MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Chatbot AI Platform (CAIP)

SEP490_G47	
Group Members	Ngô Minh Đức - Leader - HE173177 Phạm Viết Long - Member - HE170788 Trần Anh Đức - Member - HE176711 Hoàng Mai Dung - Member - HE176572 Ngô Minh Tuấn Anh - Member - HE176111
Supervisor	Nguyễn Trung Kiên - Lecturer

- Hanoi, Jan 2025 -

Table of Contents

Acknowledgement

ChatBees was a significant project that most team members hadn't experienced before, and we initially doubted whether we could complete it. Looking back on our journey from the beginning, we've received tremendous support from many people. While the rest of this document may be technical, we believe it's important to take a moment to express our sincere gratitude.

First and foremost, we extend our deepest thanks to our team supervisor, Mr. Nguyen Trung Kien. Not only is he highly skilled in design and software expertise, but he is also a tremendous motivator, constantly encouraging us to push forward and never give up during this challenging journey.

We also want to express our gratitude to the dedicated lecturers at FPT University. Throughout this semester and our Capstone Project, their guidance and support have been invaluable. Their lessons have not only helped shape us into capable professionals but have also given us the confidence to pursue our ambitions.

ChatBees is just the beginning of our long journey in software engineering. Nevertheless, we have every right to be proud of the resilience of our team and the invaluable knowledge and memories we've gained along the way.

ChatBees Project

Hanoi, April 29th, 2025

Definition and Acronyms

Acronym	Definition
AWS	Amazon Web Services
BA	Business Analysis
BR	Business Rule
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
PM	Project Manager
SDD	Software Design Description
SPMP	Software Project Management Plan
SRS	Software Requirement Specification
UC	Use Case
API	Application Program Interface

I. Project Introduction

1. Overview

1.1 Project Information

- Project name: ***Chatbot AI Platform***
- Vietnamese name: ***Nền tảng AI giao tiếp***
- Project code: ***CAIP***
- Group name: ***SEP490_G47***
- Software type: ***Web App***

1.2 Project Team

Full Name	Role	Email	Mobile
Nguyễn Trung Kiên	Lecturer	kiennt@fpt.edu.vn	0912345678
Ngô Minh Đức	Leader	ducnmhe173177@fpt.edu.vn	0355761159
Phạm Viết Long	Member	longpvhe170788@fpt.edu.vn	0368353613
Trần Anh Đức	Member	ductahe176711@fpt.edu.vn	0918478364
Hoàng Mai Dung	Member	dunghmhe176572@fpt.edu.vn	0834235369
Ngô Minh Tuấn Anh	Member	anhnmthe176111@fpt.edu.vn	0708328518

Table 1.2.1. Project Team

2. Product Background

In today's fast-paced business environment, both employees and customers struggle to access the right information quickly. Customer service reps waste time sifting through disjointed knowledge bases or escalating queries due to incomplete data, leading to frustrated customers and missed opportunities. Similarly, employees across departments, from sales to HR to engineering, face challenges in retrieving actionable insights, hampering productivity and decision-making.

Traditional chatbots, while popular, are limited to simple, predefined responses and fail to handle complex, context-rich queries. As businesses evolve, these legacy systems can't keep up, creating a demand for smarter solutions.

Enter the Conversational AI Platform, powered by Retrieval-Augmented Generation (RAG). This innovative technology combines advanced information retrieval with generative AI to deliver accurate, real-time responses to nuanced queries. Unlike static chatbots, RAG dynamically pulls from vast data repositories, ensuring up-to-date and contextually relevant answers.

In Vietnam's rapidly digitizing economy, this platform is a game-changer. For e-commerce, it handles surges in customer inquiries during peak times like Tet, providing instant answers on product availability and promotions. In banking, it simplifies complex transactions and reduces call center reliance. Tourism benefits from personalized, multilingual support, while healthcare sees improved patient interactions and faster access to medical information.

By bridging the gap between fragmented data and actionable insights, this platform enhances efficiency, customer satisfaction, and competitiveness in Vietnam's dynamic market.

3. Business Opportunity

Businesses worldwide, including in Vietnam, face growing customer service challenges: high costs, slow responses, and uneven quality. Our Conversational AI Platform solves these problems, tapping into a global need and a unique opportunity in Vietnam's fast-digitizing market.

