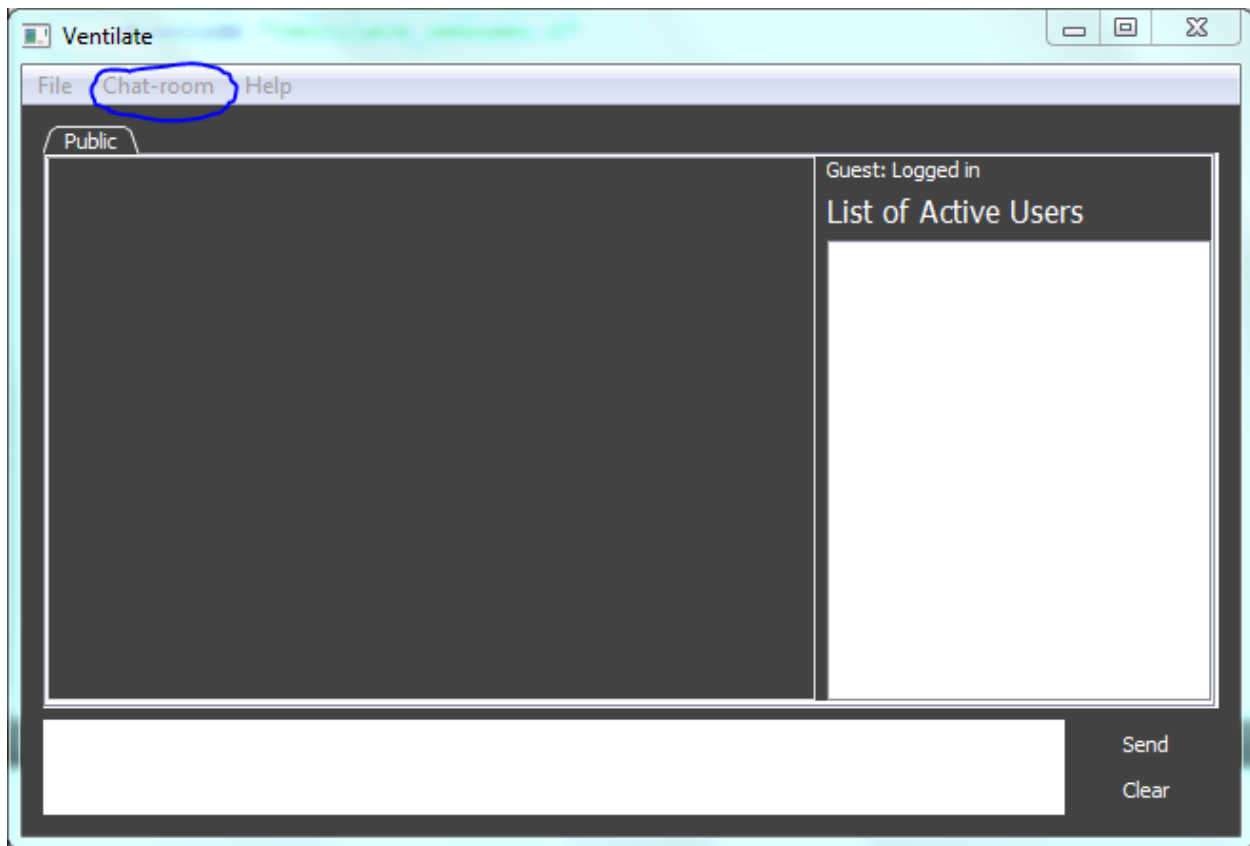


From the Login dialog, type your username and password into the appropriate text boxes then press "OK".



