LLD OF CONNECTX
BACKEND

Name(s) : Neel Pritesh Khambhata (7167380), Nitish Bhardwaj (7201791)

Overview

ConnectX is an advanced automation software designed to enhance social media engagement for businesses. By utilizing cutting-edge AI technologies, ConnectX enables dynamic, personalized interactions with customers across various social media platforms. This platform empowers businesses to streamline their social media operations, improve customer engagement, and gain valuable insights through data analytics

Class Diagram

CLASSES

User

- Attributes:
  - + user Id: String
  - + name: String
  - + email: String
  - + password: String
  - + subscription Plan: Subscription
  - - Accounts: List<Account>
- Methods:
  - + signup (): void
  - + login (): Boolean
  - + view Dashboard (): void
  - + logout (): void

2. Account

- Attributes:
  - + accountId: String
  - +username:String
  - + linked Date: Date
  - + status: String
  - - posts: List<Post>
- Methods:
  - + logout (): void
  - + disconnect (): void
  - + fetch Posts (): List<Post>
  - + login (): Boolean

3. Dm Handler

- Attributes:
  - + automationId: String
  - + triggerKeyword: String
  - + responseMessage: String
- Methods:
  - + setAutomation(triggerKeyword: String, response: String): void
  - + triggerDM(message: String): boolean
  - + handleError(error: DMError): void

4. Subscription

- Attributes:
  - + plan_Id: String
  - + plan_name: String
  - + price: Double
  - + features: List<String>
- Methods:
  - + upgrade Plan (planId: String): void
  - + Cancel Subscription (): void

5.AIResponse

- Attributes:
  - + responseId: String
  - + contextData: String
  - - trainingData: List<String>
- Methods:
  - + generateResponse(message: String): String
  - + trainModel(data: List<String>): void

6. Analytics Dashboard

- Attributes:
  - + dashboard_id: String
- Methods:
  - + displayDashboard(): void

7. Notification Manager

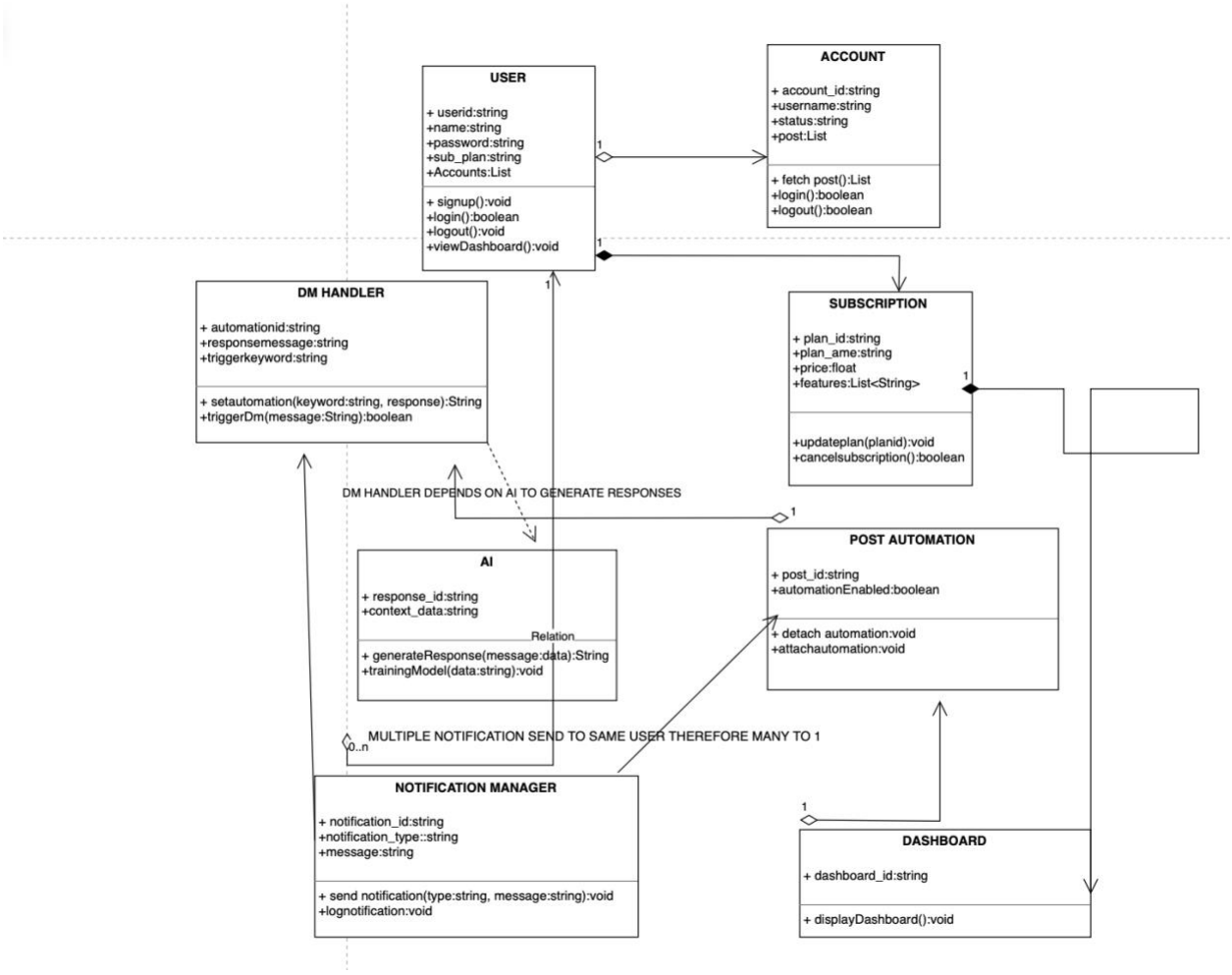
- Attributes:
  - + notification_Id: String
  - + type: String
  - + message: String
- Methods:
  - + sendNotification(type: String, message: String): void
  - + logNotification(): void

