Design Document for Watch

Version 0.0: Jun 4, 2015
Team: Watch / Team 4

Contents

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

1.3 References

1. Android Development (Google): http://developer.android.com/training/index.html
2. Android Wear Development (Google): https://developer.android.com/training/building-wearables.html

Design Considerations

1.4 Assumptions

- 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

1.5 Constraints

The smart watch's limited API and hardware space means that we have to account for a limited user environment on our smart watch. This means in order to allow the users to utilize their watches we have to limit what parts of the application they can use on the watch. Because of these constraints, the watches will only allow for quick replies back to the user's group.

1.6 System Environment

Our chat application will run on the Android mobile platform which acts as an operating system aimed at mobile phones and devices. We will be primarily targeting and operating on handheld mobile phones that range greatly in size and performance, and also Android wear devices specifically watches.

Architectural (High-Level) Design

1.7 Overview

Our system will use the shared server architecture. We will have 3 components: a phone, a smart watch, and a server. The server is responsible for containing and relaying messages to the other 2 components. The phone and smartwatch will both be responsible for sending and receiving messages from the server, while they also have a storage of more recent messages.

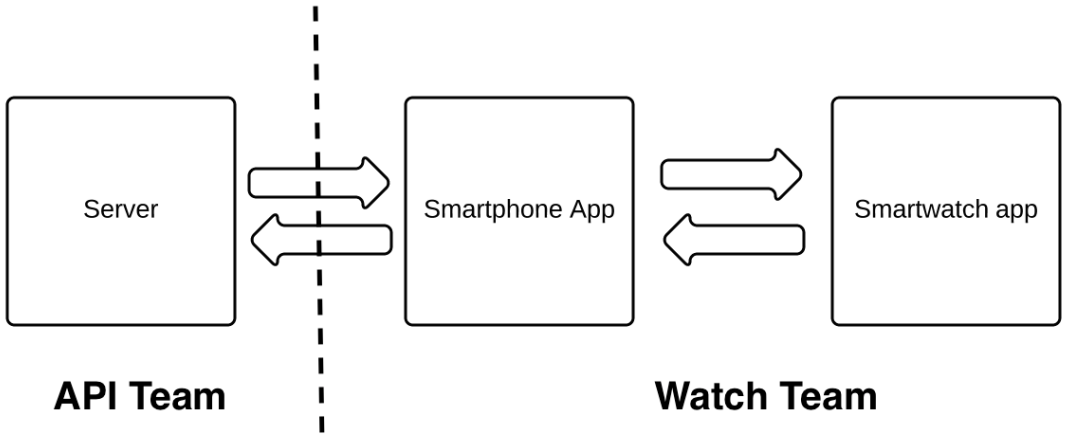
The watch shall be capable of receiving replies from the rest of the group, and shall be able to message other users with quick replies. All of this will be accomplished through correspondence with the phone. The watch will contain a basic UI, a small number of messages, and it's quick replies.

The phone can receive and send messages, and will store messages , and will have a more fleshed out UI.

