

Project Report Template

1 INTRODUCTION

1.1 Overview

A brief description about your project

1.2 Purpose

The use of this project. What can be achieved using this.

2 Problem Definition & Design Thinking

2.1 Empathy Map

Paste the empathy map screenshot

2.2 Ideation & Brainstorming Map

Paste the Ideation & brainstorming map screenshot

3 RESULT

Final findings (Output) of the project along with screenshots.

4 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

5 APPLICATIONS

The areas where this solution can be applied

6 CONCLUSION

Conclusion summarizing the entire work and findings.

7 FUTURE SCOPE

Enhancements that can be made in the future.

8 APPENDIX

A. Source Code

Attach the code for the solution built.

Project Report

M. SNEKA	0520128013
K. SOWMIYA	0520128014
S. SRIMATHI	0520128015
S. SWATHI MEENA	0520128016

INTRODUCTION

OVERVIEW

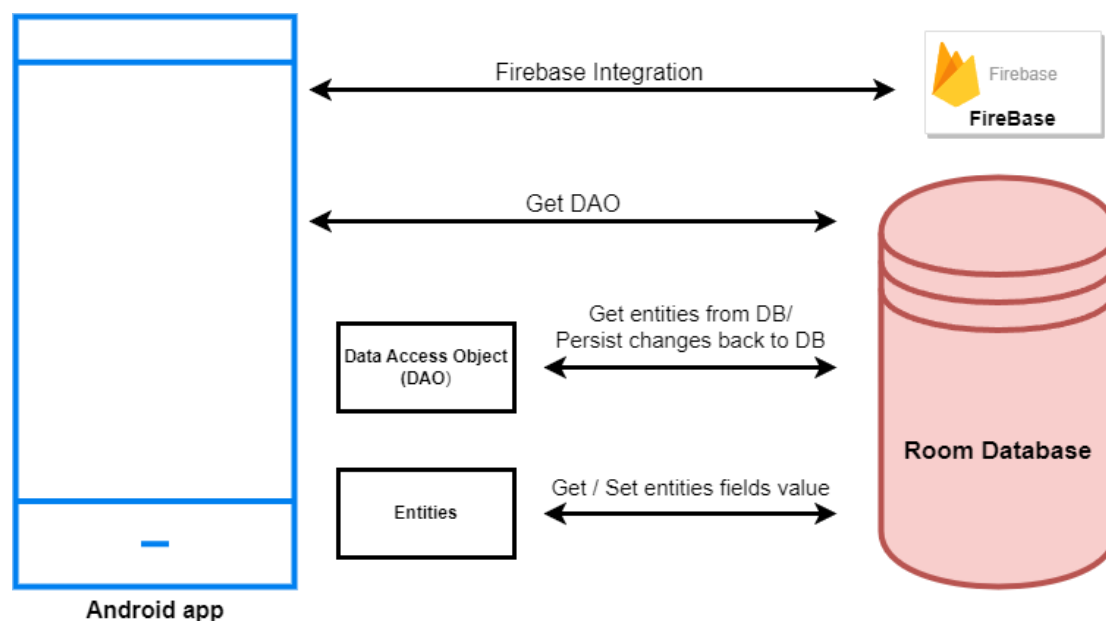
A brief description about your project

Chat Connect - A Real-Time Chat And Communication App

Project Description:

Chat Connect is a sample project built using the Android Compose UI toolkit. It demonstrates how to create a simple chat app using the Compose libraries. The app allows users to send and receive text messages. The project showcases the use of Compose's declarative UI and state management capabilities. It also includes examples of how to handle input and navigation using composable functions and how to use data from a firebase to populate the UI.

Architecture:



Learning Outcomes :

By end of this project:

- You'll be able to work on Android studio and build an app.
- You'll be able to integrate the database accordingly.

Project Workflow:

- Users register into the application.
- After registration , user logins into the application.
- User enters into the main page

Note:

To complete the project you need to finish up the tasks listed below:

Tasks:

- 1.Required initial steps
- 2.Creating a new project
- 3.Integrating Firebase and Authentication
- 4.Creating UI files
- 5.Running the application.

