SEN2022 Software Engineering Analysis And Design

AI ASSISTANT

OVERVIEW

- Chat interface integrated with OpenAI API.
- Firebase Authentication / Realtime Database.
- User login, account management, chat history management.
- Demonstrates API integration, Firebase usage and mobile app development.

ASSUMPTIONS

Internet Access

API Availability

DEPENDENCIES

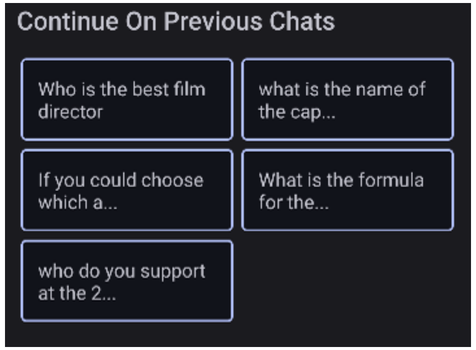
OpenAI API

Firebase Authentication Service

Firebase Realtime Database Service

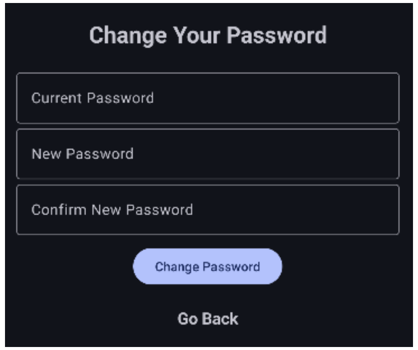
FUNCTIONAL REQUIREMENTS

- User Authentication
- Send & Receive messages from AI assistant
- Chat history & manipulation
- Account information & manipulation

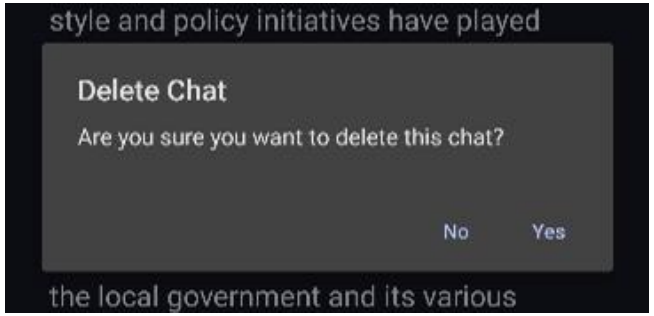


Continue On Previous Chats

Who is the best film director
what is the name of the cap...
If you could choose which a...
What is the formula for the...
who do you support at the 2...



Change Your Password



style and policy initiatives have played

Delete Chat

Are you sure you want to delete this chat?

