CS3300: Introduction to Software Engineering

Georgia Tech College of Computing

Requirements for Project 1: Watch

Version 0.0: June 2, 2015

Team: Team 4

Table of Contents

1. Introduction

1.1 Overview (Executive Summary)

Real-Time Team Chat is a multi-platform messaging application that allows friends to communicate with each other across all devices. Our plan is to develop a smart phone application that will allow users to sign in and sign out, to send and receive messages, to receive alerts upon receiving a message, and to create group chats. We are also planning to implement a smart watch application to accompany it. The smart watch module will allow users to receive notifications upon receiving a new message, to read recent messages, to “quick reply” to users using templates, and to select which chat or group they would like to view messages from. Our goal for this application is to provide a messaging application that is not restricted by browser or device choice.

1.2 Definitions and Acronyms

UI	User Interface
GUI	Graphic User Interface
IDE	Integrated Development Employment
QA	Quality Assurance

2. User Requirements

2.1 Software Interfaces

Android SDK
Android
Version 5
The application must be created with the Android Software Development Kit. All data in the application, including user messages and group chats, will be displayed and managed by the Android SDK. Furthermore, all user interaction with the application will be handled by the Android SDK.

REST API
API
Version 0.1
This application will be handling all data interactions by making HTTP requests to a REST API service. This includes functionality such as sending and receiving messages, authentication, and creating chat rooms.

2.2 User Interfaces

The interface for our application will be a touch-controlled GUI. The smart phone interface should be sure to include quick access to each planned feature of the messenger and should be scalable in order to appropriately fit any reasonable screen size. The smart watch interface specifically should be highly simplified, as screen space is highly limited on a watch. Though it will also be touch controlled, there may also be room for limited voice control of this application should time allow.

2.3 Product Functions

1. Authentication
    a. Users shall be able to log in to the chat client securely
    b. Users shall be able to log out of the chat client securely

2. Persistence
    a. Users shall be able to see their old messages after logging out and logging back in

3. Messaging
    a. Users shall be able to send messages to other users
    b. Users shall be able to receive messages from other users
    c. Users shall receive notifications upon receipt of a new message
    d. Users shall be able to switch between chats

4. Groups
    a. Users shall be able to create group chats
    b. Users shall be able to leave group chats

5. Quick Reply
    a. Users shall be able to reply to their most recently received message by selecting a response from a list of template responses

2.4 User Characteristics

Our app should be simple enough that anyone with access to a smart phone or smart watch should be able to immediately know how to use it. We should expect our core user demographic to be people from ages 16 to 30, with the possibility that someone as young as 13 could use our application. We should also expect our core demographic to be on-the-go, and want to spend as little time as possible sending and reading messages. As such, the UI should be simple and responsive, placing performance and feature access as our highest priorities.

2.5 Assumptions and Dependencies

- Android SDK will function on both the watch and the mobile phone
- The phone and watch will be capable of interacting
- The Android Wearable SDK will be capable of accomplishing our goals
- The REST API web server connection is required to send and retrieve messages

2.6 Apportioning of Requirements

- Smart watch voice control

3. System Requirements

3.1 Functional Requirements

- **3.1.1** The system shall grant users the ability to log in and out of the system
- **3.1.2** The system shall enable users to message each other
- **3.1.3** The system shall allow users to reply using either device
    - The watch in this case shall be formatted quick replies
- **3.1.4** The system should allow user to receive notifications
- **3.1.5** The user shall be able to store received messages
- **3.1.6** The user shall be able to create and leave groups

3.2 Non-Functional Requirements

3.2.1 Software Quality Attributes

- **3.2.1.1** The system should strive for minimal loss of messages or data
- **3.2.1.2** The system shall persist data for as long as is necessary or until it is no longer capable
- **3.2.1.3** The system shall be usable across any Android compatible phone or watch
- **3.2.1.4** The messages should be secured from outside sources
- **3.2.1.5** Upon completion, the application should be maintained through release on the app store if necessary
- **3.2.1.6** The system shall not crash under normal circumstances

3.2.2 Other Non-functional Requirements

- **3.2.2.1** The application should be deployed on the the Google Play Store
- **3.2.2.2** The application may be sold at minimal costs, if not free
- **3.2.2.3** Our project should allow functions at all possible times and locations