**ACS 560**

**Software Engineering**

**Software Requirements Specification**

**Document**

**Teammembers**

**Jingru Ruan**

**Md**

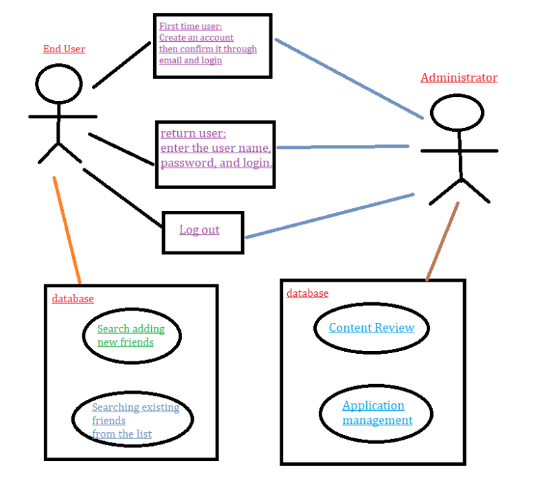
**September 19th, 2017**

**Introduction**

1.1 Purpose

The purpose of this project is developing a chatting interface on computer, providing an alternative chatting experience from phone to the computer.

**Use-case model**



**Functional Requirements**

* Interface: It will provide user interface to user through graphical portal.
* It will provide a comfortable chatting environment, for user to communicate with each other through this platform.
* Versions: We will implement both English version and Chinese version.
* Optional feature: Group chatting.
* Autentication: All users need to login to portal first with their ID and password, to use this application.
* Authorization Level:
* Admin

Admin is super user for this application, which will administrate application with respect to

* User administration (setup and maintain account)
* Maintaining system
* Maintaining account
* Password and identity management
* Monitor application performance
* End Users
* As a user of the client application, I would like to exchange messages with other client application users.
* As a user of the client application, I would like to have a one to one conversation with other client application users.
* As a user of the client application, I would like the server to suggest responses based on my previous chat history.
* As an administrator of the server, I would like to add and manage users in the system.
* As a user of the client application, I would like to have the ability to add users to a contact list.
* As a user of the client application, I would like to have the ability to remove users from my contact list.
* Database manager

We will have one database to store all the chatting information, including conversations between users, pictures and documents

Users can add, retrieve and delete their messages.

**Non-functional Requirements**

* **Environmental:**

Use of client-server model, the client and server using different operation systems and different programming languages(i.e., GO and C#), the server application will use sqlite database to store user information and conversation history.

Client: Windows desktop with C#.net

Server: Linux server with Go language

* **Maintainablity:**

For maintaining database with respect to account management, data content review and it will have down time as minimum 10 minutes in working hours.