Deliverable II

Melanie Brady

Project 4: Messaging Web Application

COP4331 Fall 2020

TABLE OF CONTENTS

High Level Design

Melanie Brady

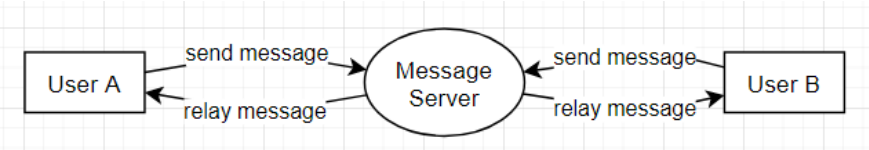
Project 4: Messaging Web Application

COP4331 Fall 2020

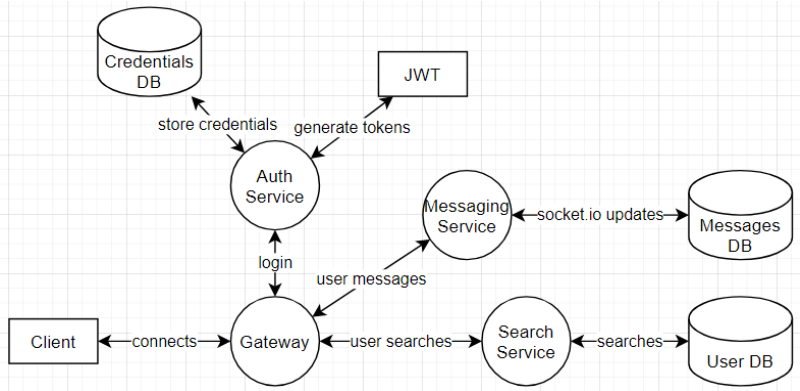
Contents of this Document

High-level Architecture

Client-Server Model High-Level View:



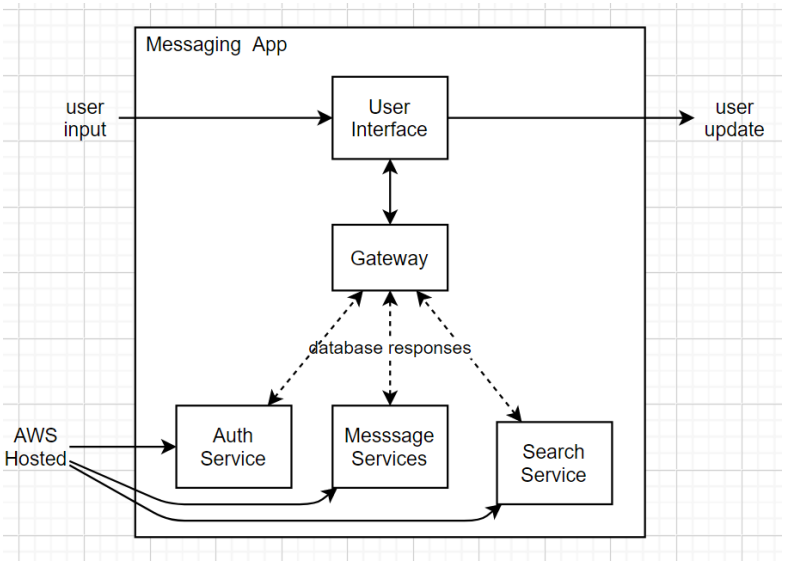
Overview:



The messaging application relies on a client-server relationship, so the client-server model is the best fit. Whenever a user sends a message to another user, it must first go through the server then update the other user. The major parts of this messaging client are the client's user interface, the gateway, the authentication service, messaging service, and search service, and their related databases. The gateway allows for more services (such as the messaging service and search service) to be added easier to the software without having to refactor much. It allows for the software to easier to sustain and grow. The authentication service communicates the credentials database to verify the user and JWT creates the

session token. The messaging services sends the string message to the user database then socket.io sends an update to the other user. That is why I choose to use the client-server architectural model.