## Create Chat

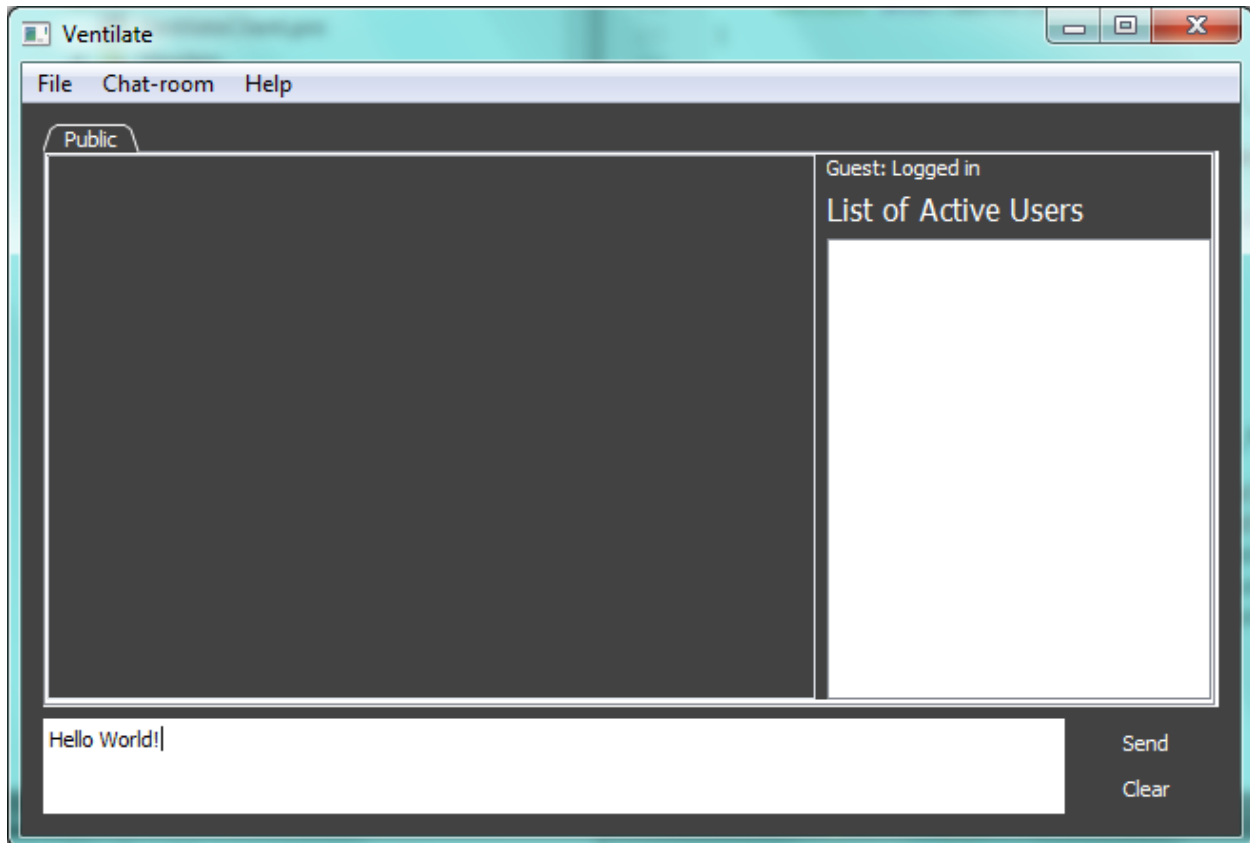
To create a chat, click on the “Chat-room” tab in the toolbar and select “Create Chat-room”. This will open the Create Chat-room dialog window.



From the Create Chat-room dialog, fill in the chat’s name, add any moderators you want, then press “OK”.

## Send Message

To send a message, type the message in the text box at the bottom left of the main window, then press “Send”.



## Compile and Run

Ventilate must be compiled with the open source Community edition of QtCreator. QtCreator Community can be downloaded for free at <https://www.qt.io>.

Open QtCreator, select "Open Project", then browse to the src/Ventilate folder of this project and select the Ventilator.pro.user file.

Ventilate can be compiled and run from the "Build" menu in QtCreator's toolbar.

## Automated Tests

Ventilate has no automated tests. Due to time constraints and problems with communication we were unable to write any tests for Ventilator.

## Features and Limitations

Ventilate was submitted in a non-functioning state. The server is able to respond to requests as well as store and retrieve information from the database. The client can connect to the server, but the client can't make any requests to the server. Additionally while the GUI design is complete, the GUI does not

interface with the client in any functional way. The user has no access to the little functionality that is complete in the client. Many features were cut in an effort to have a functioning program by the submission date.

### Implemented Functionality

- Server handles all requests it receives from clients
- Server can store information in the database
- Server can retrieve information from the database
- Server can remove information from the database
- Server can create new user accounts
- Server can create new chat rooms
- Server can delete user accounts
- Server can delete chat rooms
- Server can store chat messages
- Server pushes new messages to all online clients
- Client can authenticate (login) existing users

### Missing Functionality

- Server can't reset user passwords
- Server can't change user passwords
- Private rooms still unimplemented, all chat rooms are public
- Clients can't connect to each other (no peer-to-peer network)
- Client can't create new user Accounts
- Client can't create new chat rooms
- Client can't send messages to chat rooms
- Client can't delete user accounts
- Client can't delete chat rooms
- Client can't request chat history
- Client can't join chat rooms
- Client can't leave chat rooms
- Client can't request chat room list

## Team Contributions

### Christopher Hines

Chris helped with the Power Point presentation for Deliverable III.



### Austin Hoppe

Austin helped with the product description for Deliverable I and did all the UML diagrams for Deliverable II.

