**Project Requirements**

Brady Berner

CST-451 Capstone Project Requirements Document

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 1.0

Date: 11/1/20

**ABSTRACT**

Communication is vital in today’s world. Companies rely on it to stay operating at break-neck paces. Consumers have grown so used to near-instantaneous communication that it is a must-have for many. However, many of the market applications today have security flaws, reliability issues, or cost consumers to use. This project will provide uses with a messaging application built to be fast, reliable, and secure MicroChat will allow users to easily communicate with each other, potentially from virtually any platform they desire. The application will remain simple to appeal to as many consumers as possible, regardless of demographics. To operate the application, a consumer can go to the website, log in, or create an account, select whom they would like to contact, and begin sending and receiving messages. The application could also serve as a form of consumable entertainment with a feature that would enable users to select whether they would like to start a conversation with another user chosen at random. The application could be expanded upon to allow for encrypted messaging to ensure user security and privacy and support multimedia messages such as pictures, videos, files, et cetera.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 11/1/20 | Brady Berner | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Functional Requirements 4

Non-Functional Requirements 5

Technical Requirements 6

Logical System Design 7

User Interface Design 9

Functional Requirements



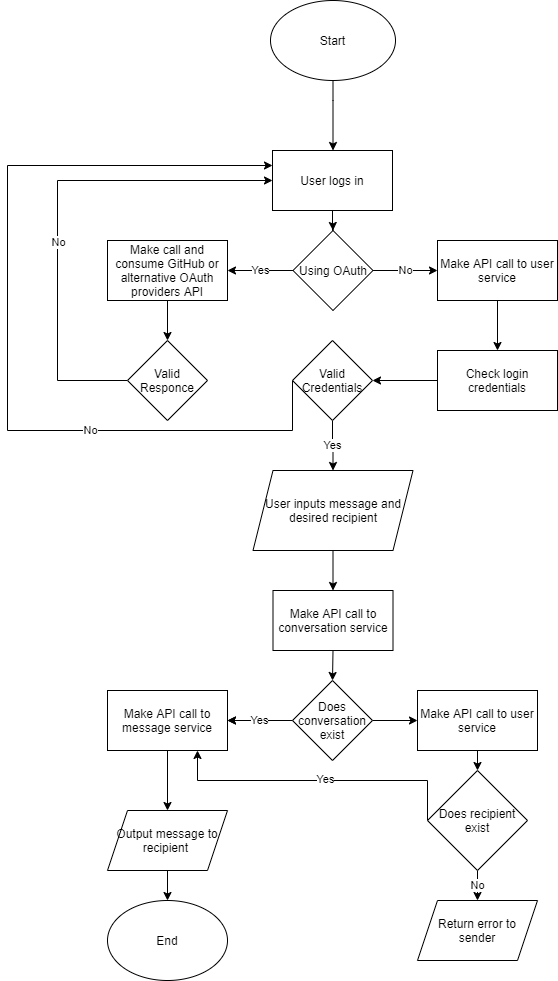
Non-Functional Requirements

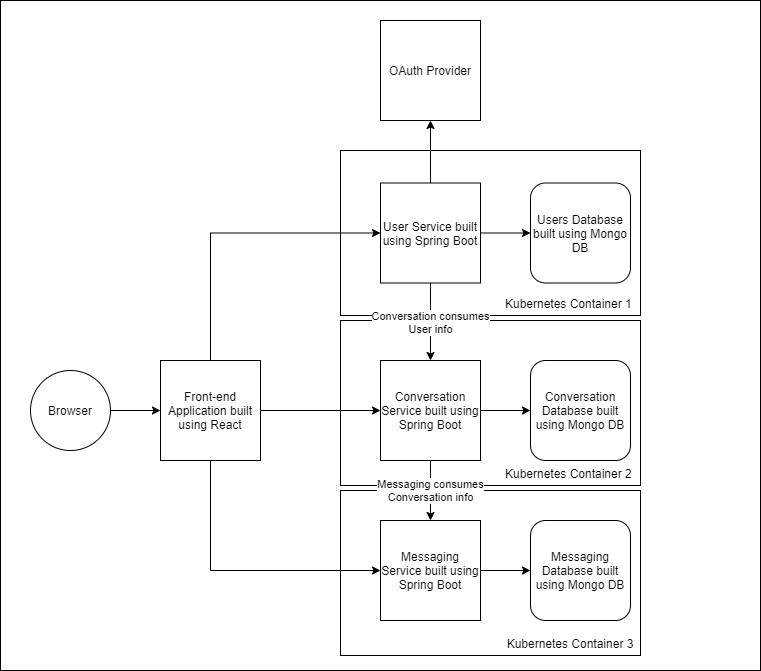
|  |  |  |
| --- | --- | --- |
| **Use Case or User Story** | **Approval Date** | **Justification** |
| 90% Uptime | 11/1/20 | Since the application will be locally hosted at first there is a high chance of system downtime however a 90% uptime would still only allow for 2.4hrs of downtime a day or ~17hrs a week. |