System Interfaces:



Input is user input, whether it's a username and password, message to a specific friend, or searching for a specific friend. The major parts of this messaging client are the client's user interface, the gateway, the authentication service, messaging service, and search service, and their related databases. The gateway relays the input to the correct database then returns the data the user is looking for. Messaging services handles sending and receiving messages, search services returns what users are in the database that are similar to the input username searched for, and the auth service verifies accuracy of username and authenticates passwords.

Design Issues

This project is dependent on the client's application to communicate with a third-party database (MongoDB). MongoDB is a very reliable. A possible issue that could happen is if the server that the code is being hosted from goes down, then users will not be able to communicate. One of the challenges will be interfacing the client with the REST API, by using MongoDB an object-oriented database, it is a lot easier than using a relational database. From there I will use Nodejs due to it being industry standard developing an API. By breaking the software down into separate services, it allows for the code to be refactored, reused, maintained, and for it be easier to add additional services. By using react and Electron, it will allow for the application to be used regardless of being a mobile app, desktop app, or website giving it portability.

Detailed Design

Melanie Brady

Project 4: Messaging Web Application

COP4331 Fall 2020

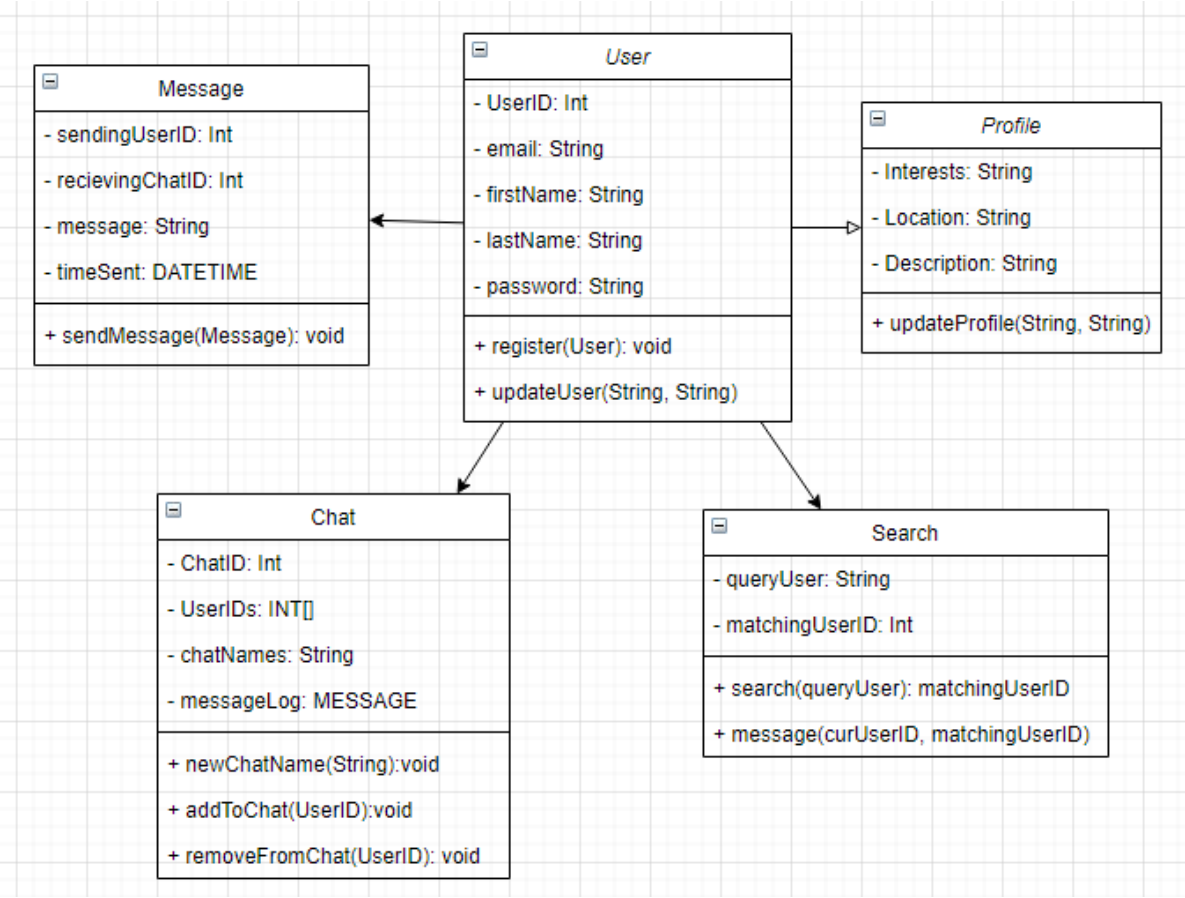
Contents of this Document

Detailed Design Issues

By breaking the software down into separate services, it allows for the code to be refactored, reused, maintained, and for it be easier to add additional services. By using react and Electron, it will allow for the application to be used regardless of being a mobile app, desktop app, or website. This project is dependent on the client's application to communicate with a third-party database (MongoDB). MongoDB is a very reliable. A possible issue that could happen is if the server that the code is being hosted from goes down, then users will not be able to communicate. One of the challenges will be interfacing the client with the REST API, by using MongoDB an object-oriented database, it is a lot easier than using a relational database. From there I will use Nodejs due to it being industry standard developing an API. I will be using more third-party software to make the app as portable and streamlined as possible. The problem with using third party software is when it eventually has an update, then there is a risk of it breaking your already established software.

Detailed Design Information

