

A PROJECT REPORT ON
Chat Connect - A Real-Time Chat and Communication App

BACHELORE OF COMPUTER APPLICATION

DEVANGA ARTS COLLEGE,

(Affiliated to Madurai Kamaraj University)

Submitted

BY

M.JEEVA (C0S41809)

MD.BARATH SIVA (C0S41804)

S. SANTHOSH KUMAR (C0S41819)

S. PRABHU(C0S41817)



UNDER THE GUIDENCE OF

Mr.V.Gnanasekaran, M.Sc.,

DEPARTMENT OF COMPUTER APPLICATION

DEVANGA ARTS COLLEGE,

ARUPPUKOTTAI – 626101.

1.INTRODUCTION

1.1 Overview

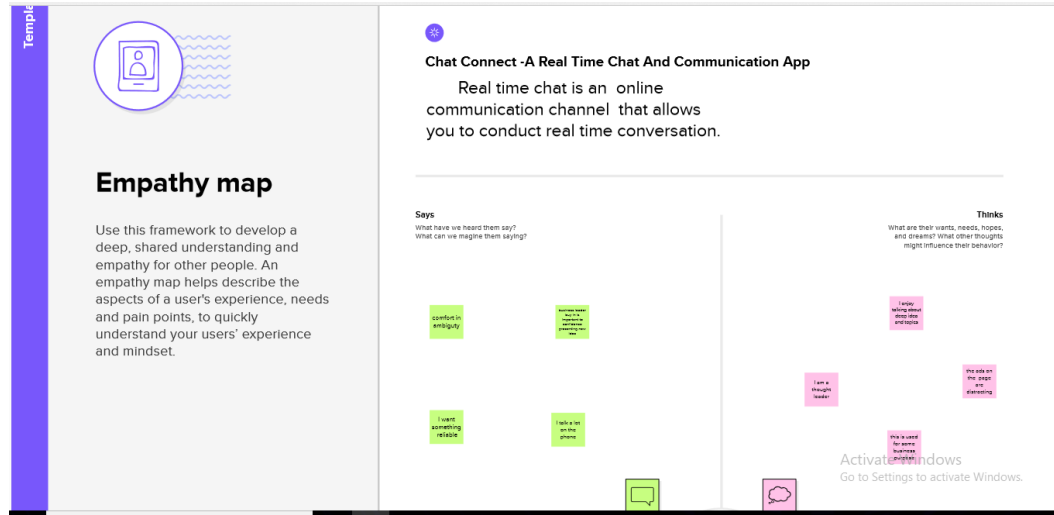
Chat connect allows you to interact with other ITIL staff in a productive way using a familiar chat tool similar to SMS on your smartphone. There are two ways to view Connect Chat, via the Sidebar or Workspace.

1.2 Purpose

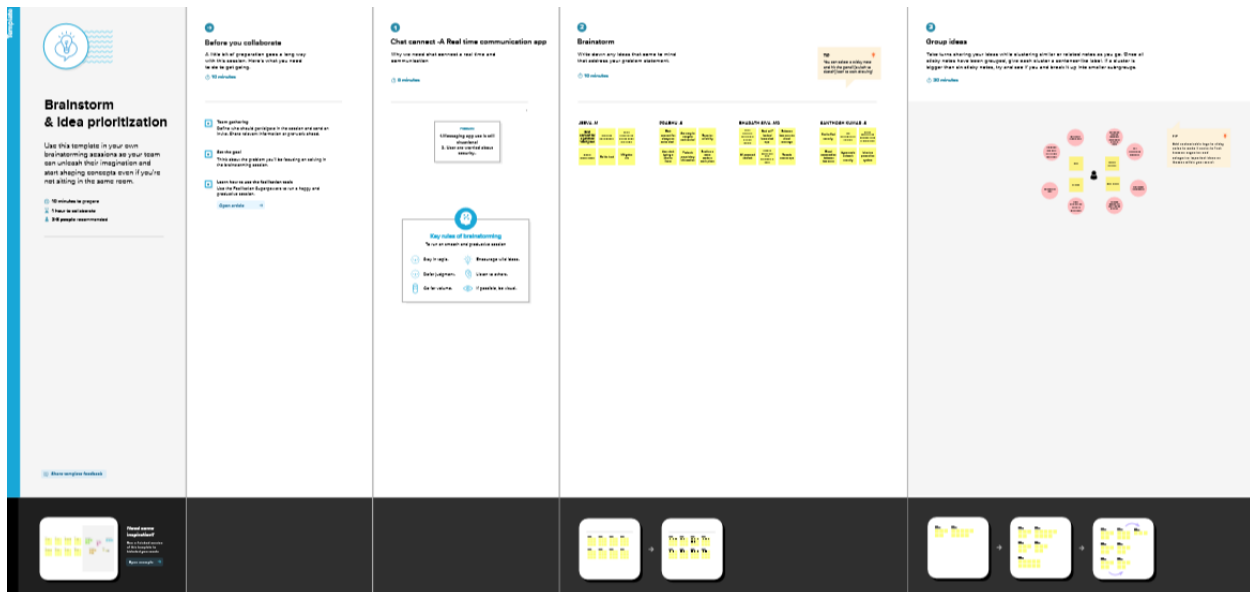
Our Mission. WhatsApp started as an alternative to SMS. Our product now supports sending and receiving a variety of media: text, photos, videos, documents, and location, as well as voice calls. Some of your most personal moments are shared with WhatsApp, which is why we built end-to-end encryption into our app.

