

TalkyTalky

Requirements Document

Table of Contents

1	Introduction.....	3
1.1	<i>Purpose and Scope.....</i>	3
1.2	<i>Target Audience</i>	3
1.3	<i>Terms and Definitions.....</i>	4
2	Product Overview	5
2.1	<i>Users and Stakeholders.....</i>	5
2.1.1	Prof. Fei Xie and Bin Lin.....	5
2.2	<i>Use cases.....</i>	5
2.2.1	Register	6
2.2.2	Login	6
2.2.3	Send Message.....	6
2.2.4	Message history	6
2.2.5	Logout	6
3	Functional Requirements	7
3.1	<i>Send messages.....</i>	7
3.1.1	Send messages to a particular user.....	7
3.1.2	Send messages to all users	7
3.1.3	Send messages to all users when a user logs in	7
3.2	<i>Lookup history</i>	7
3.3	<i>Register new account</i>	7
3.4	Login/Logout	8
4	Nonfunctional Requirements	9
4.1	<i>Availability</i>	9
4.1.1	Handle large number of users	9

4.2	<i>Privacy</i>	9
4.3	<i>Testability</i>	9
5	Milestones and Deliverables	10
5.1	<i>Creating a server</i>	10
5.1.1	Using server socket	10
5.2	<i>Creating clients</i>	10
5.2.1	Using client socket	11
5.3	<i>Register</i>	11
5.4	<i>Login/Logout</i>	11
5.5	<i>Send/receive messages</i>	11
5.5.1	Send messages to a particular Talker	11
5.5.2	Send messages to all online Talkers	12
5.5	<i>Check history</i>	12
5.5	<i>Display online Talkers</i>	12

1 Introduction

The introduction of the Requirements document provides an overview of the entire document with purpose, scope, target audience, terms, and functional/nonfunctional requirements for the Chat Application. The aim of this document is to gather and analyze and give an in-depth insight of the complete Chat Application named **TalkyTalky** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs to develop the application. The document details the functionality required for **TalkyTalky**.

1.1 Purpose and Scope

The purpose of this document is to present a detailed description of the user-view and server-view requirements for the Chat Application. User-oriented requirements describe the application from a client's perspective. Server-oriented requirements describe the application from the server's perspective. These requirements describe functional, data, performance, and other important requirements.

The scope of the Chat Application includes its features such as:

- ♦ the server provides connections between users and each user can message to either a specific user or all other users.
- ♦ the new users must register before using the application.
- ♦ the users must log in before communicating with the server or other users.
- ♦ the users can check the history of chat for their own.
- ♦ the server informs all online users when a user login or logout.
- ♦ all users can check who is online or offline.

1.2 Target Audience

The target audience for this Requirements Document includes individual users and stakeholders. Specifically, users shall be the ones who are willing to communicate with other people through Internet in real time.

1.3 Terms and Definitions

Term	Definition
Talker/Use/Clientr	Any person who is using the application and is communicating with others over Internet in real time.
Server	A program that provides connections between users and delivers messages from one user to others.
Stakeholder	Any person with an interest in the project who is not a developer.
Use Case	A list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal.

2 Product Overview

This document contains scope limitations that the current application is facing which is messaging an image to other users or blocking a particular user not to receive a message from them. It further contains a list of users and stakeholders of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercises. It further lists and briefly describes the major features and a brief description of each use case of the application.