Real-World Problems and Cost Burdens

From multinational corporations to local enterprises, the traditional customer service model is under strain:

High Operating Costs: Globally, customer service labor costs are soaring, think $30,000 USD annually per agent in Western markets. In Vietnam, salaries range from 7-15 million VND/month ($280-600 USD), depending on experience. (ref: vietnamworks)

Add training costs of up to $1,000 USD per employee, and a team of 50 in Vietnam could rack up billions of VND yearly, a burden felt worldwide, but especially by Vietnam's small and medium enterprises. (ref: shrm)

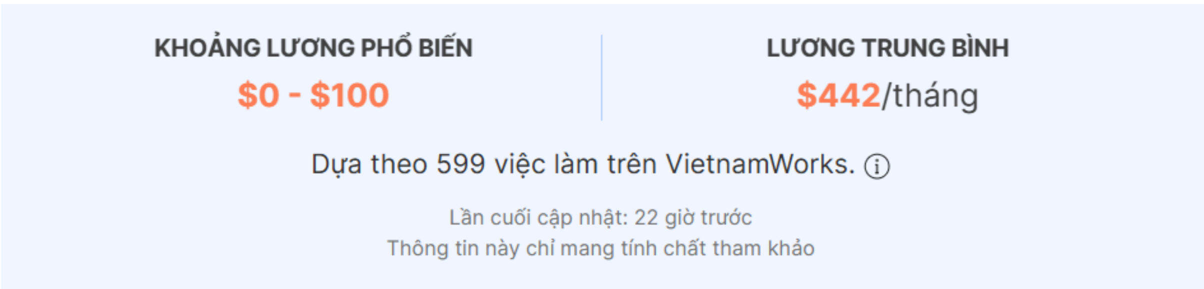


Figure 3.1. Common salary range and average pay

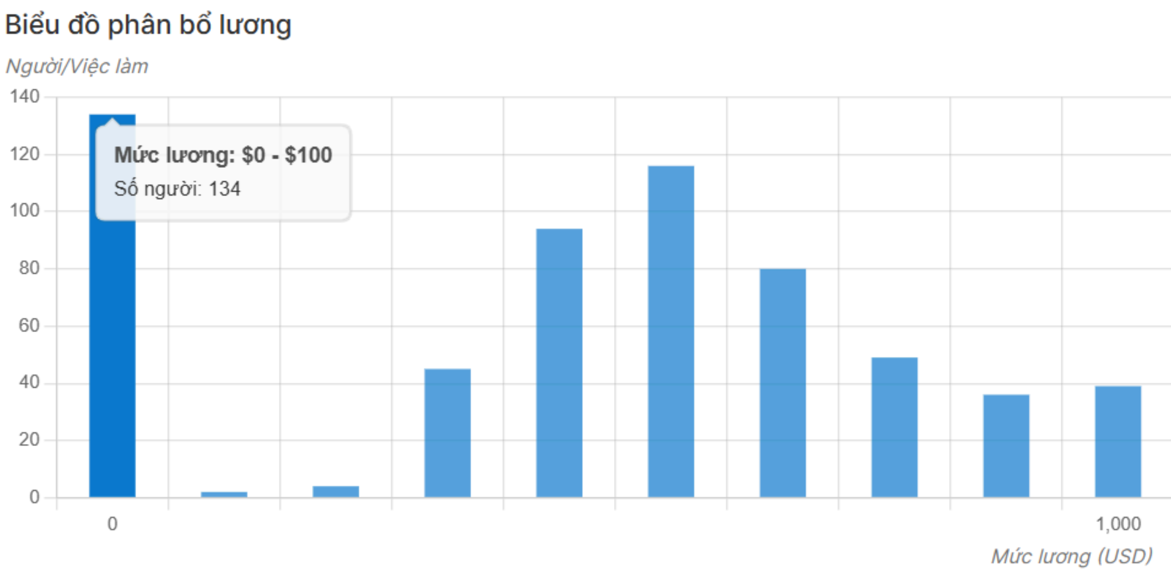


Figure 3.2. Salary Distribution Chart from VietnamWorks

Figures 1 and 2 chart highlights Vietnam's salary spread, underscoring the financial pressure on local businesses.