2 Problem Definition & Design Thinking

2.1 Empathy Map



2.2 Ideation & Brainstorming Map

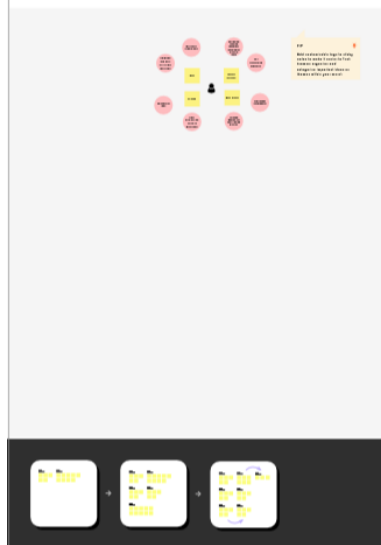


2

Group Ideas

There is one sharing your ideas while studying similar or related notes as you go. Once all study notes have been grouped, give each cluster a sentence-line label. If a cluster is bigger than six study notes, try analyzing it you and break it up into smaller subgroups.

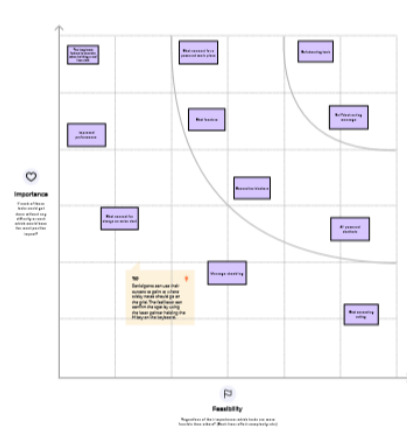
183 instruction



4

Priorities

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

 TED www.ted.com

After you collaborate

You can export the mural as an image or pdf to be shared with members of your community who might find it helpful.

Quick add-ons

Share the moral
Share a video online or the moral with exhortations to keep them in the bag about the successes of the seed oil.

Report the murder
Report a copy of the murder as a 2012 or 2013 as a search in email, include in others, or even in your drive.

Keep moving forward

Energy blueprint
Define the components of a new, lean energy.

Customer experience journey map
Understand customer needs, motivations, and expectations. See also: service blueprint.