the local government and its various

NON-FUNCTIONAL REQUIREMENTS

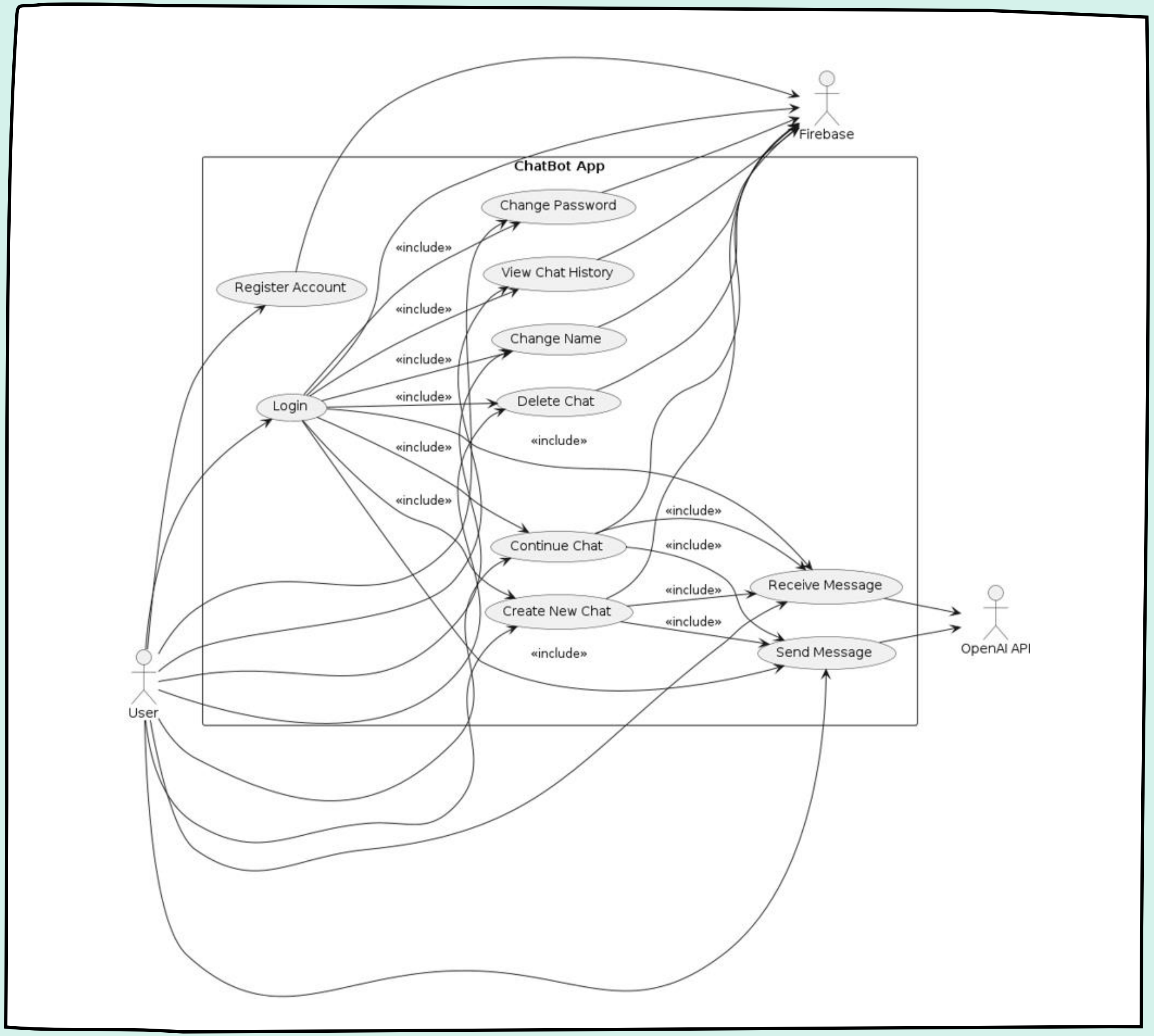
Responsive UI

Send & Receive messages with minimal delay

User data security

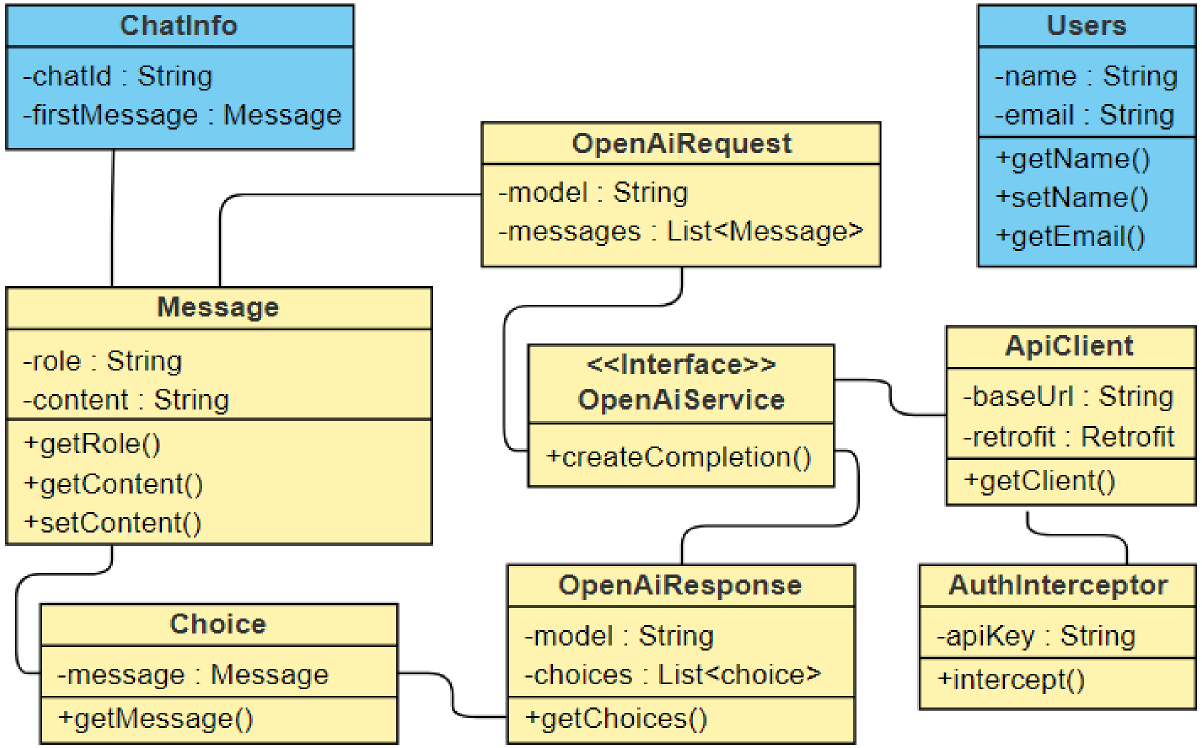
Reliable & Maintainable

Compatible



USE CASE DIAGRAM

CLASS DIAGRAM



ChatInfo
-chatId : String
-firstMessage : Message

Users