The program starts off with a user being able to register. From there, the user inputs their information and the register function go through the Auth service and updates the user's credentials to allow for the user to login. The auth service verifies accuracy of username and authenticates passwords. The user later can update their email, first name, last name, or password. The user may also create an optional profile displaying more information about themselves. The user can search for another user and then message them. Once a user is messaging another user, they can add more users to the chat, change the chat name, and send messages. The gateway relays the input to the correct database then returns the data the user is looking for.



Trace of Requirements to Design

ID	Requirement	Architecture Reference	Design Reference
1	Verify the user has internet connection	Auth Service	User
2	User can create an account	Auth Service	User
3	User can log in	Auth Service	User
4	Users can search other users	Search Service	Search
5	User send another user a message	Messaging Service	Message
6	User can update their profile	Auth Service	User
7	Users can call other users	Messaging Service	Message
8	Users can schedule calls with other users regardless of friend status with an optional conversation prompt based off of their interests.	Messaging Service	Message
9	Users can reset their password.	Auth Service	User
10	All data stored in the database will be accurate.	Auth Service	User
11	When logging in, the database should return the correct profile's information.	Auth Service	User
12	Verify the server is pushing newly sent messages to the receiving user.	Messaging Service	Message
13	Verify data stored in the database is being stored as the correct data type.	Messaging Service	Message
14	The programs should be able to run as long as they are connected to the internet.	Auth Service	User
15	The programs should be able to work on Android devices.	N/A	N/A
16	The application will have user profile.	Auth Service	User
17	The application will have admin profile.	Auth Service	User
18	Users are expected to be familiar with Android and websites. If not, there will be instructions for how to use the app.	Auth Service	User
19	All advanced documentation will be available through GitHub	N/A	N/A
20	The users will be able to click a "help me" mode for instructions on how to use the app!	Auth Service	User
21	Only simple math will be needed for this program.	N/A	N/A
22	The program will use low precision due to solely using integers for calculation.	N/A	N/A
23	The program will most retain strings in the form of user data. The only numbers will be related to time stamps, userIDs, and time zones.	Messaging Service	Message
24	The program will be made by one developer.	N/A	N/A
25	A website will be needed for accessibility.	N/A	N/A
26	A database will be needed to maintain data accuracy.	Messaging Service	Message
27	The application will be compatible with android devices.	Messaging Service	Message
28	The access to the system will be controlled by the application and its users.		