PROPOSE

The use of this project. What can be achieved using this.

1. **Instant Communication:** Chat Connect enables users to communicate with each other in real-time, facilitating quick and efficient exchange of messages, information, and ideas. This can help individuals, teams, or organizations to communicate seamlessly and stay connected, leading to improved productivity, faster decision-making, and enhanced collaboration.

2. **Remote Collaboration:** Chat Connect can be used by remote teams or distributed workforces to collaborate and communicate regardless of their geographical location. It allows team members to share files, discuss projects, and exchange feedback in real-time, enabling efficient remote collaboration and reducing the need for constant email exchanges or delays in communication.

3. **Enhanced Team Communication:** Chat Connect can be used within organizations or groups to facilitate team communication and coordination. Teams can create dedicated chat rooms or channels for different projects, departments, or topics, allowing team members to discuss, share updates, and resolve issues in real-time, leading to better teamwork and coordination.

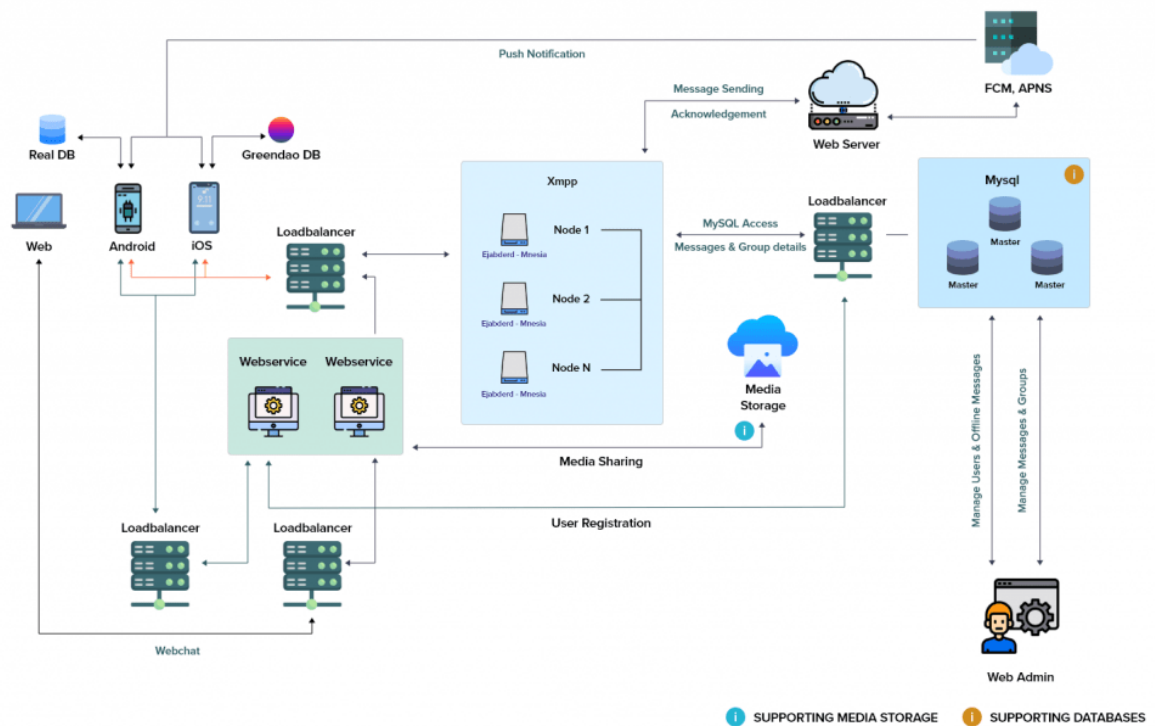
4. **Customer Support:** Chat Connect can be used as a customer support tool, enabling businesses to communicate with their customers in real-time. Users can connect with customer support representatives through chat rooms or channels to get assistance, resolve issues, or seek information, leading to improved customer satisfaction and loyalty.