Technical Requirements

|  |  |  |
| --- | --- | --- |
| **Technology or Tool** | **Approval Date** | **Justification** |
| MongoDB (Version 4.4) | 11/1/20 | Easy to use cloud-based solution |
| MongoDB Compass (Latest) | 11/1/20 | Recommended MongoDB GUI |
| Spring Boot (Version 2.4.0 M3) | 11/1/20 | Great framework to create micro services with |
| IntelliJ IDEA (Latest) | 11/1/20 | Best known Java IDE and an industry standard |
| React (Version 16.13.1) | 11/1/20 | Powerful and popular front end technology |
| WebStorm (Latest) | 11/1/20 | Another JetBrains IDE so the GUI and other functionality is easy to adapt to |
| Material UI (Latest) | 11/1/20 | Well known React components collection |
| Kubernetes (Version 1.19.2) | 11/1/20 | Industry leading container technology for containerizing/deploying services |
| OAuth 2.0 | 11/1/20 | Easy way to allow users to use existing accounts to login to the application |
| REST API | 11/1/20 | Industry standard way of communicating between services and applications |
| Java 8 | 11/1/20 | Most familiar language for back-end |
| MongoDB Atlus | Currently Out of Scope | Takes the weight of MongoDB off of the application and eases deployment |
| GitHub | 11/1/20 | Standard source control solution |