Slow Responses, Especially During Peak Times: Worldwide, customers demand speed; 70% expect replies within an hour. In Vietnam, this tightens to 70% within 30 minutes, yet during Tet, wait times stretch into hours, driving customers away and slashing revenue a universal peak-season struggle, amplified by Vietnam's cultural calendar. (ref: bell24vietnam)

Inconsistent Quality: Globally, human variability frustrates customers; in Vietnam, this inconsistency erodes trust in brands, a gap traditional chatbots can't fill in either market.

These shared pain points signal a global need and a Vietnamese opportunity for transformation.

Solutions from Our Conversational AI Platform

Our platform delivers a breakthrough with worldwide appeal, fine-tuned for Vietnam:

- Significant Cost Reduction: Globally, it slashes interaction costs from $2-3 with humans to $0.01, saving millions annually a win for any economy. In Vietnam, this translates to tens of millions of VND saved, irresistible for cost-conscious firms.
- Instant 24/7 Responses: Meeting global speed demands, it responds instantly, aligning with Vietnam's 30-minute expectation for 70% of customers, especially during Tet, retaining clients and boosting sales. (ref: bell24vietnam)
- Consistent Quality: Standardized answers ensure professionalism everywhere, eliminating the variability plaguing both global and Vietnamese businesses.
- Picture a customer anywhere asking about an order at midnight, or a Vietnamese shopper during Tet, both getting instant replies, that's our game-changer.

Why Our Product Will Succeed

The opportunity spans the globe, with Vietnam as a standout:

- Ready Market: The global conversational AI market is booming projected to hit $32 billion USD by 2030 driven by e-commerce and banking. In Vietnam, these sectors grow 20% yearly, with firms like Vietcombank leveraging chatbots for millions of queries, proving local demand.
- Competitive Edge: Against pricey OpenAI or payment-limited Coze, we're cheaper, faster, and globally scalable, with Vietnam-specific optimizations giving us an edge.
- Expansion Potential: From Vietnam to Southeast Asia and beyond, our platform rides a worldwide automation wave.

Cost and Benefit Comparison Table

Here's how our platform stacks up against human staff, highlighting its value globally and in Vietnam:

Factor	Human Staff Service	Conversational AI Platform
Cost per Interaction	$2-3 (global avg.)	$0.01

Response Time	Minutes to hours, peak delays	Instant, 24/7
Quality Consistency	Inconsistent	Standardized, uniform
Training Cost	$1,000 USD/employee (global)	None, cost-saving
Scalability	Limited, hiring bottlenecks	High, no staff dependency

Table 3.1. Cost and Benefit Comparison

4. Existing Solutions

4.1 N8n (n8n)

N8n is an open-source workflow automation platform that allows users to connect applications and automate tasks easily. It offers a flexible drag-and-drop interface, supports custom scripting for advanced logic, and can be fully self-hosted, giving users complete control over their data. With its event-driven architecture and broad integration capabilities, n8n helps streamline and scale complex automation processes efficiently.

Figure 4.1.1. ChatGPT

Highlight	Disadvantages
• Provides a highly flexible automation platform capable of connecting a wide range of applications and services. • Open-source and self-hostable, giving	• Requires users to manually build and configure workflows, which can be time-consuming and technically demanding.

businesses full control over their workflows and data privacy. ● Supports complex logic, data transformation, and advanced scripting for customized automation scenarios. ● Visual workflow builder enables users to design tailored processes according to their needs.	● As workflows grow in complexity, managing and troubleshooting them can become challenging, especially for non-technical users. ● Hosting, maintaining, and scaling n8n can incur high operational costs, including infrastructure expenses and the need for technical personnel, making it less accessible for small businesses.

Table 4.1. Highlight and disadvantages of n8n

4.2 Coze(Coze)

Coze is a next-generation AI application development platform that enables users to build conversational agents without the need for coding. It offers a user-friendly interface with drag-and-drop configurations, allowing for the creation of complex conversational flows and decision trees. Coze supports integration with various data sources, enabling chatbots to interact intelligently with real-time information.