2.1 Users and Stakeholders

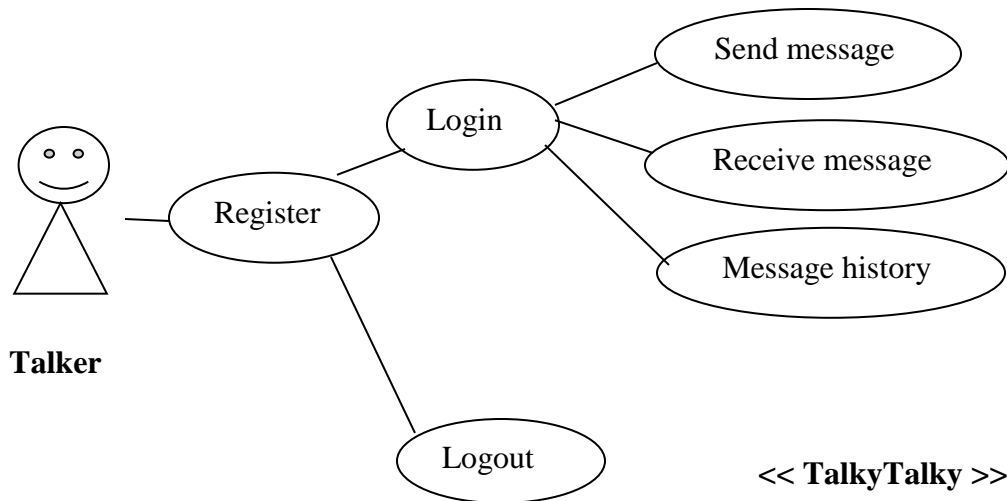
A user is an intended audience for TalkyTalky and is able to communicate with other users via TalkyTalky. A stakeholder is an individual, group, or organization, who may affect a decision, activity, or outcome of the application.

2.1.1 Prof. Fei Xie and Bin Lin

These stakeholders shall give ideas and advices to lead the best outcome of the project.

2.2 Use cases

A use case models an interaction between TalkyTalky and the users of TalkyTalky. A talker represents a user of TalkyTalky and they play a critical role in each use case.



2.2.1 Register

The Talker chooses the option “Create a new account” on the login screen and then the system presents the registration form. The Talker enters the required information to create a new account like ID and passwords and press the “Submit” button. The system validates the data making sure the ID is unique and the password meets predefined complexity rules such as length and mix of characters and numbers.

2.2.2 Login

The Talker chooses the “Login” option on the login screen and the system presents “Login” form. The Talker enters their ID and passwords and presses the “Submit” button, then the system verifies that the provided information is correct. In successful case, the Talker is logged in to the system and is in the main screen. In failure case, the Talker is not logged in to the system and is returned to the login screen.

2.2.3 Send message

The Talker decides who the receiver for the message is, either another Talker or all online Talkers. For 1-1 case, the Talker clicks another online Talker’s ID to send a message to the specific Talker individually. The Talker is directed to the messaging screen with the chosen receiver Talker’s ID. The message from the Talker is only available for the chosen receiver Talker. For 1-many case, the Talker goes to the messaging screen and chooses a receiver to be “All Talkers” to send a message to all online Talkers. Any Talker who is currently online receives a message from the Talker.

2.2.4 Message History

The Talker clicks “history” button and goes to the history screen to see the history of message. The Talker can check who the sender is and what content of message is. The newer message goes to the bottom while the older message goes to the top of the screen.

2.2.5 Logout

The Talker can see and click “Logout” button once they already logged in to the system. The Talker goes back to the login screen in case they want to re-log in to the system.

3 Functional Requirements

This section contains functional requirements that TalkyTalky shall do. The functional requirements are describing the behavior of the system as it relates to the system's functionality. Each functional requirement is described precisely in subsection.

3.1 Send messages

This function is the main feature of TalkyTalky. The system shall send a message for users communicate to each other through TalkyTalky in real time.

3.1.1 Send messages to a particular user

A user shall send a message to a particular user simply by clicking their ID from a list of online users provided by server.

3.1.2 Send messages to all users

A user shall communicate with all online users in real time. Any online user shall join the conversation at any time.

3.1.3 Send messages to all users when a user logs in

The server shall notify all online users by sending a notification message whenever new user logs in. In users-view, every user receives a notification from the server whenever new user logs in.

3.2 Look up history

Every user shall look up the history of their messages in history section. Messages in history shall be organized by time like the older messages shall be shown on the top and the newer messages shall be shown on the bottom of the screen.

3.3 Register new account

A new user registers for TalkyTalky to get an access and start messaging with other people through TalkyTalky. Unique ID is required to register for each user and passwords

has to be long enough and mixed with characters and numbers. Each ID and the matching passwords are stored together in the database.

3.4 Login/Logout

Every user shall login to get started communicating with other users through TalkyTalky in real time once they registered. Their ID and passwords have to be matching with data in database to prove an existing user. Pressing a logout button lets users to log out from TalkyTalky and they shall be no longer receive messages from other users.