-name : String
-email : String
+getName()
+setName()
+getEmail()

OpenAiRequest
-model : String
-messages : List<Message>

Message
-role : String
-content : String
+getRole()
+getContent()
+setContent()

<<Interface>>
OpenAiService
+createCompletion()

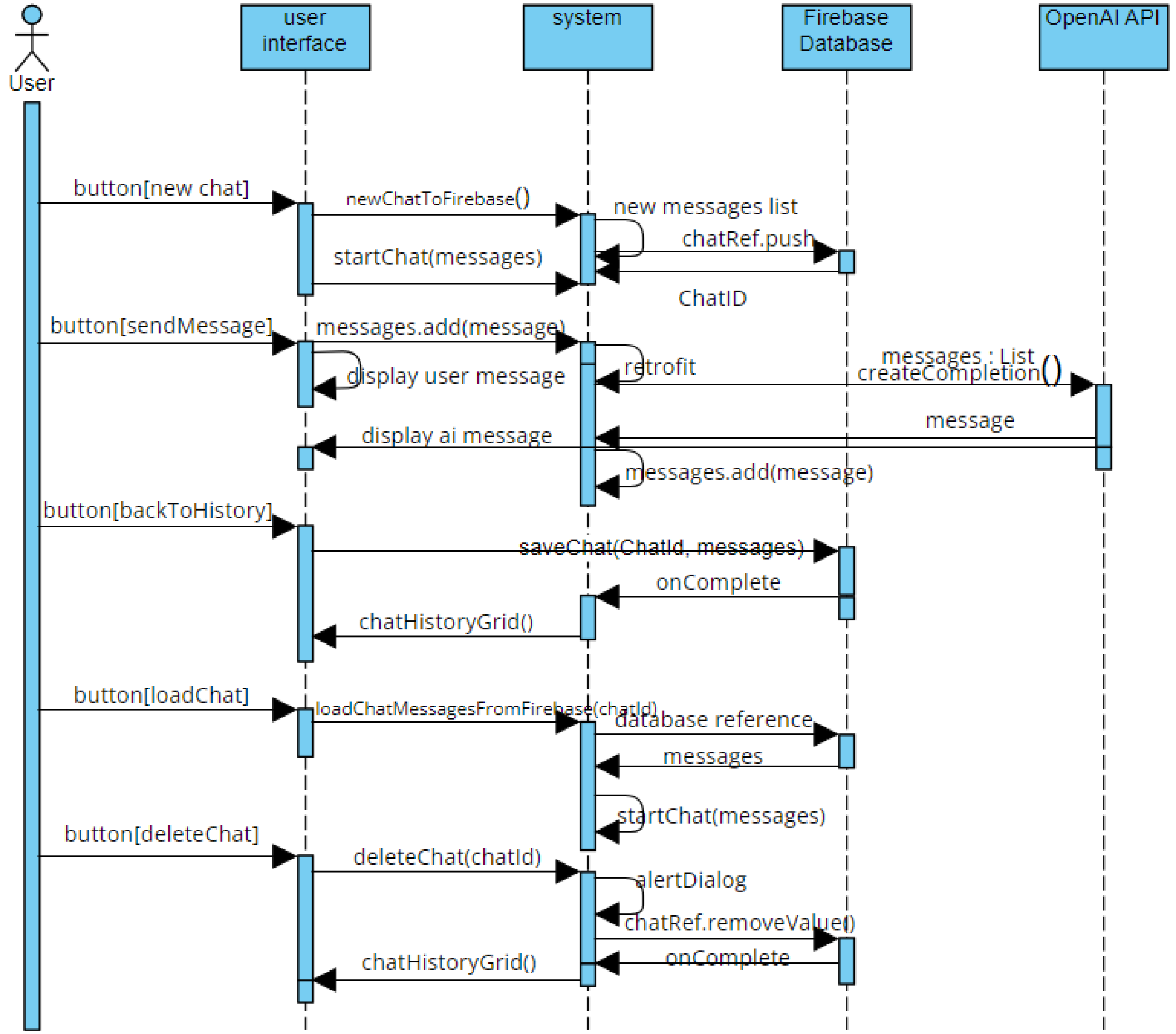
ApiClient
-baseUrl : String
-retrofit : Retrofit
+getClient()