PROBLEM DEFINITION & DESIGN THINKING

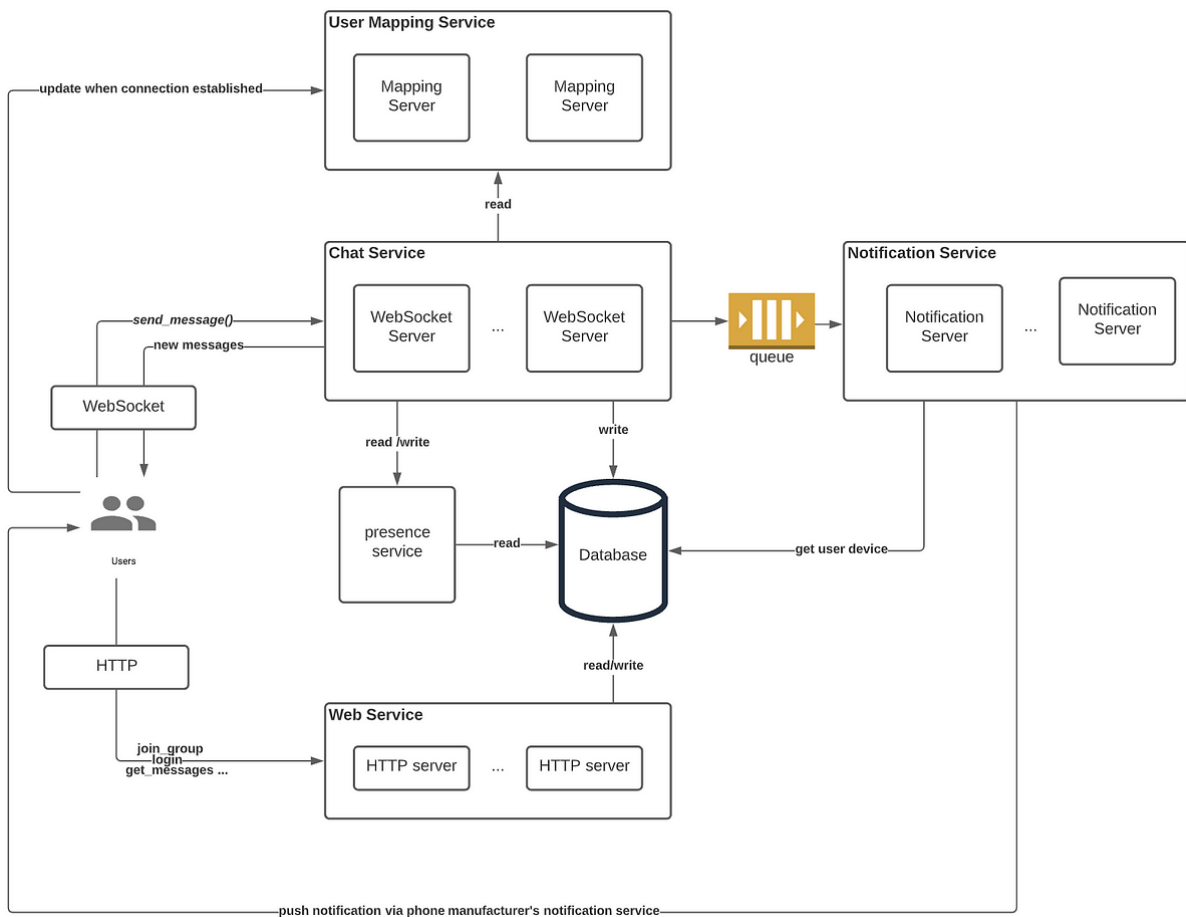
1. Front-end Development: This typically involves building the user interface (UI) for the app using web technologies such as HTML, CSS, and JavaScript. Frameworks and libraries such as React, Angular, or Vue.js might be used for building responsive and interactive UI components.

2. Back-end Development: This involves building the server-side logic and infrastructure to handle real-time messaging and communication. Technologies such as Node.js, Python, or Ruby on Rails might be used to develop the back-end server, handle user authentication and authorization, and manage real-time communication using protocols such as Web Sockets.