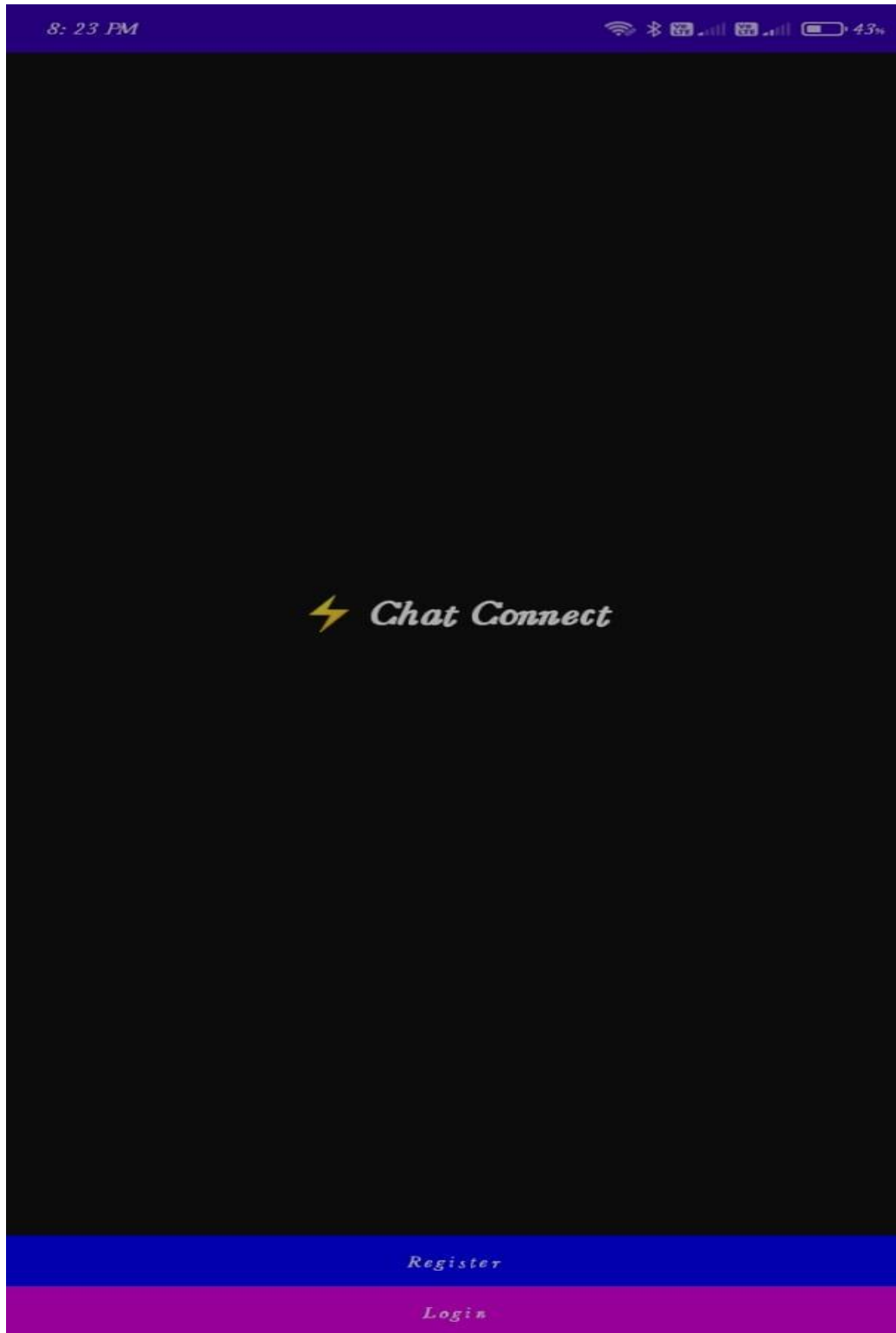
Strength, resilience, opportunities & stress

Open the template.

- Have complete feedback



3.RESULT



Register page:

8: 24 PM

← Register

Email

jesva@gmail.com

Password

Register

9: 20 PM 9.1KB/s

Bluetooth, Wi-Fi, 4G, VoLTE, 4G, 4G, 44%

← Login

Email

prabhu@gmail.com

Password

Login

8: 26 PM 9.5KB/s S



Type Your Message



Home screen:



ADVANTAGES

Sharing Information. The best platform for sharing and accessing information is chat connect.

Electronic Learning. Chat connect has recently evolved into a fictitious e-learning platform.

Protection and Confidentiality.

Facilitates Global Connection.

There Are No Ads.

Possibility for business.

DISADVANTAGES

A Means of Spreading False Information.

Leads to Issues with Mental Health.

Problems Associated with Health.

Sexual Stimulation.

No Content Censorship.

Financial Fraud.

APPLICATIONS

1. Contacts list: Users can add other users to their contacts list and see when they are online and available to chat.
2. Group chats: Users can create or join group chats with multiple users to communicate with them simultaneously.

3. Emojis and stickers: Users can use emojis and stickers to express themselves in their messages
4. End-to-end encryption: Some chat connection apps offer end-to-end encryption to ensure that messages and files are only visible to the sender and intended recipient.
5. Notifications: Users can receive notifications when they receive new messages or when someone joins or leaves a group chat.
6. Integration with other apps: Some chat connection apps integrate with other apps, such as social media platforms or productivity tools, to provide additional functionality.

Overall, the features of a chat connection app may vary depending on the specific app and its intended audience.