Choice
-message : Message
+getMessage()

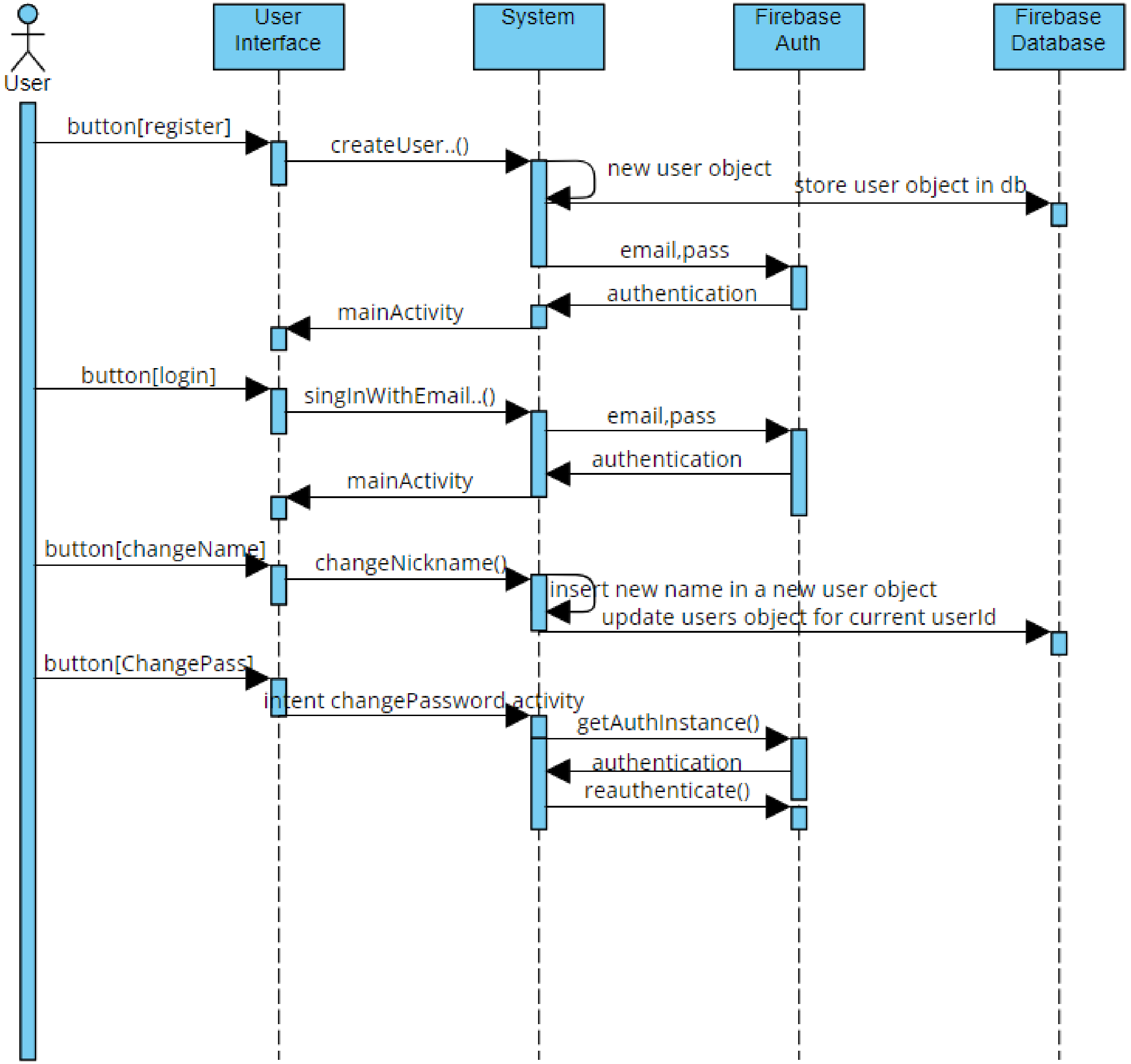
OpenAiResponse
-model : String
-choices : List<choice>
+getChoices()

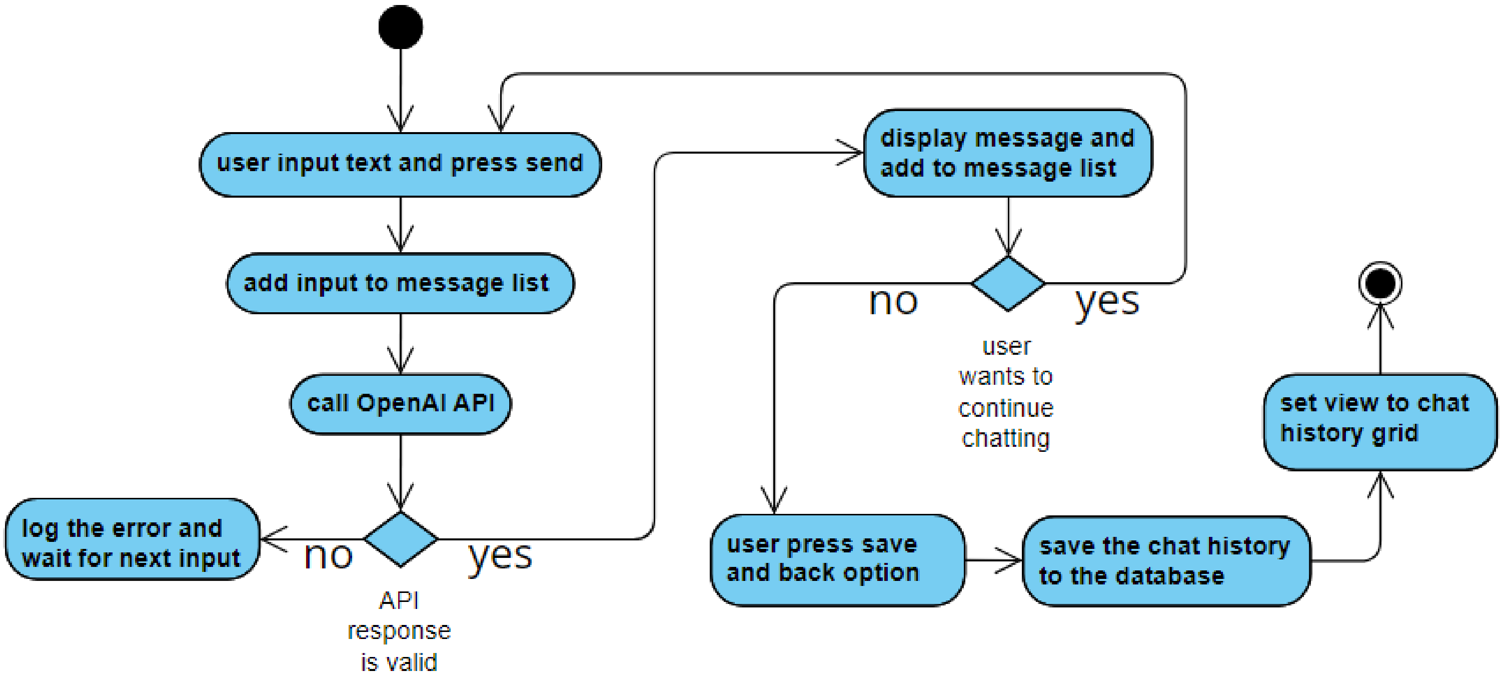
AuthInterceptor
-apiKey : String
+intercept()