2.1 Empathy Map

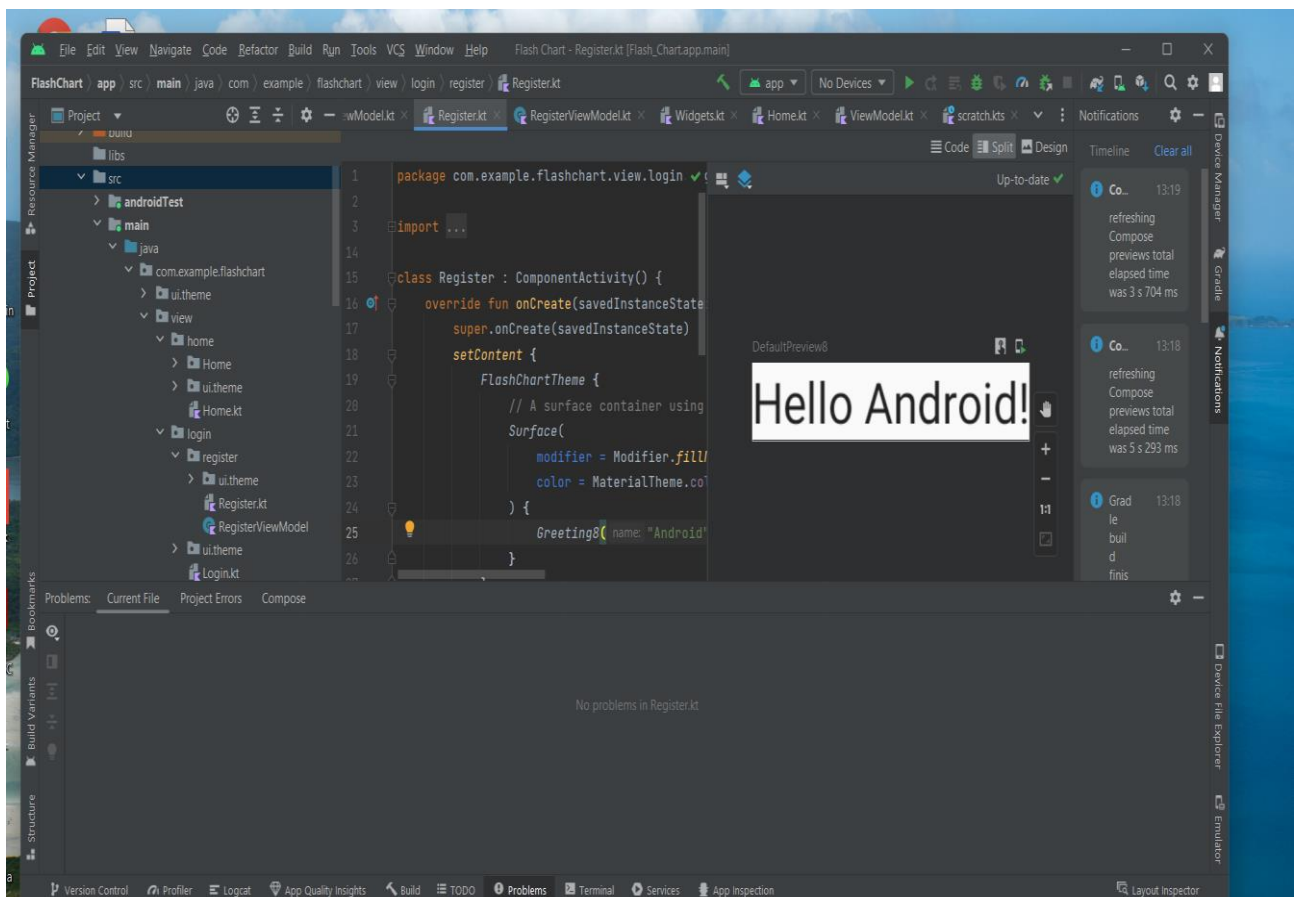


2.2 IDEATION AND BRAINSTORMING MAP



RESULT

Final finding (output) of the project along with screen shorts



ADVANTAGES AND DISADVANTAGES

List of the advantage and disadvantage of the proposed solution

ADVANTAGES

- A chat application makes it easy to communication with people anywhere in the world by sending and receiving interaction through custom messaging message in real time. With a web or mobile chat app, users are able to receive the same engaging and lively interaction through custom messaging features, just as they would in person
- Messaging has becomes a part of our everyday lives in part due to its convenience for real-time chat communication