1.8 Rationale

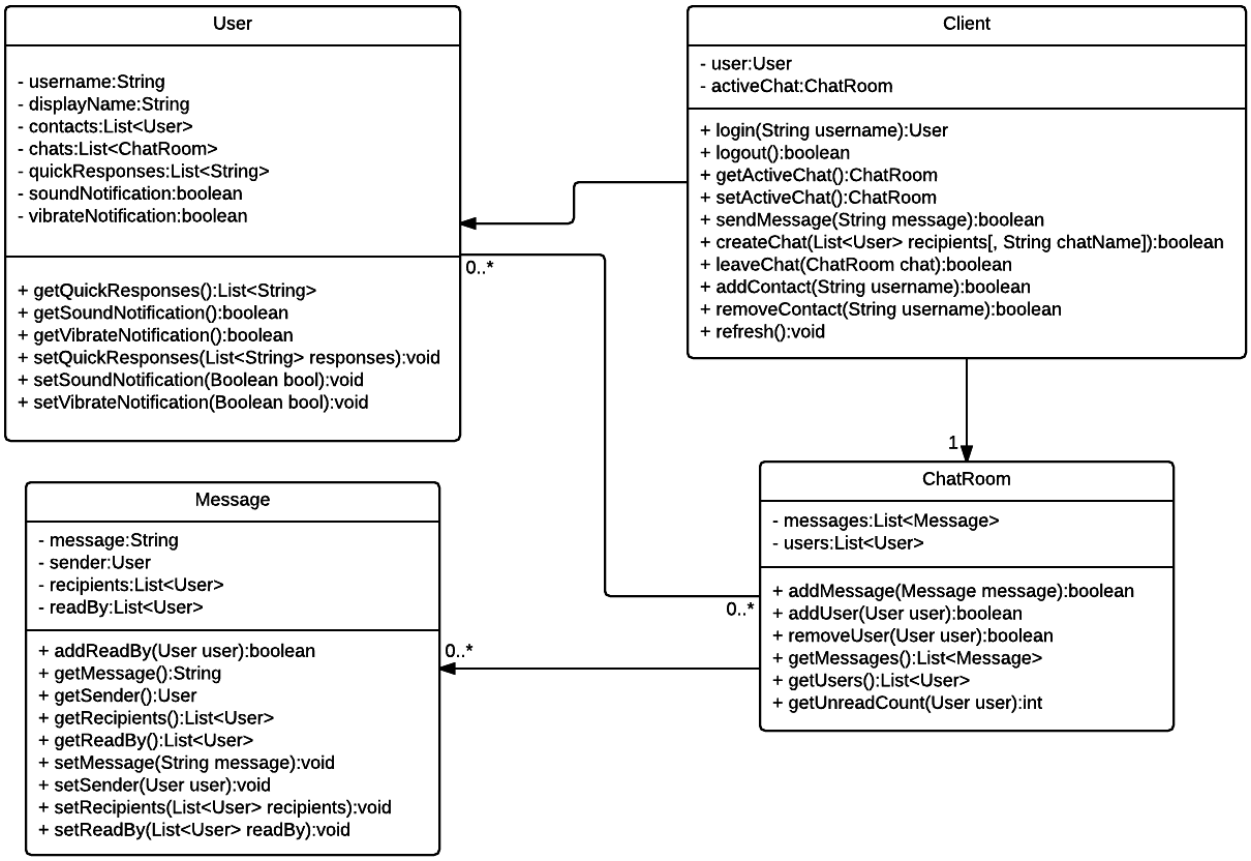
The shared server seems the most appropriate and allows for a simple set up. This keeps the setup simple while maintaining usability and reliability in the event of needing to recover earlier information. Its greatest flaw can be mitigated by storing some data on each of the components.