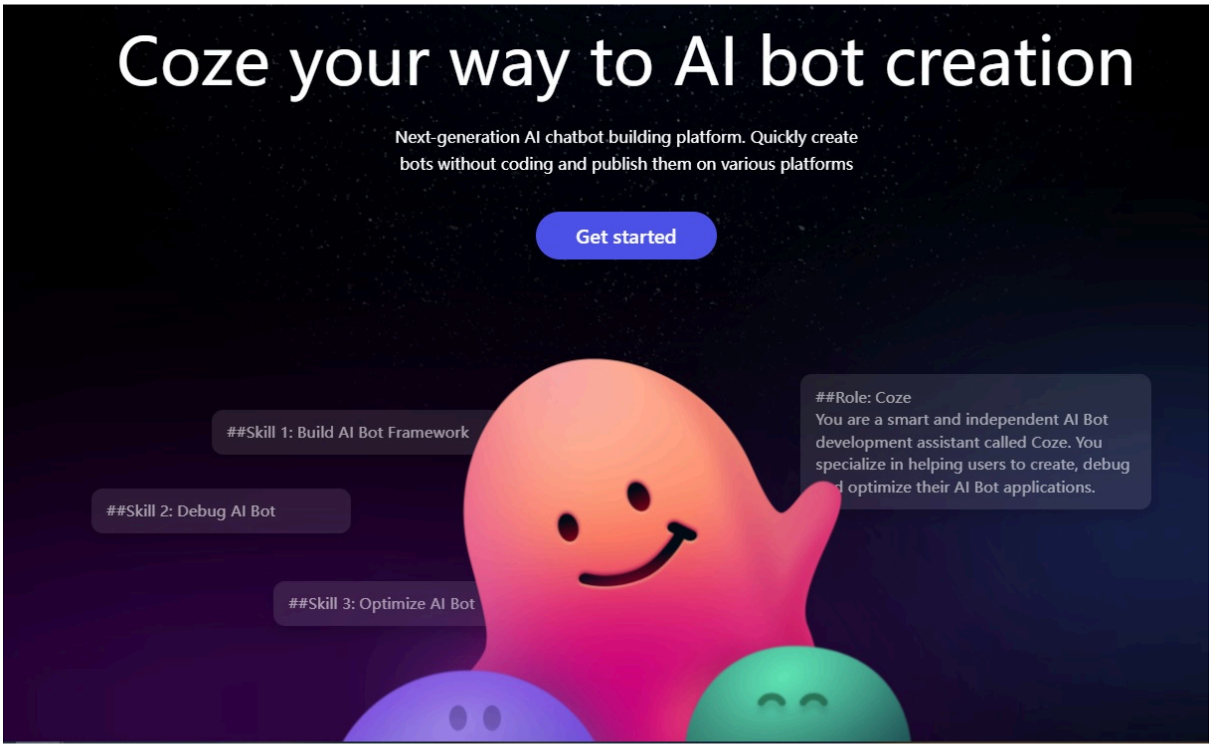


Figure 4.2.1. Coze

Highlight	Disadvantages

● Enables personalized chatbot experiences with excellent UX/UI. ● Allows users to easily create custom conversation flows tailored to their needs. ● Provides a no-code platform for building conversational agents, making development more accessible.	● The payment system can be inconvenient in Vietnam, as it often requires a Visa card or a credit card. ● Lacks a centralized live chat management system, making it difficult to monitor and control all active chat sessions effectively. ● Bots are unable to autonomously assess conversation quality or notify human agents when intervention is needed (e.g., incorrect responses, sales closing opportunities), hindering human-in-the-loop workflows. ● Despite offering many features, the platform can be overwhelming, requiring users to spend considerable time learning, integrating, and navigating the documentation before achieving optimal use. ● Integration challenges and documentation complexity can slow down deployment and user adoption, especially for non-technical teams.

Table 4.2.1. Highlight and disadvantages of Coze

4.3 Chatling(Chatling.ai)

Chatling.ai is an emerging conversational AI platform that emphasizes simplicity and adaptability. Designed to serve both small businesses and large enterprises, it offers tools to build chatbots that are intuitive, scalable, and capable of handling complex interactions. Chatling.ai supports seamless integration with external knowledge bases and third-party applications, making it a versatile choice across industries.