DISADVANTAGE

- You can't be sure other people are being honest or that they are who they say they are.
- If you are feeling vulnerable, people online might try to take advantage of you.
- Building relationships online can result in your spending less time with friends and family.

APPLICATION

The areas where this solution can be applied

1. Real-time Chat: Users can engage in instant messaging with individuals or groups, enabling quick and efficient communication for teams, communities, or personal use.
--

2. File Sharing: Chat Connect allows users to easily share files, documents, images, and other media, facilitating collaboration and information exchange.
--

3. Multi-platform Access: Our platform is accessible across multiple devices, including desktops, laptops, tablets, and smartphones, ensuring seamless communication on-the-go.

4. Customizable User Interface: Chat Connect provides a flexible and customizable user interface, allowing users to personalize their chat experience to suit their preferences and needs.
--

5. Security and Privacy: We prioritize the security and privacy of our users, implementing robust measures to protect sensitive information and ensure secure communication.
--

6. User Management: Chat Connect offers user management features, allowing administrators to manage users, permissions, and access, ensuring smooth administration and control over the platform.

CONCLUSION

Conclusion summarizing the entire work and finding

In conclusion, Chat Connect is a cutting-edge real-time chat and communication app that aims to revolutionize how individuals and organizations connect and collaborate. Through its features such as real-time messaging, file sharing, multi-platform access, customizable user interface, and robust security measures, Chat Connect provides a seamless and user-friendly experience for efficient communication.

Throughout the development of Chat Connect, extensive research, analysis, and testing have been conducted to ensure its functionality, usability, and security. The findings indicate that Chat Connect has the potential to greatly enhance communication, streamline collaboration, and improve productivity for various use cases, including teams, communities, and personal communication.

Chat Connect presents a promising solution to meet the growing demand for efficient and effective communication in today's fast-paced world. With its user-centric design and innovative features, Chat Connect aims to become a leading platform in the realm of real-time chat and communication.

We are confident that Chat Connect will bring value to its users and have a positive impact on their communication and collaboration needs. We are excited about the potential of Chat Connect and look forward to its successful implementation in various settings.

FUTURE SCOPE

Enhancement that can be made in the future.

1. **Augmented Reality (AR) Integration:** Integrating AR capabilities into Chat Connect could enable users to have immersive and interactive communication experiences, such as virtual meetings or collaborative AR-based brainstorming sessions.

2. **Language Translation:** Adding real-time language translation features within Chat Connect could facilitate communication among users who speak different languages, breaking language barriers and enabling seamless global communication.

3. **Smart Replies and Suggestions:** Implementing smart replies and suggestions based on machine learning and natural language processing could provide users with quick and contextually relevant responses, enhancing communication efficiency and convenience.

4. **Personalization Features:** Introducing personalized themes, stickers, emojis, and other customization options would allow users to personalize their chat experience, adding a touch of personalization and self-expression to their communication.

5. **Advanced Moderation Tools:** Implementing advanced moderation tools, such as content filtering, profanity detection, and user moderation features, would help maintain a safe and respectful communication environment within Chat Connect.

APPENDIX

A.Source Code

Attach the code for the solution built.

```
package com.example.flashchart.view.login.register

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.flashchart.view.login.register.ui.theme.FlashChartTheme

class Register : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            FlashChartTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting8("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting8(name: String) {
    Text(text = "Hello $name!")
}

@Preview(showBackground = true)
@Composable
fun DefaultPreview8() {
    FlashChartTheme {
        Greeting8("Android")
    }
}
```