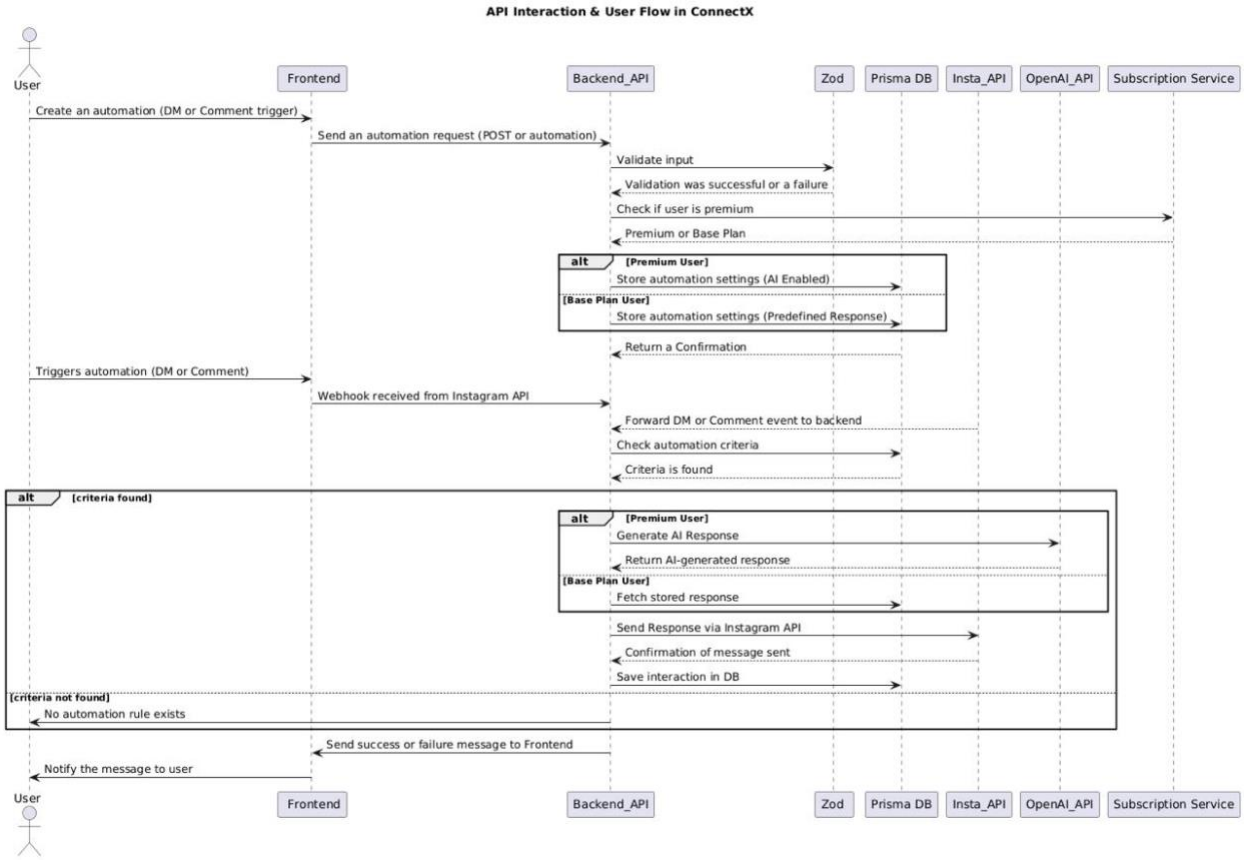
8. Post Automation

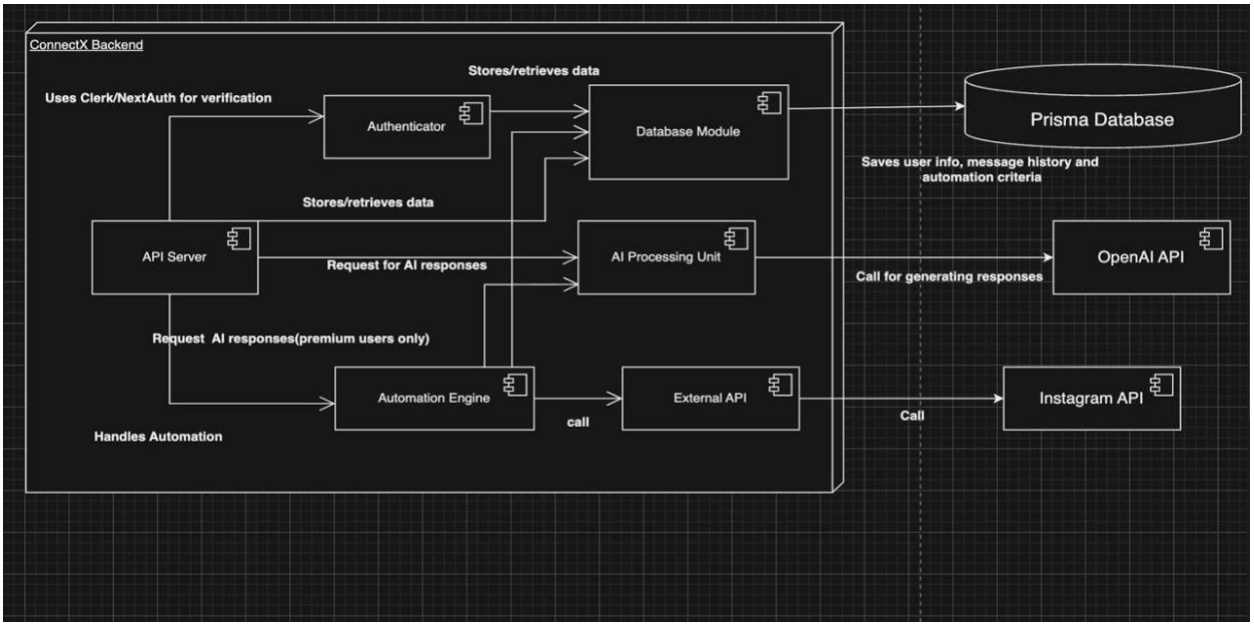
- Attributes:
  - + postId: String

    - + automationEnabled: boolean
    - + associatedTriggers: List<DMHandler>
- Methods:
    - + attachAutomation(): void
    - + detach Automation(): void