CONCLUSION

I'd suggest either expanding upon our chat app and add features like a user system where users have to authenticate, direct/private messages, chat rooms, and some sort of durable storage like a database to store things.

You could also make some sort of real time drawing applications where many users can draw on one canvas in real time. Try to flex those real time muscles! You could also rebuild the frontend in your framework of choice like React to see how your preferred frameworks interacts with a real time client.

FUTURE SCOPE

Chat connect is working on a bunch of features and it is expected to release them in the near future. The company is reportedly planning to add a new option for last seen, a new

disappearing chats feature and a redesigned group info page. Chat connect could also soon allow you to send high-resolution videos or images

1. User profiles: Users can create a profile that includes information about themselves, such as their name, profile picture, and status message.
2. Group chats: Users can create or join group chats with multiple users to communicate with them simultaneously.
3. Emojis and stickers: Users can use emojis and stickers to express themselves in their messages.
4. File sharing: Users can share files such as images, videos, and documents with each other.
5. Voice and video calls: Some chat connection apps also offer the ability to make voice and video calls with other users.
6. Notifications: Users can receive notifications when they receive new messages or when someone joins or leaves a group chat.

APPENDIX

A.Source Code

MainActivity.kt

```
package com.project.pradyotprakash.flashchat

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import com.google.firebase.FirebaseApp

/**
 * The initial point of the application from where it gets started.
 *
 * Here we do all the initialization and other things which will be required
 * thought out the application.
 */
class MainActivity : ComponentActivity() {
```

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    FirebaseApp.initializeApp(this)
    setContent {
        NavComposeApp()
    }
}
}

```

NavComposeApp.kt

```

package
com.project.pradyotprakash.flashchat

import androidx.compose.runtime.Composable
import androidx.compose.runtime.remember
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import com.google.firebase.auth.FirebaseAuth
import com.project.pradyotprakash.flashchat.nav.Action
import
com.project.pradyotprakash.flashchat.nav.Destination.AuthenticationOption
import
com.project.pradyotprakash.flashchat.nav.Destination.Home
import
com.project.pradyotprakash.flashchat.nav.Destination.Login
import
com.project.pradyotprakash.flashchat.nav.Destination.Register
er
import
com.project.pradyotprakash.flashchat.ui.theme.FlashChatTheme
e
import
com.project.pradyotprakash.flashchat.view.AuthenticationView
w
import
com.project.pradyotprakash.flashchat.view.home.HomeView
import
com.project.pradyotprakash.flashchat.view.login.LoginView
import
com.project.pradyotprakash.flashchat.view.register.Register
View
/**

```

```

* The main Navigation composable which will handle all the
navigation stack.
*/
@Composable
fun NavComposeApp() {
    val navController = rememberNavController()
    val actions = remember(navController) {
        Action(navController) }
    FlashChatTheme {
        NavHost(
            navController = navController,
            startDestination =
                if (FirebaseAuth.getInstance().currentUser != null)
                    Home
                Else
                    AuthenticationOption
        ) {
            composable(AuthenticationOption) {
                AuthenticationView(
                    register = actions.register,
                    login = actions.login
                )
            }
            composable(Register) {
                RegisterView(
                    home = actions.home,
                    back = actions.navigateBack
                )
            }
            composable(Login) {
                LoginView(
                    home = actions.home,
                    back = actions.navigateBack
                )
            }
            composable(Home) {
                HomeView()
            }
        }
    }
}

```

Navigation.kt

```
package com.project.pradyotprakash.flashchat.nav
```

```
import androidx.navigation.NavHostController
import com.project.pradyotprakash.flashchat.nav.Destination.Home
import com.project.pradyotprakash.flashchat.nav.Destination.Login
import com.project.pradyotprakash.flashchat.nav.Destination.Register
```

```
/**
 * A set of destination used in the whole application
 */
object Destination {
    const val AuthenticationOption = "authenticationOption"
    const val Register = "register"
    const val Login = "login"
    const val Home = "home"
}

/**
 * Set of routes which will be passed to different composable so that
 * the routes which are required can be taken.
 */
class Action(navController: NavHostController) {
    val home: () -> Unit = {
        navController.navigate(Home) {
            popUpTo(Login) {
                inclusive = true
            }
            popUpTo(Register) {
                inclusive = true
            }
        }
    }
    val login: () -> Unit = { navController.navigate(Login) }
    val register: () -> Unit = { navController.navigate(Register) }
    val navigateBack: () -> Unit = { navController.popBackStack() }
}
```