CHAT SEQUENCE DIAGRAM



USER AUTHENTICATION SEQUENCE DIAGRAM





user input text and press send

add input to message list

call OpenAI API

API response is valid

no

log the error and wait for next input

yes

display message and add to message list

user wants to continue chatting

yes

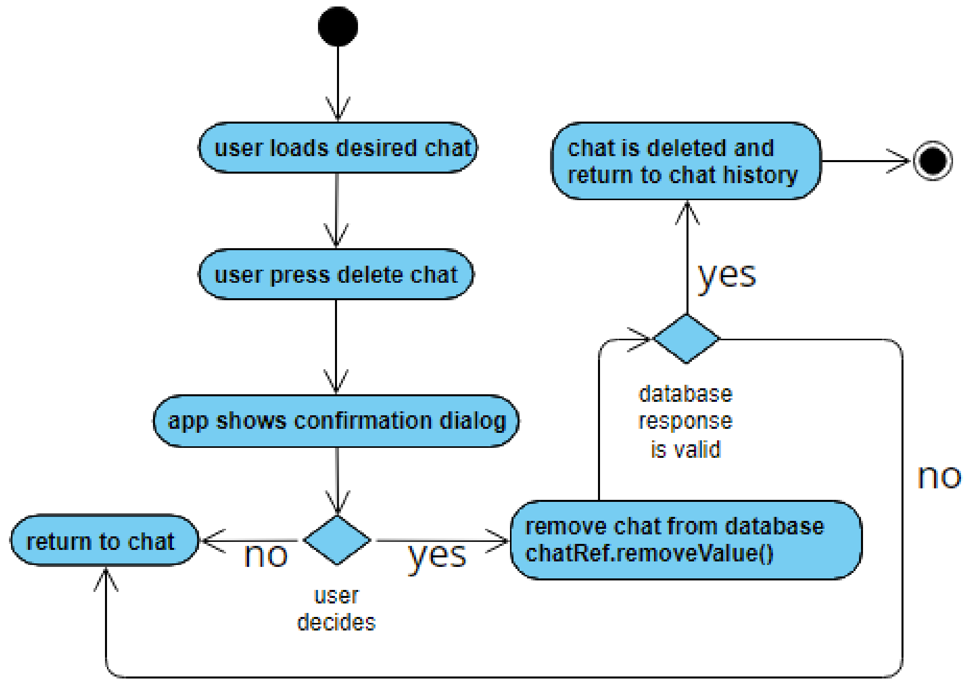
no

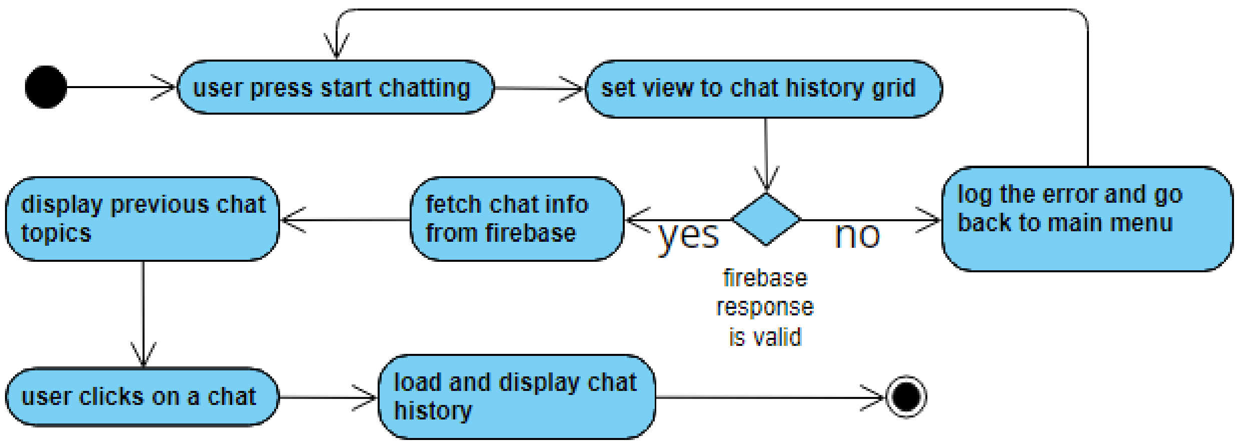
user press save and back option

save the chat history to the database

set view to chat history grid

SEND & RECEIVE MESSAGE ACTIVITY DIAGRAM

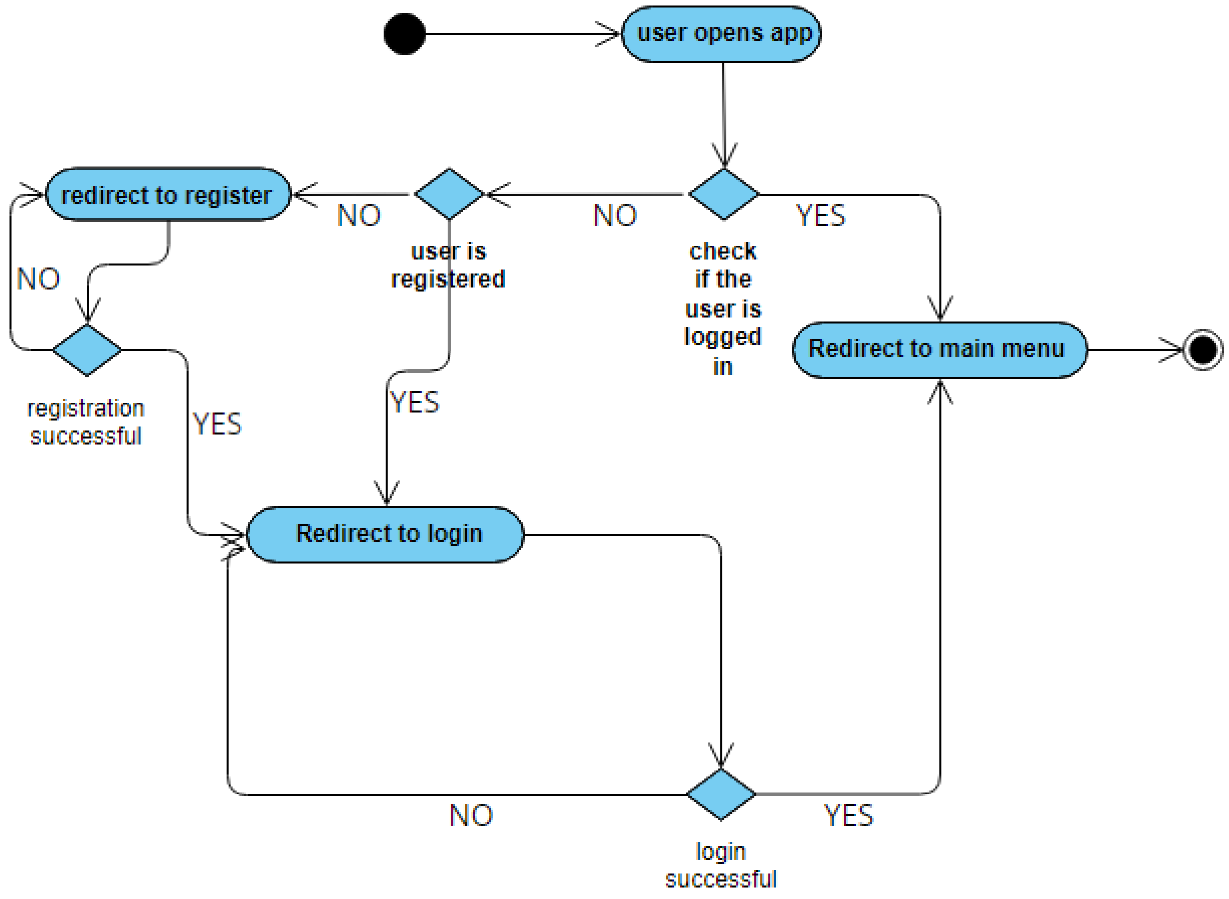

user loads desired chat
user press delete chat
app shows confirmation dialog
user decides
no
yes
return to chat
remove chat from database chatRef.removeValue()
database response is valid
yes
no
chat is deleted and return to chat history

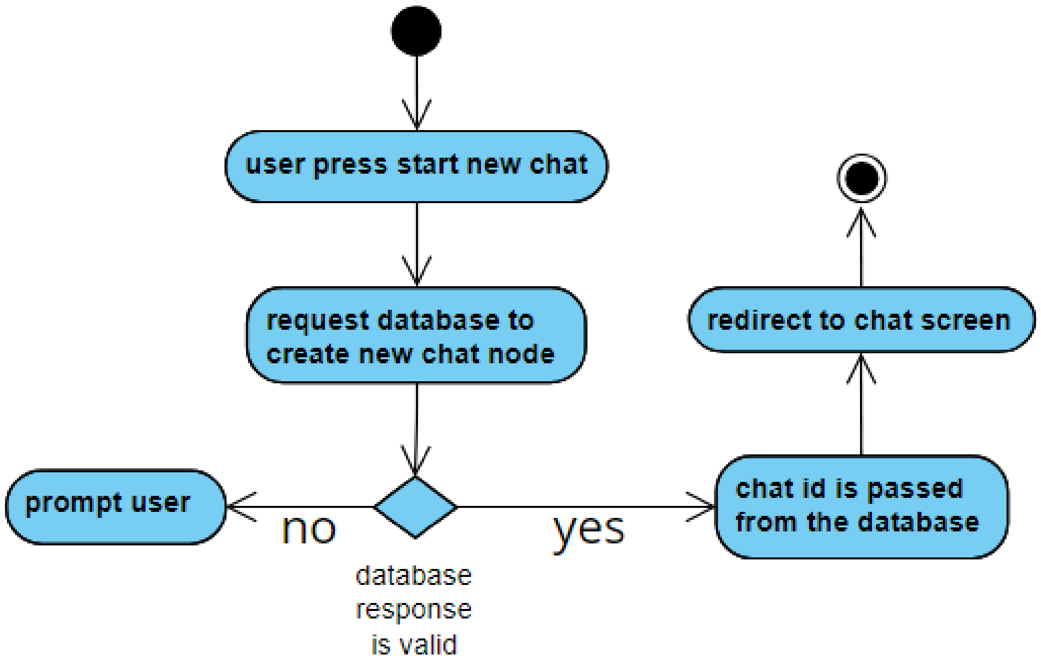

user press start chatting
set view to chat history grid
firebase response is valid
yes
no
log the error and go back to main menu
fetch chat info from firebase
display previous chat topics
user clicks on a chat
load and display chat history