CLASS DIAGRAM




API Interaction & User Flow in ConnectX
User
Frontend
Backend_API
Zod
Prisma DB
Insta_API
OpenAI_API
Subscription Service
Create an automation (DM or Comment trigger)
Send an automation request (POST or automation)
Validate input
Validation was successful or a failure
Check if user is premium
Premium or Base Plan
alt
[Premium User]
Store automation settings (AI Enabled)
[Base Plan User]
Store automation settings (Predefined Response)
Return a Confirmation
Triggers automation (DM or Comment)
Webhook received from Instagram API
Forward DM or Comment event to backend
Check automation criteria
Criteria is found
alt
[criteria found]
alt
[Premium User]
Generate AI Response
Return AI-generated response
[Base Plan User]
Fetch stored response
Send Response via Instagram API
Confirmation of message sent
Save interaction in DB
[criteria not found]
No automation rule exists
Send success or failure message to Frontend
Notify the message to user
User
Frontend
Backend_API
Zod
Prisma DB
Insta_API
OpenAI_API
Subscription Service


ConnectX Backend
Stores/retrieves data
Uses Clerk/NextAuth for verification
Authenticator
Database Module
Prisma Database
Saves user info, message history and automation criteria
Stores/retrieves data
API Server
Request for AI responses
AI Processing Unit
Call for generating responses
OpenAI API
Request AI responses(premium users only)
Automation Engine
call
External API
Call
Instagram API
Handles Automation

STRATEGIES TO HANDLE ERRORS AND EXCEPTIONS

Exception Type	SCENARIO	HOW WE WILL HANDLE
Network error	API failure when connecting social accounts	Retry request with exponential backoff; Show user-friendly error message
Validation Error	Invalid user input during automation	Display inline validation messages
Authentic Error	Invalid login credentials	Show "Incorrect credentials" message
DataNotfound		No data available with responsive UI
Timeout Exception	Long running request to APIS	Notify user to try later

API ENDPOINTS

User Management

Endpoint	Method	Description	Request Body	Response	Error Codes
`/api/auth/login`	POST	Authenticates user	`{username, password }`	`{token }`	401, 500
`/api/auth/register`	POST	Registers a new user	`{name, email, password }`	`{ userId, token }`	400, 500
`/api/auth/logout`	POST	Logs user out	`{token }`	`{success: true }`	401, 500

Account Management

Endpoint	Method	Description	Request Body	Response	Error Codes
`/api/account/connect`	POST	Connects Instagram account	`{ accountType, accessToken }`	`{ success: true, accountId }`	400, 403, 500
`/api/account/disconnect`	DELETE	Disconnects an account	`{ accountId }`	`{ success: true }`	404, 500

Endpoint	Method	Description	Request Body	Response	Error Codes
`/api/account/details`	GET	Fetches account details	`{ accountId }`	`{ accountData }`	404, 500

DM Handler

Endpoint	Method	Description	Request Body	Response	Error Codes
`/api/dms/send`	POST	Sends an automated DM	`{ recipientId, message }`	`{ messageId }`	404, 500
`/api/dms/list`	GET	Fetches DM history	`{ accountId, dateRange }`	`{ messages }`	404, 500
`/api/dms/delete`	DELETE	Deletes a DM	`{ messageId }`	`{ success: true }`	404, 500

Subscription Management

`/api/subscription/create`	POST	Creates subscription plan	`{ userId, planType, paymentInfo }`	`{subscriptionId }`	400, 402, 500
`/api/subscription/cancel`	POST	Cancels subscription	`{ subscriptionId }`	`{ success: true }`	404, 500
`/api/subscription/status`	GET	Fetches subscription status	`{ userId }`	`{subscriptionDetails }`	404, 500

Post Automation

Endpoint	Method	Description	Request Body	Response	Error Codes
`/api/posts/schedule`	POST	Schedules a post	`{ accountId, content, scheduleTime }`	`{ postId }`	400, 500

`/api/posts/delete`	DELETE	Deletes a scheduled post	`{ postId }`	`{ success: true }`	404, 500
`/api/posts/list`	GET	Fetches scheduled posts	`{ accountId }`	`{ posts }`	400, 500

AI Response Manager

Endpoint	Method	Description	Request Body	Response	Error Codes
`/api/ai/generate-response`	POST	Generates DM response	`{ messageContext }`	`{ aiResponse }`	400, 500
`/api/ai/feedback`	POST	Sends feedback to AI	`{ responseId, feedback }`	`{ success: true }`	400, 500