Software Requirements Specification

For

Ventilate

Version 1.0

Prepared by Ryan Porterfield

4444 Chat Group 5

2015-10-02

# Introduction

## Purpose

This document specifies all functional and non-functional requirements for Ventilate version 1.0.0. Ventilate is a peer-to-peer chat application similar to IRC but with the addition of private rooms. Ventilate communicates directly with other peers on the network, but uses a centralized server simply to store a list of peers which are currently online so that peers can find each other. This document defines functionality for both the client and the server application.

## Document Conventions

There are some key terms used in this document that need to be defined.

**Client** – A client is a user who is running Ventilate on their local machine to talk to other users. Typically client is used to denote a user who is performing an action.

**Server** – The server is the central authority of Ventilate which keeps track of clients who are currently connected to the network so that new clients can be added to the network.

**Peer** – A peer is a client connected to the network who is not performing an action. In this document we will refer to a client getting information from, or sending information to peers. Peers are also clients on the network, but not the client completing the action being described.

**DHS** – A DHS is a distributed hash table which is a method to quickly store and retrieve information in a peer-to-peer network without every peer having to store all the information. See <https://en.wikipedia.org/wiki/Distributed_hash_table> for more information.

## Intended Audience and Reading Suggestions

## Project Scope

Ventilate is a peer-to-peer chat application. Ventilate uses chat rooms similar to IRC, but adds additional functionality. Ventilate uses distributed hash tables to store user accounts including email addresses, usernames, and passwords as well as chat history. Keeping information off a central server reduces the risk of getting hacked and losing all customer information. It also reduces the load on a central server during peak usage hours, instead distributing the load among all the online clients. The chat history not being in a central place also means that even if served a court order, the owners of Ventilate can’t turn over chat history, so users privacy is nominally protected. This doesn’t stop users from joining public rooms and reading back through all the stored history, up to a week, which Ventilate keeps as a feature for its users.

## References

https://en.wikipedia.org/wiki/Distributed\_hash\_table

# Description

## Product Perspective

Ventilate is a new standalone chat application. This document defines the specification for the first production release of Ventilate.

## Product Features

Ventilate allows users to create and delete accounts which have unique user names and are password protected. Ventilate allows users to change or reset passwords for their accounts. Ventilate also allows users to join chat rooms, create chat rooms, and leave chat rooms.

## User Classes

Ventilate will potentially be used by many different types of users. Some classes of users who might use Ventilate are less technically skilled users who just want a simple, easy to use, well designed chat client that they can communicate with their friends on. Other groups who might use Ventilate are users who want the privacy benefits of a peer-to-peer decentralized client. The last class of people who might use Ventilate are technically skilled users who want the performance benefits of a chat application which is well designed and can continue to perform well even under heavy stress.

## Operating Environment

Ventilate will run on a wide range of hardware and operating systems. Ventilate will run on Windows 7, 8, and 10, OS X 11.10 and 11.11, as well as most versions of Linux. Ventilate will also run on x86 and amd64 processors. Ventilate should not conflict with any major software that users may have installed.

## Design and Implementation Constraints

The biggest constraint in designing and implementing Ventilate is time. Ventilate is a senior level group project for a semester long class. This means the developers have only 16 weeks to write specifications and implement functionality for Ventilate. The other major constraint is programming language. In order to save time and maximize the effectiveness of each group member the developers are restricted to a language that every member already knows, and the only language each developer has in common is C++.

## User Documentation

Ventilate will ship with a simple user manual that describes how to do basic functions so that even users who aren’t comfortable with technology can reliably use Ventilate.

## Assumptions and Dependencies

Ventilate is built on the Qt 5 framework, which is an open source, full featured, cross platform framework for designing applications. The Qt libraries simplify network communication and provide a graphics library for designing easy to use Graphical User Interfaces.

# System Features

The program Ventilate should have the following functions:

## Create an account

### Description and Priority

Users of Ventilate should be able to create persistent accounts with unique user names, protected by passwords.

The ability to create an account is **high** priority. All of Ventilate’s features are based around users having unique accounts.

### Stimulus/Response

On the main page of Ventilate users will have the option to log-in or create an account. Creating an account will be triggered by clicking the “Create new account” button on the main page of the application.