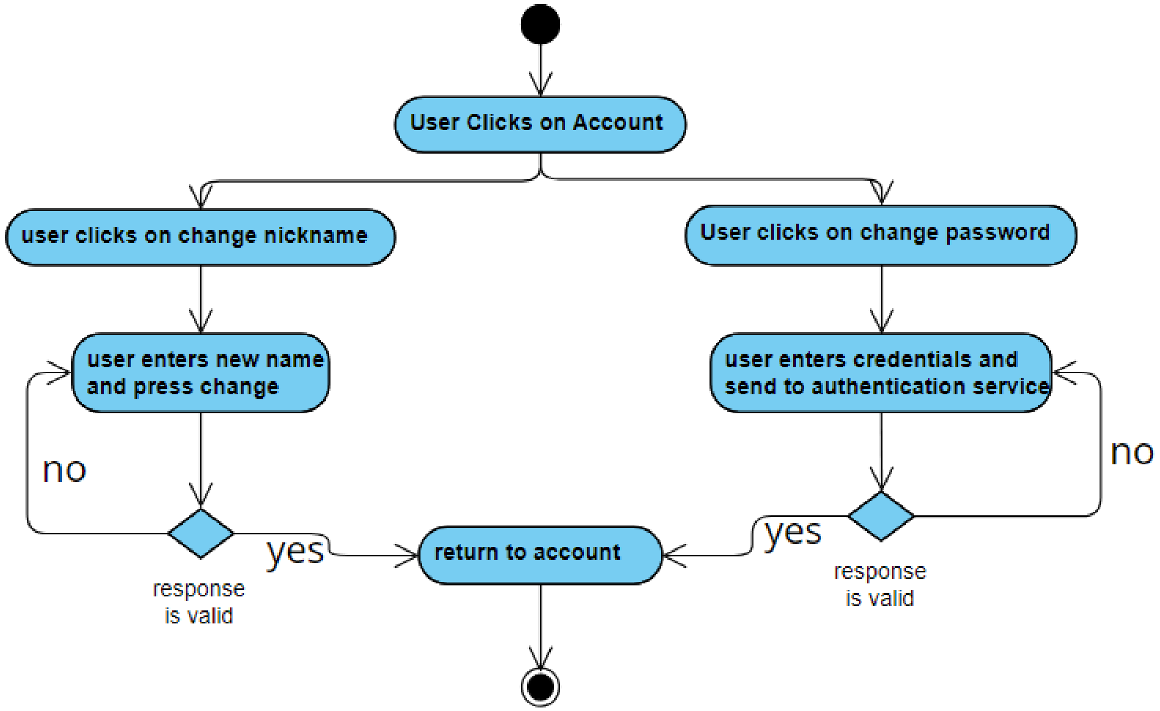
DELETE CHAT HISTORY ACTIVITY DIAGRAM

LOAD PREVIOUS CHAT ACTIVITY DIAGRAM

USER AUTHENTICATION ACTIVITY DIAGRAM

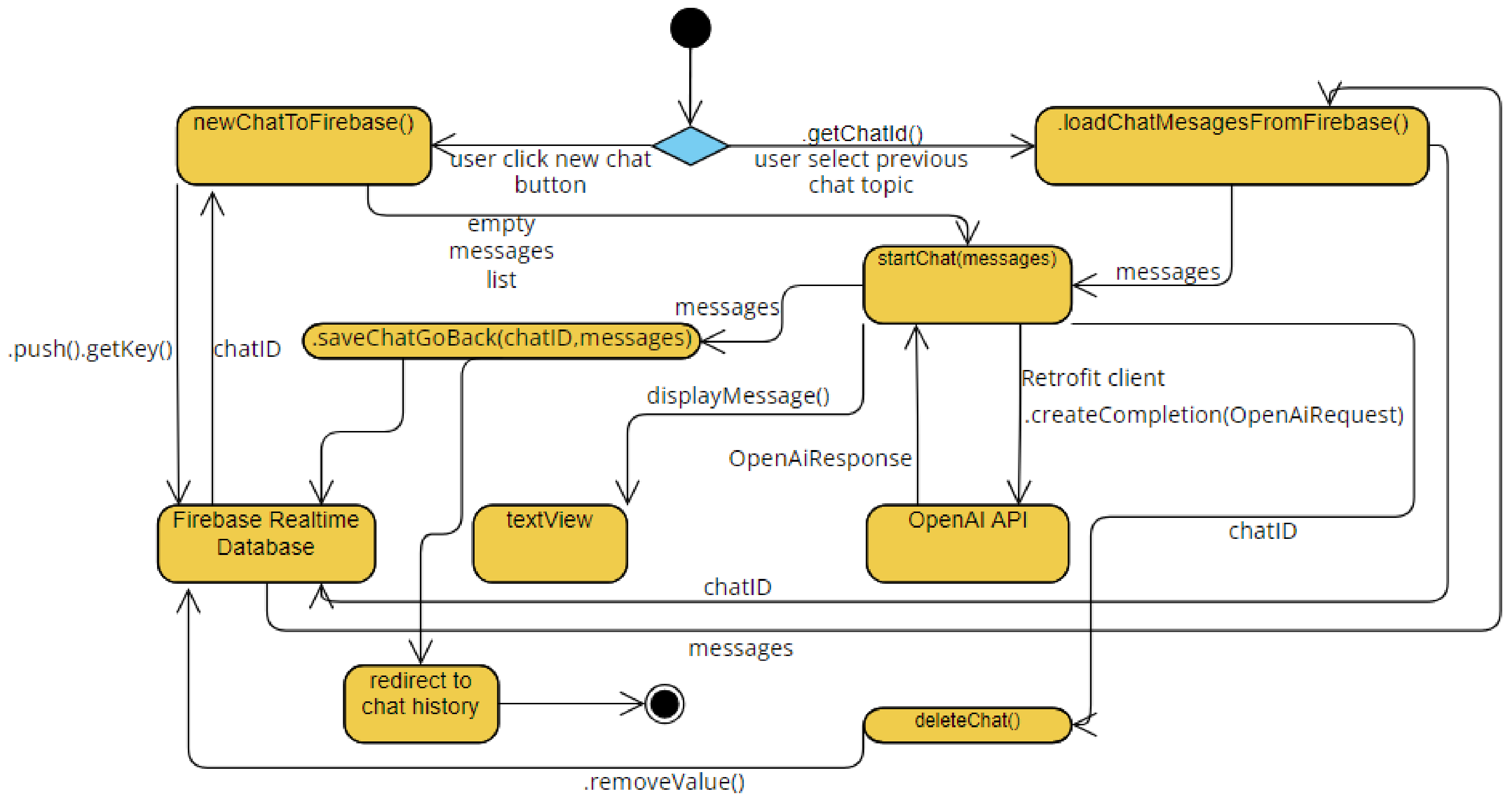



user press start new chat
request database to create new chat node
prompt user
no
database response is valid
yes
chat id is passed from the database
redirect to chat screen


User Clicks on Account
user clicks on change nickname
User clicks on change password
user enters new name and press change
user enters credentials and send to authentication service
no
response is valid
yes
return to account
yes
response is valid
no

START A NEW CHAT ACTIVITY DIAGRAM

CHANGE ACCOUNT CREDENTIALS ACTIVITY DIAGRAM



newChatToFirebase()

user click new chat button

.getChatId()
user select previous chat topic

.loadChatMesagesFromFirebase()

empty messages list

startChat(messages)

messages

messages

.saveChatGoBack(chatID,messages)

.push().getKey()

chatID

displayMessage()

Retrofit client .createCompletion(OpenAiRequest)

