Software Requirements Specification

Version 1.0

Enterprise Chat App “eChat”

Contents

[**Introduction** 3](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.gjdgxs)

[**Customers** 3](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.30j0zll)

[**Functionality** 3](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.1fob9te)

[**Platform** 3](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.3znysh7)

[**Development Responsibilities** 3](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.2et92p0)

[**User Class and Characteristics** 4](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.tyjcwt)

[**System Features** 4](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.3dy6vkm)

[**Performance Requirements** 4](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.1t3h5sf)

[**Safety Requirements** 4](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.4d34og8)

[**Security Requirements** 5](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.2s8eyo1)

[**Software Quality Attributes** 5](https://docs.google.com/document/d/1C29IhCbw0LR0QL7HuDsBf6NtrkIycOgp/edit#heading=h.17dp8vu)

**Introduction**

This document details the project plan for the development of “eChat.”

It is intended for developers, designers, and testers working on “eChat” as well as project investors. This plan will include a summary of:

* how the system will function
* the scope of the project from the development viewpoint
* the technology used to develop the project, and
* the metrics used to determine the project’s progress
* Overall Description

Companies need remote communication tools, especially now that more people are working from home. The problem is that most companies end up using multiple applications to accomplish this: one for text chat, one for video chat, and one for conferences and meetings. “eChat” will solve this problem by combining these features in one application.

**Customers**

The customers will be enterprise companies. It will not target the general public.

**Functionality**

* Users should be able to sign up with enterprise LDAP accounts.
* Users should be able to create ad hoc chat groups comprising sets of users and send private messages to individual users.
* Users should be able to have text chats that they can break into threads.
* The application should be able to handle group video chat of up to 100 users at a time.

**Platform**

The application will be developed in [React Native](https://relevant.software/blog/how-to-hire-react-native-developer-step-by-step-guide-for-executives/) to enable the creation of a web-based application, an [iOS](https://relevant.software/blog/hire-ios-developers/) mobile app, and an [Android](https://relevant.software/blog/how-to-hire-android-app-developer/) mobile app.

These applications will connect to a REST API built with .NET Core to store and retrieve data from a MySQL database.

Authentication will be through existing LDAP installations.

**Development Responsibilities**

The developers on the “eChat” team will be responsible for writing all the code for the application, developing the database, and managing releases.

**User Class and Characteristics**

There will be a class of users called admin that will have permissions to access all functionality of the app, including:

* Creating channels where multiple users can interact
* Making these channels public to the entire company or private for a group of people
* Deleting these channels
* Archiving these channels

Standard users will have access to all functionality of the app except those listed above.

**System Features**

**Functional Requirements**

* Users should be able to create ad hoc chat groups comprising sets of users and send private messages to other users.
* Users should be able to have text chats that they can break into threads.
* Chats should be able to be archived indefinitely so users can reference past chats.
* Users should be able to upload files to chats for reference.
* External Interface Requirements

**User Interfaces**

* Front-end software: React Native
* Back-end software: .NET Core
* Database software: MySQL
* LDAP connection: Authentication in an enterprise environment

**Hardware Interfaces**

* Both Mac and Windows operating systems through their default web browser
* iPhone
* Android
* Non-Functional Requirements

**Performance Requirements**

* The application should load and be usable within 3 seconds
* The application should update the interface on interaction within 2 seconds
* The database should be normalized to prevent redundant data and improve performance
* The database should be distributed to prevent outages

**Safety Requirements**

* Databases should use sharding to be redundant to prevent loss of data.
* Backups of the databases should be done hourly and be kept for one week.

**Security Requirements**

* Any keys used for the REST API should be stored securely.
* Only the REST API should be able to connect to the databases.
* Databases should be behind a firewall.

**Software Quality Attributes**

* Availability: Because this application is critical to business communication, we will have a goal of four nines (99.99%) availability.
* Correctness: The application should never allow anyone to read messages or discussions not intended for that person.
* Maintainability: The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime.
* Usability: The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.