1.9 Conceptual (or Logical) View

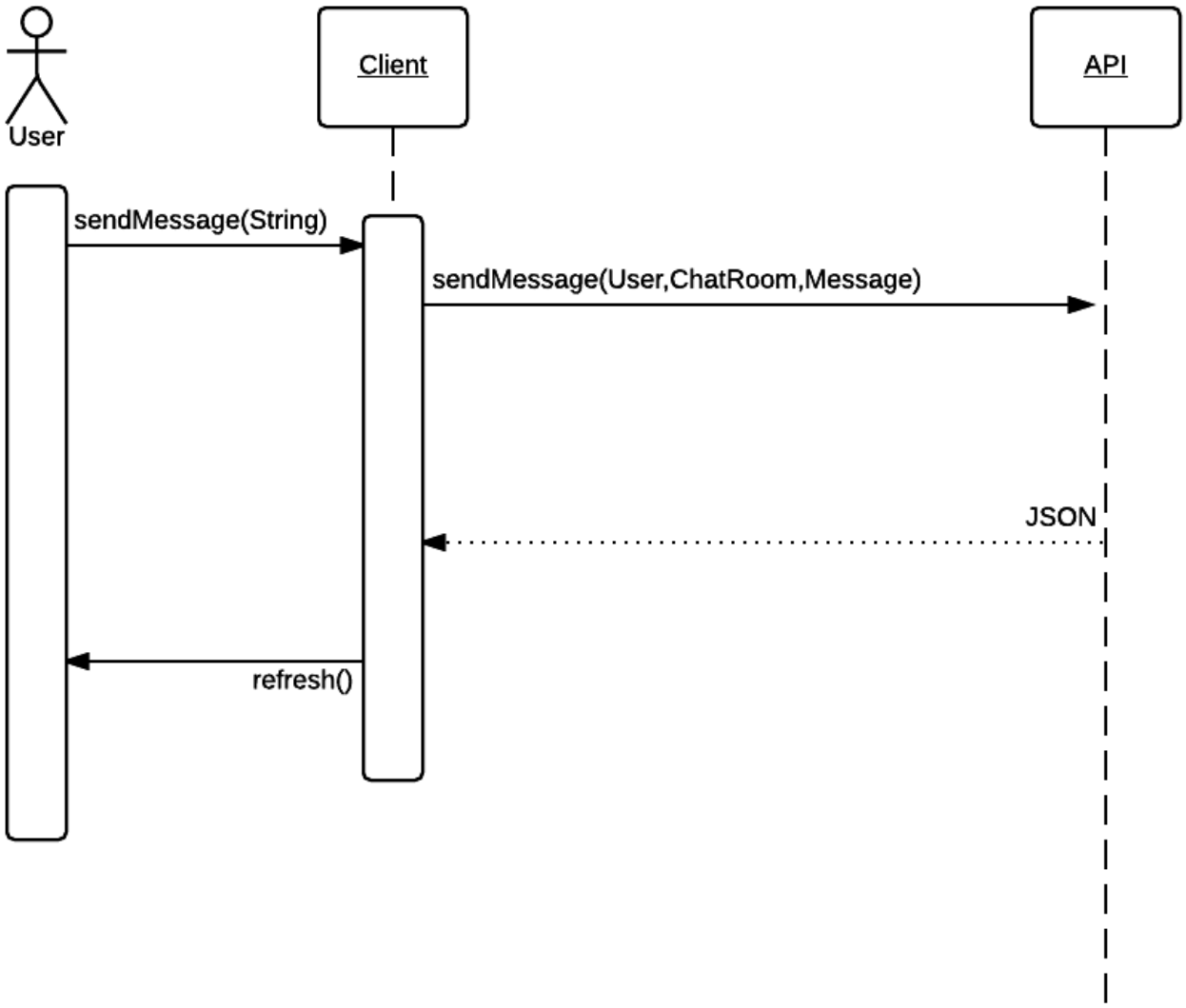


Low Level Design

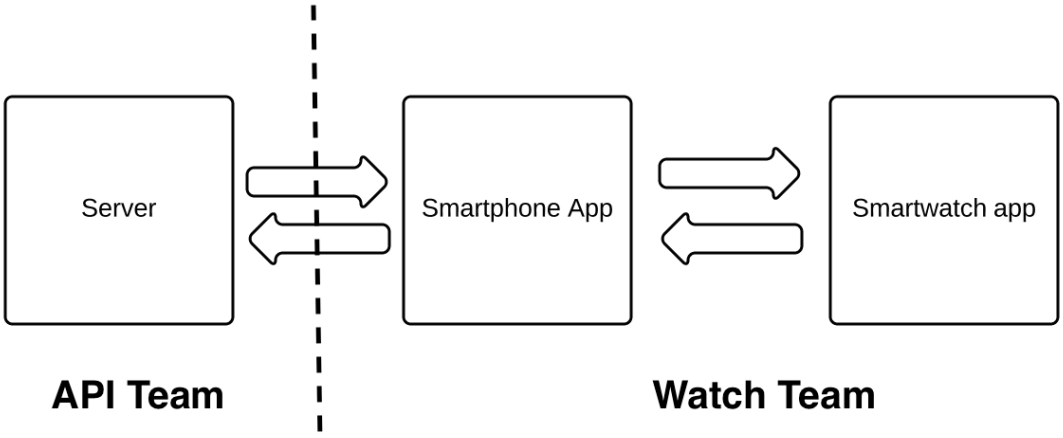
1.10 Class Diagram



1.11 Sequence Diagram



1.12 Component Diagram



User Interface Design