OpenAiResponse

Firebase Realtime Database

textView

OpenAI API

chatID

chatID

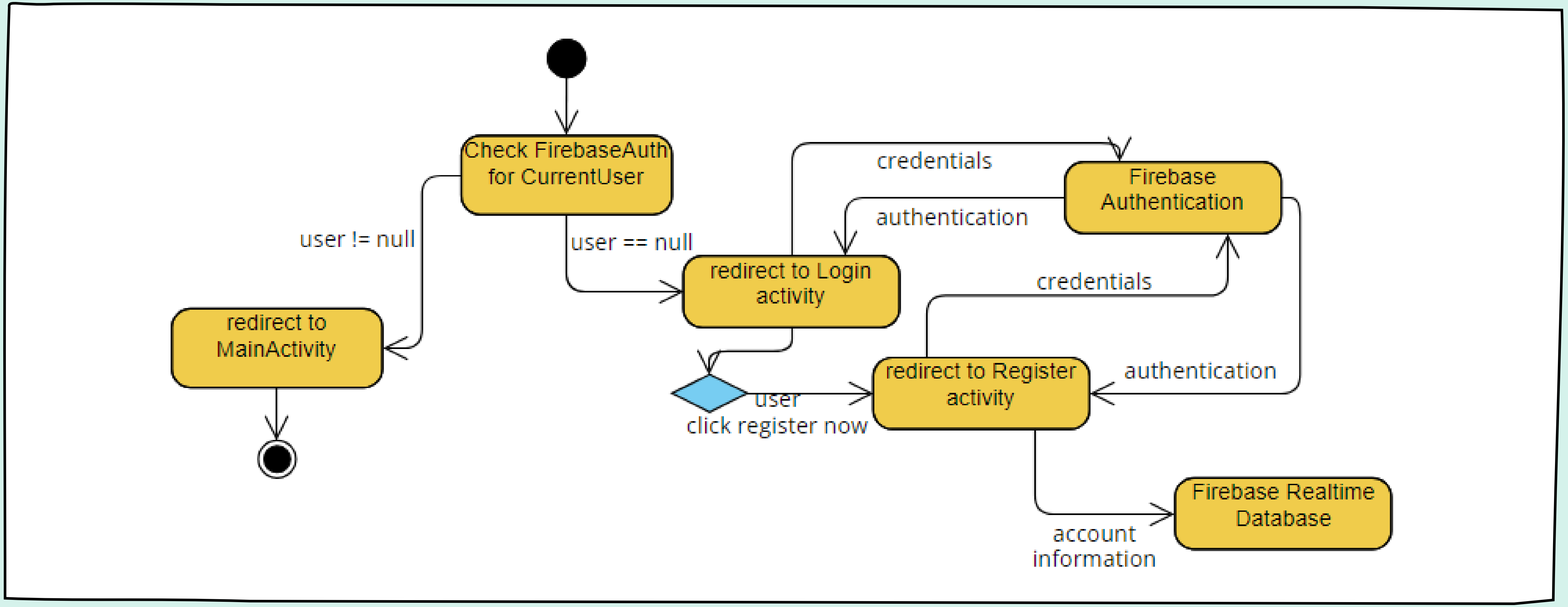
messages

redirect to chat history

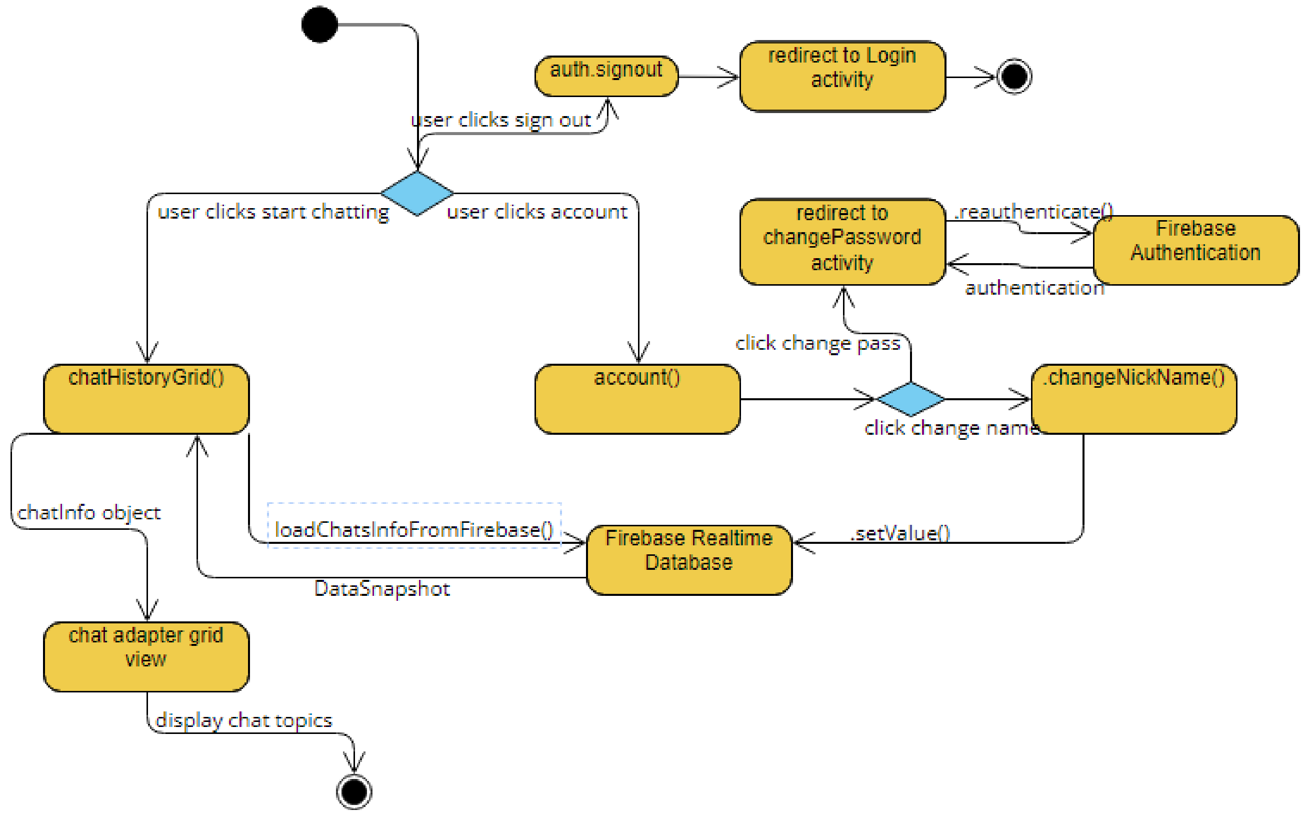
deleteChat()

.removeValue()

CHAT SCREEN STATE DIAGRAM



USER AUTHENTICATION STATE DIAGRAM



MAIN MENU AND ACCOUNT MANAGEMENT STATE DIAGRAM

PROJECT PLAN

- Agile
- Deliver Minimum Viable Product quickly
- Iterative development cycles
- Progressively added additional features & Refinement