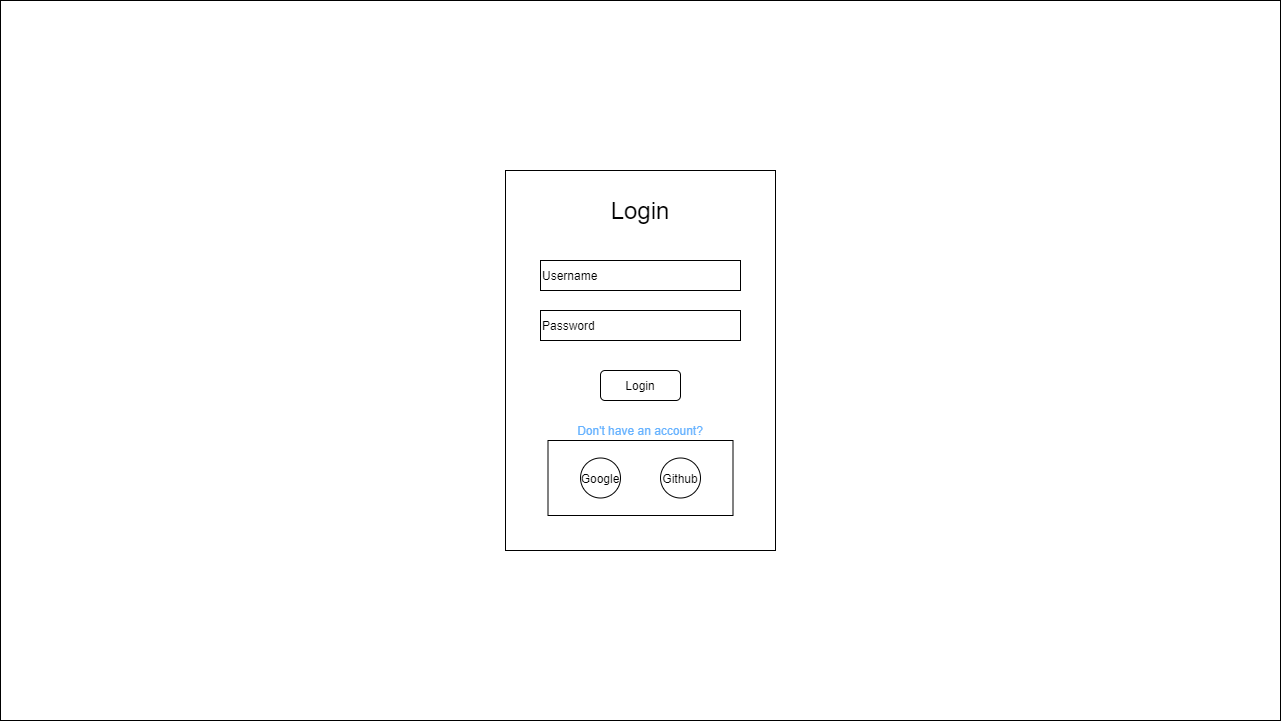
Logical System Design



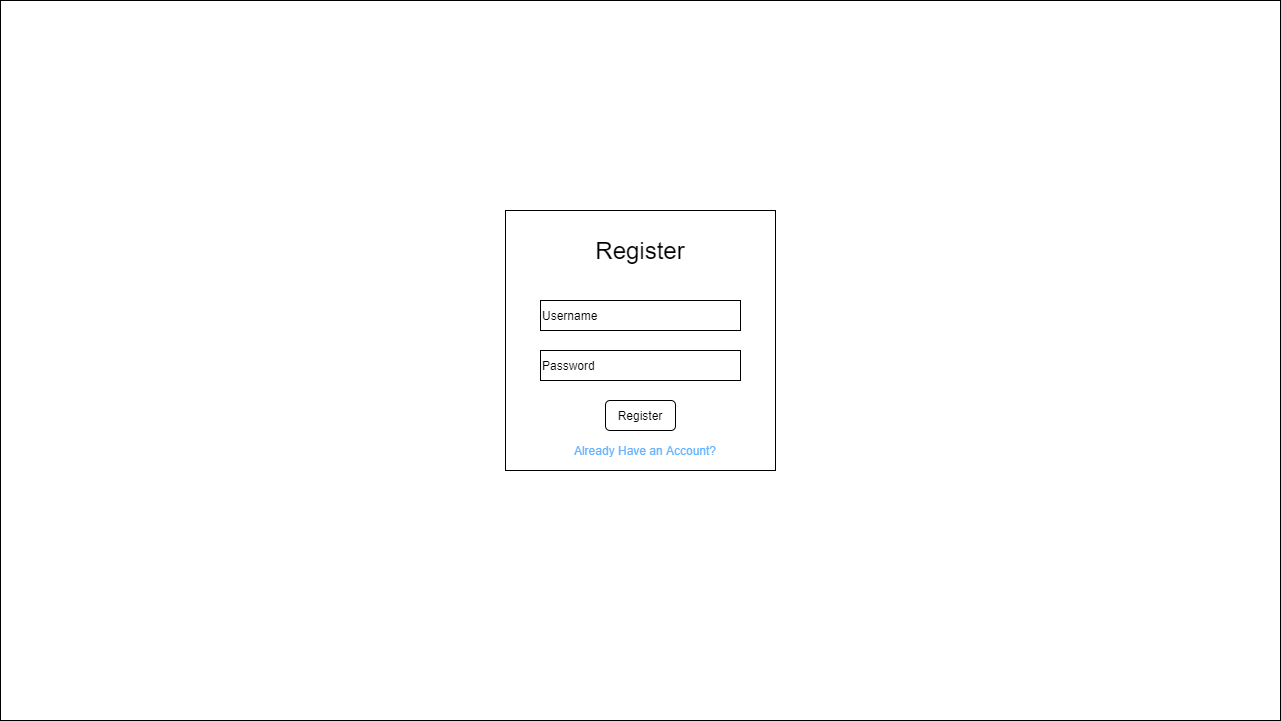


