Project Proposal

Our chat app is designed to help people connect and chat with friends in a simple and straightforward way. After creating a profile, users can easily add friends and start chatting with them one-on-one or in groups. Users will be able to share attachments to make conversations more fun and engaging. We hope that our app will help users stay in touch with friends and make new connections in a fun, convenient, and user-friendly way.

This is how it will work: There will be multiple users in the system; They will be able to interact simultaneously with the system and communicate with each other. Before that, the user needs to log in and add some details about themselves to create their profile, such as their email and password. The user also needs to accept mandatory behavior conditions before finalizing the account. Once in, users can make friends and add them to their friend list. They can create private group chats or stay on the main public chat. There will be a report option for other users to report users who violated the agreed upon conditions.

WRSPM Analysis

| World assumptions | Assuming all users have an email address that they will use to sign up for the application.  Assuming the user’s friend has an account on the chat application  Assuming users are interacting from all over the world  Assuming users have an internet connection  Assuming the devices used by users have the processing power to handle requests  Assuming some people who use the system will have conditions that deviate from the average, such as color-blindness  Assuming people have different levels of technical skills, some may navigate better than others  Assuming some users want to upload a profile picture |
| --- | --- |
| Requirements | The user wants to sign in with their email address  The user wants to add people to their friend list  Some users want to upload a profile picture  The user wants to be able to chat with other online users  The user wants his color blindness to not affect his ability to use the app  The user wants to receive messages in real-time  The user wants to hold private conversations with only certain other users  The user wants his or her privacy protected  The user wants to block users who make them uncomfortable |
| Specifications | The system must enter the user’s email address and password in input fields.    The system must take the user’s picture and add it to be shown on their profile.    Users can make "friends" by sending friend requests to other users, which the receiving user can either accept or decline. The systems will take the user’s username into an input text field to search for their profile in the database and allow the user to add them as their friend.    The user shall pick a unique username to be displayed for public to see and the email and password shall be kept hidden from the public.  The system should provide a button for users to block other users and cut off any contact with user. This will be done by stopping the interaction that is associated with their two email addresses in the app database.  The system should be color friendly for users with varying degrees of colorblindness |
| Program | XML will code the input text field, allowing the user to enter text into the user interface/GUI.  Java and Kotlin will handle the behind the scenes code that include class files    Google’s Firestore will be used to store user data and provide real-time communication for users. |
| Machine | The user’s android device will show the UI/GUI  User’s device will access the database, run the program, and display the UI/GUI. |
| Environment - hidden | Email address (not entered)  User’s picture (not posted)  Username/email (not entered) |
| Environment - visible | Entered email address  User’s picture (posted as their profile picture)  Username/email (entered in the input text field) |
| System - visible | Input text fields  User’s image  Usernames  Search button  Toast message if the user doesn’t exist  Prompt notifying the user that a new contact has been added to their friend’s list. |
| System - hidden | Data stored in the system  Image stored in the system’s database  Usernames/email list in the database and other personal user information. |