ARCHITECTURE AND DESIGN

Client : User Interface, Display messages

Server: OpenAI API; core logic provider

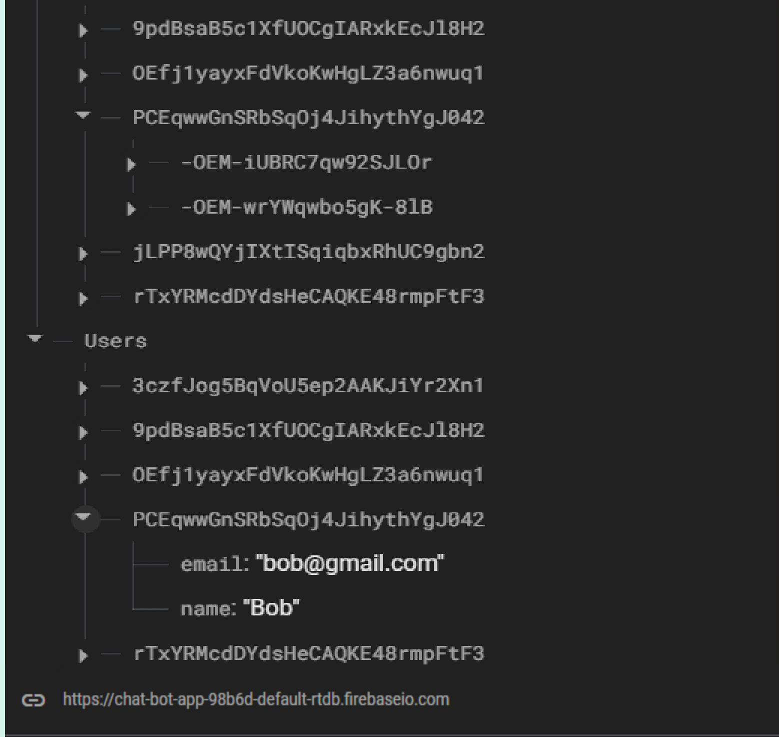
Database: Firebase Authentication & Realtime Database

DESIGN PATTERNS

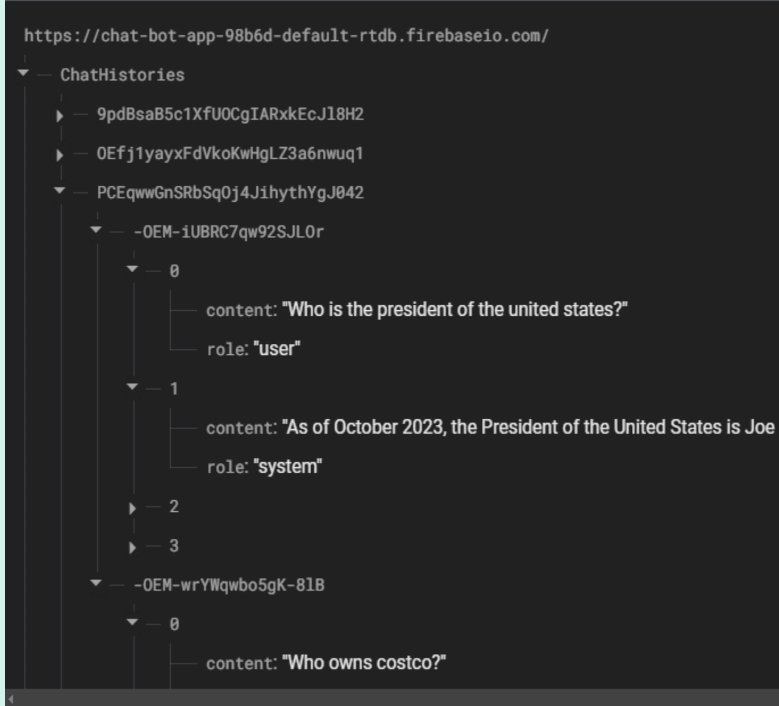
Model-View-Controller

Singleton Pattern: single instance of retrofit client

DATABASE DESIGN



```
▸ — 9pdBsaB5c1XfUOCgIARxkEcJl8H2
▸ — OEfj1yayxFdVkoKwHgLZ3a6nwuq1
▾ — PCEqwwGnSRbSqOj4JihythYgJ042
    ▸ — -OEM-iUBRC7qw92SJLOr
    ▸ — -OEM-wrYWqwbo5gK-81B
▸ — jLPP8wQYjIXtISqiqbxRhUC9gbn2
▸ — rTxYRMcdDYdsHeCAQKE48rmpFtF3
▾ — Users
  ▸ — 3czfJog5BqVoU5ep2AAKJiYr2Xn1
  ▸ — 9pdBsaB5c1XfUOCgIARxkEcJl8H2
  ▸ — OEfj1yayxFdVkoKwHgLZ3a6nwuq1
  ▾ — PCEqwwGnSRbSqOj4JihythYgJ042
        email: "bob@gmail.com"
        name: "Bob"
  ▸ — rTxYRMcdDYdsHeCAQKE48rmpFtF3
https://chat-bot-app-98b6d-default-rtdb.firebaseio.com
```



```
https://chat-bot-app-98b6d-default-rtdb.firebaseio.com/
▾ — ChatHistories
  ▸ — 9pdBsaB5c1XfUOCgIARxkEcJl8H2
  ▸ — OEfj1yayxFdVkoKwHgLZ3a6nwuq1
  ▾ — PCEqwwGnSRbSqOj4JihythYgJ042
    ▾ — -OEM-iUBRC7qw92SJLOr
      ▾ — 0
            content: "Who is the president of the united states?"
            role: "user"
      ▾ — 1
            content: "As of October 2023, the President of the United States is Joe
            role: "system"
      ▸ — 2
      ▸ — 3
    ▾ — -OEM-wrYWqwbo5gK-81B
      ▾ — 0
            content: "Who owns costco?"
```

FUNCTIONAL TESTING

- Check Whether Chat Histories Are Displayed Properly
- Check Whether the User Can Continue Chat Properly
- Check Whether the Save Works Properly
- Check Account Information functionality
- Check Whether User Authentication Works Properly
- Check Whether the Delete Chat Works Properly

NON-FUNCTIONAL TESTING

- Check Internet Connection Cut-off During Chat
- Stress Test with Rapid Chat Inputs
- Retrieve a Chat with More Than One Hundred Messages
- Check Whether Memory Usage During Prolonged Use Is Stable
- Test the App User Interface on Different Devices

FEASIBILITY

Widely available and reliable technologies: OpenAI API, Firebase

School project, development costs are the time investment

Implemented within a realistic timeline

Aligns with societal trends toward AI

Feasible