### Jacob Pebworth

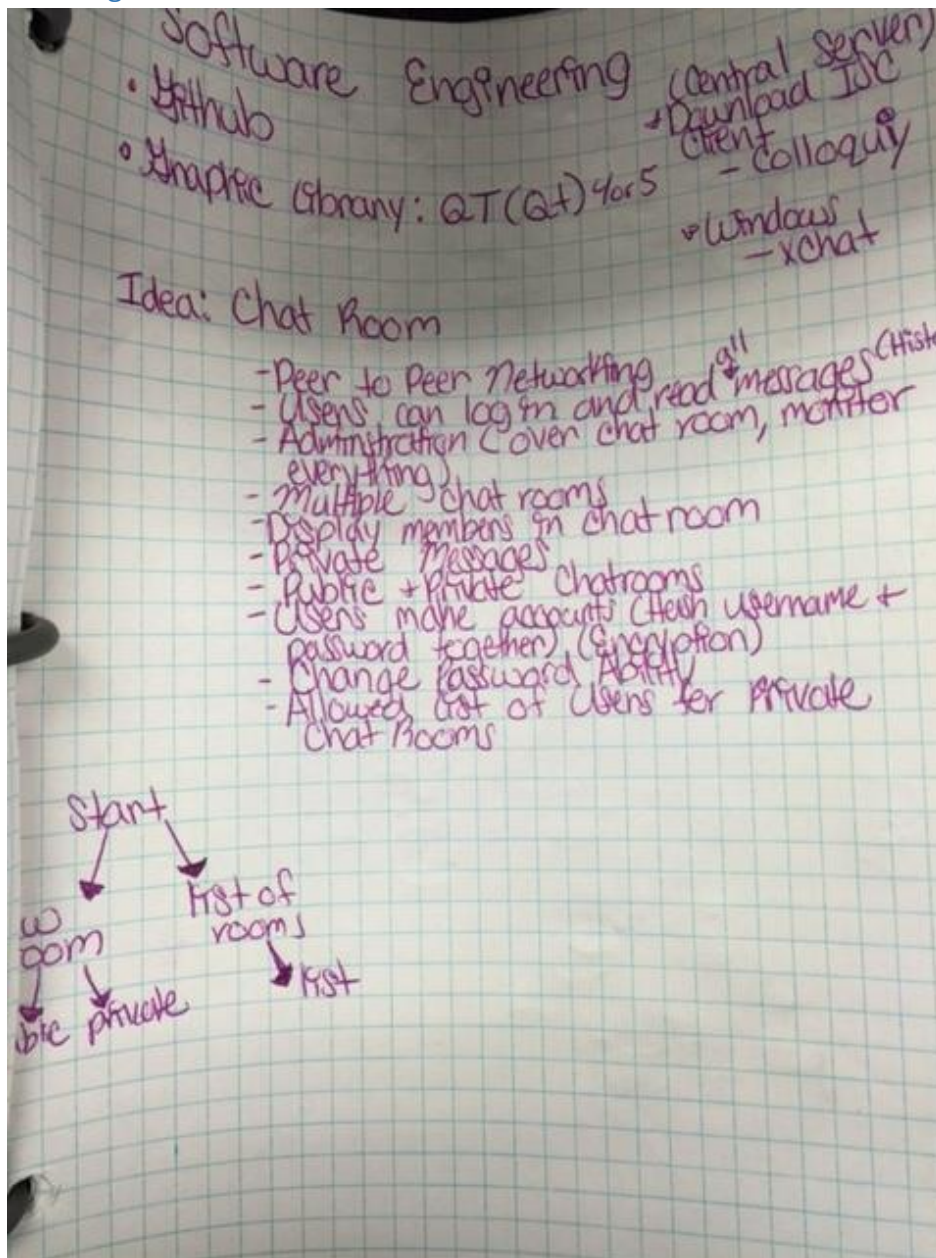
Jacob helped with Deliverable I and did all the UI design.

### Ryan Porterfield

Ryan wrote most of Deliverable I, all of Deliverable II except for the UML diagrams, and all of Deliverable III except for the Power Point Presentation. Ryan also did all of the programming for the application. This includes networking, database storage, data type modeling, functionality testing, and integrating the UI.

## Meeting Minutes

## Meeting 1



## Meeting 2

- Asked about loading list of all public chat rooms piece by piece, in small sections to avoid flooding network, getting a huge list of rooms.
- Asked if all online users would be visible (privacy concerns). Only users in same rooms as you are visible.
- Asked about sending files. Not planned.
- Asked about flagging certain words, watching for "illicit (not vulgar)" speech. Not planned.

- Asked about multiple mods/admins per room. Could implement multiple mods/admins per rooms.
- Asked how the chat history/room information would be stored. Storage is distributed among peers. Each peer stores part of the history. On login, local machine is checked for history before getting missing history.
- Thorough debate about persistency of chatrooms.

### Meeting 3

Friday, 02 October 2015

- Discussed GUI layout and design
- Discussed UML diagrams and project layout
- Discussed possibility of resetting user password via SMS