4 Nonfunctional Requirements

This section contains nonfunctional requirements that how TalkyTalky shall do. The nonfunctional requirements elaborate a performance characteristic of the system. Each nonfunctional requirement is described precisely in subsection.

4.1 Availability

A major nonfunctional requirement of Chat application is an availability. TalkyTalky shall be available 24/7 with no down time to provide users an online place for communication.

4.1.1 Handle large number of users

The system shall be available to handle large number of users' login and the situation when a user receives large number of messages at one time.

4.2 Privacy

Every user shall be needed a unique ID and complex passwords to get an access to TalkyTalky. All users shall see each other's ID, but passwords shall not be seen to other users.

4.3 Testability

TalkyTalky shall be tested in number of hypothetical test contexts before the system can be cleared for use. Creating number of users, large number of users' simultaneous login, sending lots of messages, keeping track on history, etc. are parts of a list to test for a successful outcome.

5 Milestones and Deliverables

A milestone is a point at which progress on the project may be assessed and a deliverable is an output of the project that can meaningfully be assessed. The following Gantt chart shows the amount of work done and production completed in certain periods of time in relation to the amount planned for the periods.

	4/25	5/9	5/30	6/8
Creating a server				
Creating clients				
Register				
Login/Logout				
Send/receive messages				
Check history				
Display online clients				

5.1 Creating a server

A server is a computer program or a device that provides functionality for other programs or clients. TalkyTalky consists of a chat server and an indefinite number of clients, Talkers. The server connects between Talkers and deliver messages from one Talker to other Talkers in real time.

5.1.1 Using Server Socket

Server socket will be created at the beginning of main in the server class to specify the host address and a port number. Once a server socket is instantiated with a port number, each client can use the port number to connect them with the server. Then, a while loop is used to continuously accept a new client and wait for a client to send it information.

5.2 Creating clients

A client is a computer program that relies on sending a request to a server. Each client will be connected to the server using socket and a port number.

5.2.1 Using client Socket

Unlike the way server socket is used for receiving a connection, client socket is used to connect the client to the server.

5.3 Register

Any new user is supposed to register to get started with TalkyTalky. There is a “Register” button on login screen and user can create their unique ID and complicating passwords. Once the inserted ID and passwords are valid, the user is back to the login screen to start the system.

5.4 Login/Logout

Only Talkers who already registered are able to log in to the system by inputting their own ID and matching passwords. If either ID or passwords is not valid, the Talker receives an error message from the server. The Talker can try to log in again with the correct ID and passwords, and once they successfully log in, the Talker is on the main screen to use TalkyTalky. When they want to finish the system, the Talker can click the “logout” button on the main screen and the button directs the Talker back to the login screen.

5.5 Send/Receive messages

The server accepts connections from the Talkers and delivers all messages from each Talker to other Talkers or delivers messages from one Talkers to a specific Talker if they want. This milestone is the main feature of TalkyTalky that allows Talkers to communicate with each other over Internet in real time.

5.5.1 Send messages to a particular Talker

A list of online Talkers is shown on the side of main screen and each Talker can click a particular Talker’s ID to send a message to them individually. Once the Talker double clicks any highlighted ID from the list, the receiver is the specific Talker that is chosen and any content of message is delivered with the “send” button to the specific Talker. The Talker is not able to send a message to un-highlighted Talker’s ID since currently they are not on the system.

5.5.2 Send messages to all online Talkers

A Talker can message to all online Talkers by clicking “message” button. The receiver is “all” and any content of message is delivered with the “send” button to everyone who is online. For part of that, the server messages to all online Talkers whenever someone logs in/out.

5.6 Check history

The server stores all messages sent from/to each Talker into a data structure and provides the history of messages to Talkers individually. Each Talker is not able to see others’ message history unless they log in with others’ ID and passwords. Each Talker can check their message history by clicking “history” button and the older messages they sent/received are above the newer messages.

5.7 Display online Talkers

Each Talker is able to see a list of online Talkers with highlighted ID on the side of main screen. Once any Talker on the list logs out, their ID is no longer be highlighted on the list from other Talker’s view.