29	User data will be isolated from each other within the database.	Gateway	User
30	The Operating System will handle keeping the program separate from other programs.	N/A	N/A
31	The system will back up daily in a separate directory.	N/A	N/A
32	The system will be safe from fire, water, damage, theft due to be stored on a third party database.	N/A	N/A
33	User profiles would be backed up daily and restored if needed.	Auth Service	User
34	Quality assurance test to make sure the application meets reliability, availability, maintainability, security, and portability standards.	All	All
35	The application will backup data so that no data is lost.	All	All
36	The application will have continuous uptime and will restart if any updates are needed.	All	All
37	The application will detect errors and update the user as needed	All	All

Test Plan

Melanie Brady

Project 4: Messaging Web Application

COP4331 Fall 2020

Contents of this Document

Overall Objective for Software Test Activity

The tests will assure the user a streamlined experience. The tests will help verify that users can focus solely on communicate with each other. The goal of the testing will find out any accidentally overlooked bugs. Testing will also verify that all requirements and goals for the program has been satisfied.

Description of Test Environment

Testing will first be done through testing frameworks from there. User tests will be conducted on a desktop and an android device. The developer will be testing the program. The test environment be the same environment in which the software will operate.

Stopping Criteria

Whenever a bug is discovered, the bug will be prioritized to be resolved as soon as possible. As soon as the bug had been fixed, then testing will continue. It is impossible for any software to be error-free once you start getting to a larger scale project, it is just minor bugs you haven't discovered yet. Enough to deliver will be when the program operates, all test cases pass, and it means all requirements.

Description of Individual Test Cases

ID	Objective	Description	Cond	Expected Results
1	Verify internet connection	Pings to see if there is a steady internet connection. If there is no connection, then it will notify the user.	User side.	App will either load or the error result will load.
2	User Registration and Authentication	Authenticates any new users and registered users through the Auth Service makes several API calls to verify with MongoDB that is a valid username, password, and token.	User and server	Server returns a success on authentication and existing user. User logs in.
3	Application loads	The messaging app homepage loads successfully after a user has been verified. From there the user should be able to see options to search, message other users, other users.	User and server	The application will load and show the user all their options.
4	Search test	The input username, email, first name, or last name is queried through the Search service and check within MongoDB and the resulting user or lack of user is returned the other client.	User and server	The user will be able to search for another user regardless of name, email or username. If the user is not found notify the user.
5	Message test	Each message should post updates to the user. The test will involve verifying that only authenticated (users with token) users can send and receive messages as well as verifying new messages are received as fast as possible.	User and server	Messages are sent and received timely and by only authenticated users.
6	Data Accuracy test	Data will be frequently sent and received to the databases through different services. The data will be posted then retrieved from the database.	User and server	Verify that all data has successfully posted to the database as expected.
7	Tutorial mode	Tutorial offered to newly registered users that walks through the steps of how to search and message other users.	User	Tutorial loads, explains how to use it well, and is easy to close for experienced users.
8	Website test	Verify desktop and mobile website experience is the same.	User and server	Responsively designed.
9	Client Side & Android Test & Storage	Double check all the data needed locally on Android devices for the user to use the app can load successfully without taking many resources from the OS.	User and server	App loads efficiently and successfully on Android devices.
10	Server robustness	Verify the server is consistent and reliable.	Server	Server maintains uptime.