User Interface Design

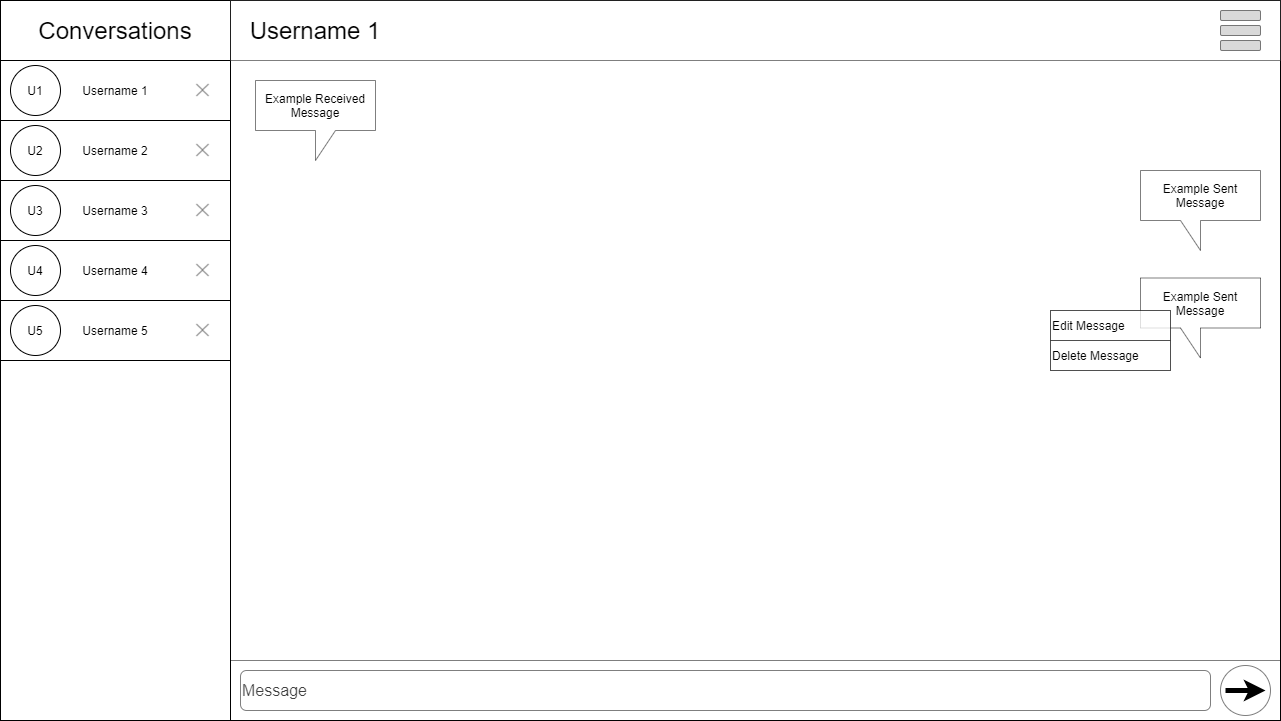
Login Page:



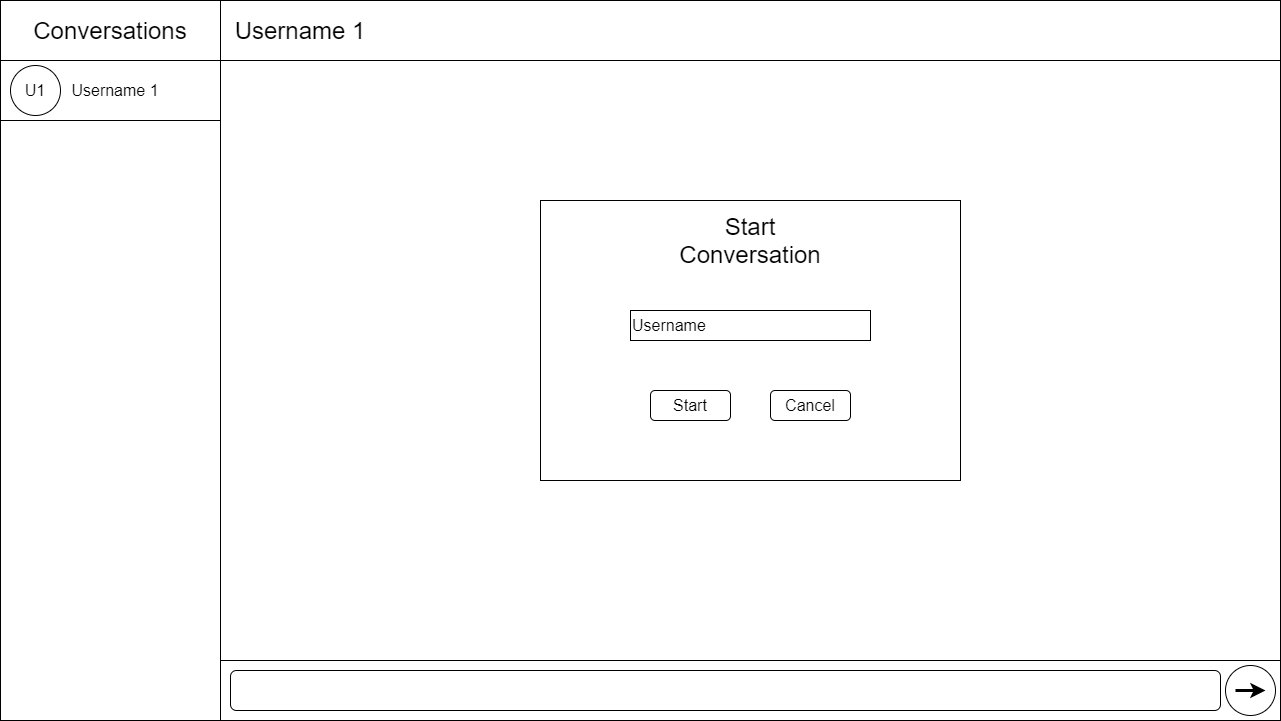
Registration Page:



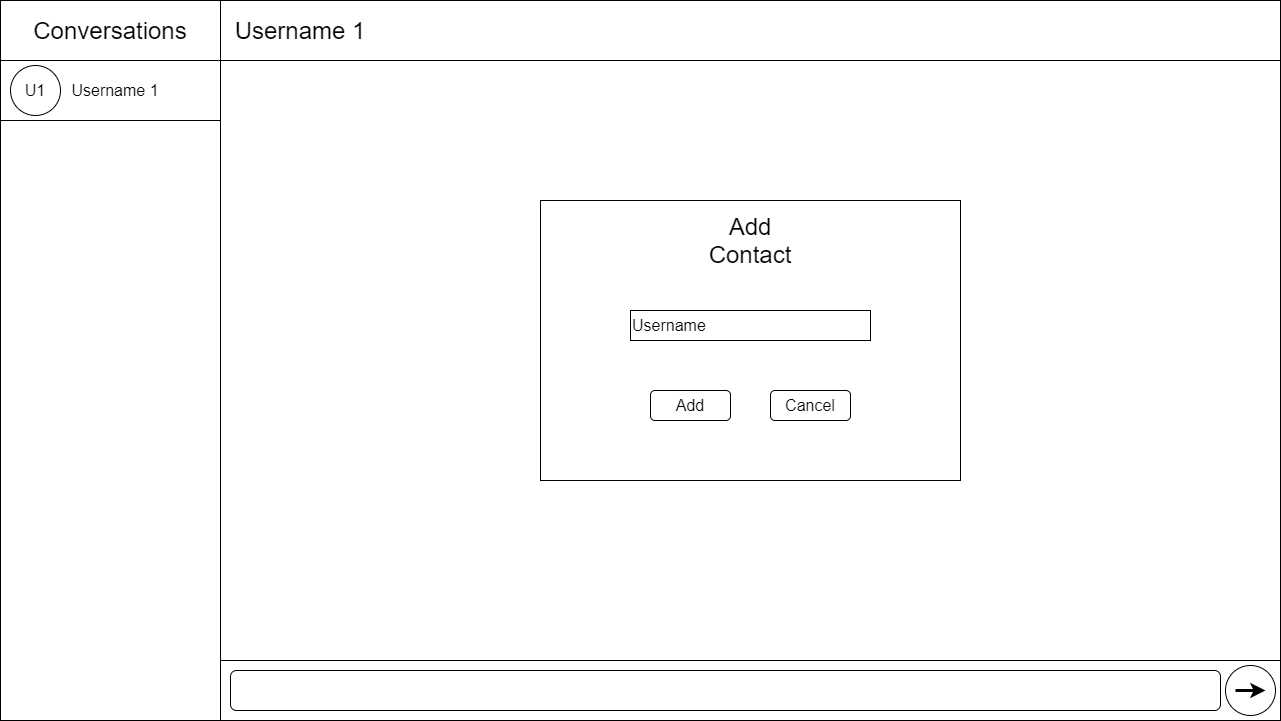
Main Page:



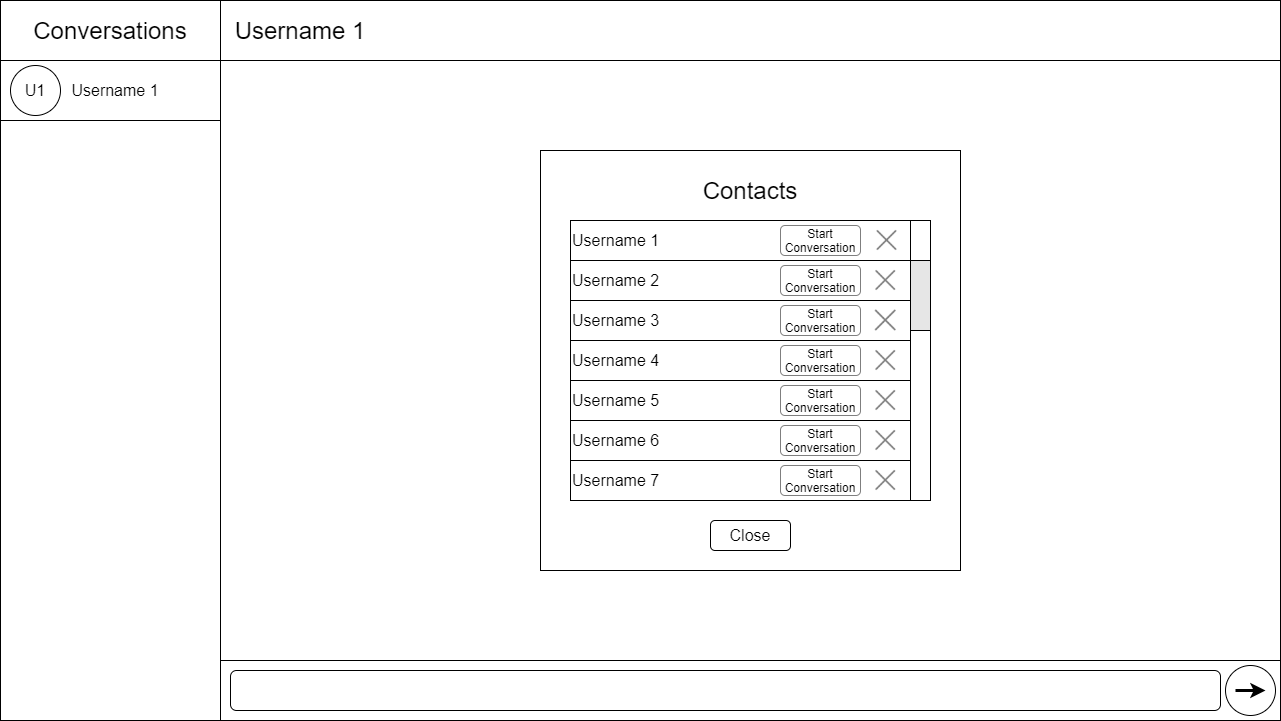
Start Conversation Modal:



Add Contact Modal:



Contacts Modal:



Sitemap:

(Most application functionality will be contained within modals on the main page)