### Functional Requirements

In order for users to create an account in Ventilate:

* Both the server and client applications need an account data-type (class) which stores the user’s username, password, email address, and optionally cell phone number and service provider.
* Ventilate needs to store account information securely in a DHS (Distributed Hash Table) shared among the peers (clients).
* Clients need to be able to securely inform peers when a new account is created.
* When an account is created peers need to be able to add the account to the DHS.

## Delete an account

### Description and Priority

Users of Ventilate should be able to permanently delete an account so that the unique user name is no longer associated with them.

The ability to delete an account is **medium** priority. Deleting an account is not critical for any of Venilate’s functions, but is a feature to protect the privacy of our users.

### Stimulus/Response

Any user who has created an account on Ventilate should be able to delete their account at any time. This event will be triggered when a user selects the “Delete account” option from the menu bar.

### Functional Requirements

The following requirements are in addition to the requirements for creating an account.

* Clients need to be able to securely inform peers when an account is deleted.
* When an account is deleted peers need to be able to remove the account from the DHS.

## Login to an existing account

### Description and Priority

Users of Ventilate should be able to login to an existing account if they have already created one.

The ability for users to login to accounts they have created is **high** priority.

### Stimulus/Response

On the main page of Ventilate users will have the option to log-in or create an account. Logging in to an account will be triggered by clicking the “Login” button on the main page of the application after providing a username and password in the appropriate text fields, also found on the main page.

### Functional Requirements

* The clients need to be able to retrieve account information from other peers to validate logins.
* Clients need a secure way to inform other peers when a new user has logged in.

## Logout of an account

### Description and Priority

Users of Ventilate should be able to logout of accounts they have logged in to so that they can login to other accounts, or login to the same account on a different device.

The ability for users to logout of accounts is **high** priority.

### Stimulus/Response

Users of Ventilate who are logged in to the application should be able to logout at any time. The logout event is triggered by selecting the “Logout” option from the menu bar.

### Functional Requirements

* Clients need to be able to securely inform peers that a user has logged out.
* When a user logs out, the network needs to rebalance itself without the user.

## Change account password

### Description and Priority

Users of Ventilate should be able to change the password associated with their account to increase security, if they know their current password.

The ability for users to change their account password is **high** priority.

### Stimulus/Response

At any time, a user who is logged in to an account should be able to change their account password by providing their old password for verification and the new password. The change password event will be triggered by selecting the “Change password” option from the menu bar.

### Functional Requirements

The following requirements are in addition to being able to validate account information as in the Login function, and store account information as in the Create Account function.

* Modify account information stored in the DHT

## Reset account password

### Description and Priority

Users of Ventilate should be able to securely reset their password in the event that they have forgotten their current password.

### Stimulus/Response

### Functional Requirements

## Create a public chat room

### Description and Priority

Users of Ventilate should be able to create a public chat room than any other user of Ventilate can view and join at any time.

### Stimulus/Response

### Functional Requirements

## Create a private chat room

### Description and Priority

Users of Ventilate should be able to create private chat rooms that can only be accessed by people who have permission of the chat owner.

### Stimulus/Response

### Functional Requirements

## Delete a public chat room

### Description and Priority

The owner of a public chat room who no longer wishes for the chat room they created to exist should be able to delete and remove all users from the public chat room.

### Stimulus/Response

### Functional Requirements

## Delete a private chat room

### Description and Priority

The owner of a private chat room who no longer wishes for the chat room they created to exist should be able to delete and remove all users from the public chat room.

### Stimulus/Response

### Functional Requirements

## Join a public chat room

### Description and Priority

Every user of Ventilate should have full, unrestricted access to all public chat rooms, which they can join at will.

### Stimulus/Response

### Functional Requirements

## Add other users to private chat rooms

### Description and Priority

Owners of a private chat room should be able to add users of their choice to the private chat room so groups of people can have private communication.

### Stimulus/Response

### Functional Requirements

## Leave a chat room

### Description and Priority

Any user of Ventilate should be able to leave both public and private chat rooms if they no longer want to be in the room for any reason.

### Stimulus/Response

### Functional Requirements

## Send a message to a chat room

### Description and Priority

Any user who joins a public room or is added to a private room should be able to post messages in the room that every other member of the room can see.

### Stimulus/Response

### Functional Requirements

## View chat history in a chat room

### Description and Priority

Any user who joins a public room or was added to a private room should be able to scroll back and view at least a week of history, or until the chat was created, whichever is more recent.

### Stimulus/Response

### Functional Requirements

# External Interface Requirements

# Other Nonfunctional Requirements