Trace of Individual Test Cases to Requirements

ID	Requirement	Test Case Reference	Test case	Testing Status
1	Verify the user has internet connection	The program will check if there's internet connection. If not then it will notify the user to try again later.	1	Passed
2	User can create an account	Check the database to see if the username exists. If the username does not exist, then update the database with the new username and password.	2	Passed
3	User can log in	Verify the user can log in with their credentials.	2	Passed
4	Users can search other users	User can successfully search for other users.	4	In Progress
5	User send another user a message	User can send and receive messages from other users.	5	In Progress
6	User can update their profile	User can store their first name, last name, time zone, location, interests, so on…	2	Passed
7	Users can call other users	User can send and receive messages from other users.	5	In Progress
8	Users can schedule calls with other users regardless of friend status with an optional conversation prompt based off of their interests.	User can schedule calls with other users mindful of their time zone.	5	In Progress
9	Users can reset their password.	User can reset their password.	2	Passed
10	All data stored in the database will be accurate.	Compare data stored in the database compared to the user input.	6	Passed
11	When logging in, the database should return the correct profile's information.	Verify the correct profile is retrieved from the database.	6 & 2	Passed
12	Verify the server is pushing newly sent messages to the receiving user.	Verify messages are successfully received right after sending.	5 & 10	In Progress
13	Verify data stored in the database is being stored as the correct data type.	Username, password, location as strings. Time zone in int. Interests as either Boolean or strings.	6	Passed
14	The programs should be able to run as long as they are connected to the internet.	Verify it works with internet connection regardless of location.	1	Passed
15	The programs should be able to work on Android devices.	Test on android devices.	9 & 1	Passed
16	The application will have user profile.	Verify user profile has successfully been created.	2	Passed
17	The application will have admin profile.	Verify admin profile has successfully been created to test beta mode.	2	In Progress
18	Users are expected to be familiar with Android and websites. If not, there	Test with different users to verify they have a basic understanding of Android devices and websites to practice using	7	Passed

	will be instructions for how to use the app.	the application. Include instructions on how to use the app.		
19	All advanced documentation will be available through GitHub	All documentation will be accessible through GitHub.	N/A	Passed
20	The users will be able to click a “help me” mode for instructions on how to use the app!	Verify with test users that the instructions are easy to understand.	7	In Progress
21	Only simple math will be needed for this program.	Verify calculations.	6	Passed
22	The program will use low precision due to solely using integers for calculation.	Verify correct data is being stored.	6	Passed
23	The program will most retain strings in the form of user data. The only numbers will be related to time stamps, userIDs, and time zones.	Verify calculations.	6	Passed
24	The program will be made by one developer.	One person will be developing this project.	N/A	Passed
25	A website will be needed for accessibility.	Have a user interface accessible online.	7	In Progress
26	A database will be needed to maintain data accuracy.	Verify accurate data within the database.	6	Passed
27	The application will be compatible with android devices.	Verify accurate data within the database.	6 & 9	Passed
28	The access to the system will be controlled by the application and its users.	Access will only be permitted to registered users.	2 & 5	Passed
29	User data will be isolated from each other within the database.	Each user will have their own ID.	2	Passed
30	The Operating System will handle keeping the program separate from other programs.	Successful running of the application.	9	Passed
31	The system will back up daily in a separate directory.	Successful backing up	10	Failed
32	The system will be safe from fire, water, damage, theft due to be stored on a third party database.	Third party databases maintain safe servers.	10	Passed
33	User profiles would be backed up daily and restored if needed.	Test data losses to verify all data is restored.	10	Failed
34	Quality assurance test to make sure the application meets reliability, availability, maintainability, security, and portability standards.	Test thoroughly to verify there are no bugs.	10	In Progress
35	The application will backup data so that no data is lost.	Remove data and test to see if it copies over recovered data properly.	10	Failed

36	The application will have continuous uptime and will restart if any updates are needed.	Verify any local files needed to be saved are locally saved.	10	In Progress
37	The application will detect errors and update the user as needed	Check for errors, log errors, maintain continuous uptime with python pulse checker.	1-10	Failed

Appendices:

Include any test files used in your test cases. If you do not use any test files, leave this section blank.