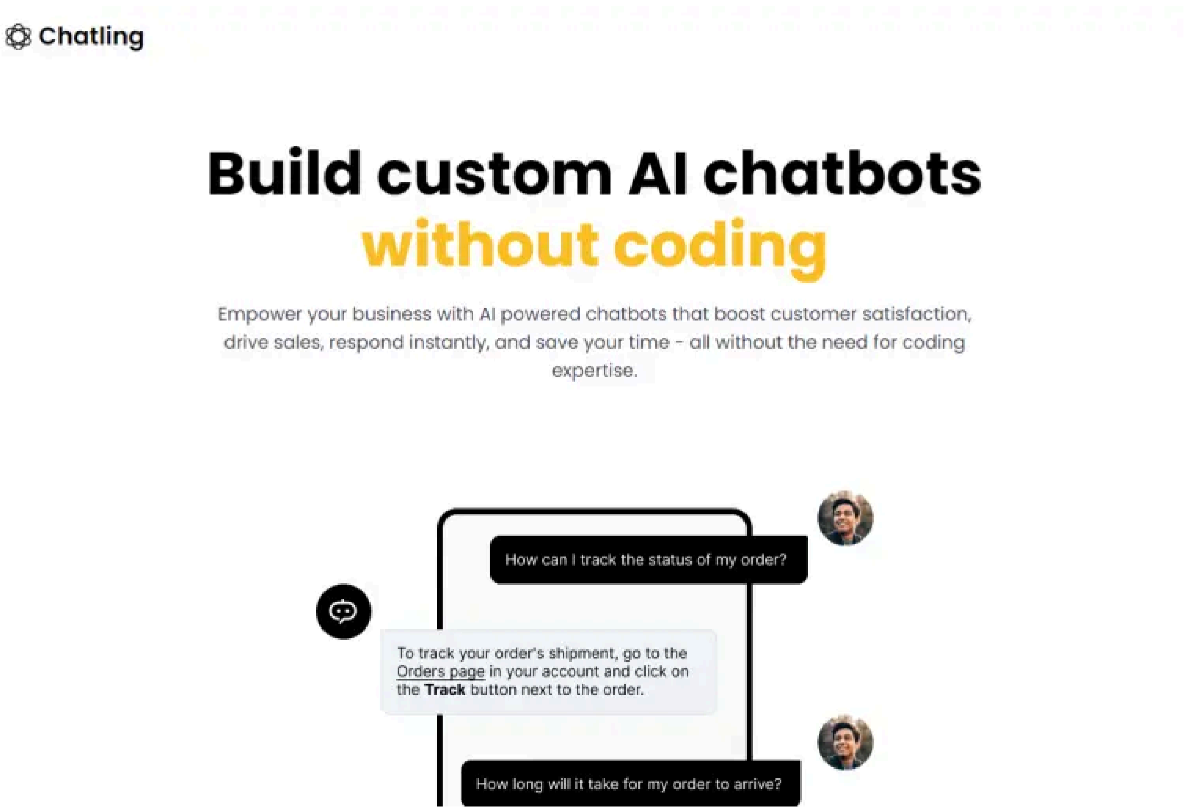


Figure 4.3.1. Chatling ai

Highlight	Disadvantages
• Provides seamless integration of chatbots on different websites. • Allows fine-tuning to enhance model accuracy and response relevance.	• It may not be cost-effective or perform optimally for large language models (LLMs), consuming significant resources. • Tends to deliver slower responses even for simpler queries. • Lacks customizable permission settings, reducing operational flexibility and potentially limiting scalability for enterprise needs.

Table 4.3.1. Highlight and disadvantages of Chatling ai

5. Evaluation of Existing Solutions

Target Market:

- Small and Medium Enterprises (SMEs) in Vietnam: Seeking affordable AI solutions to improve customer service and simplify operations.

- Key Industries: Banking, e-commerce, online education, and healthcare, where automation and accurate information retrieval are essential.
- Non-Technical Users: Businesses wanting to use AI without needing advanced skills.
- Technical Users: Developers and IT teams looking to customize bots for higher accuracy and closely monitor response workflows.

Value Proposition: The Conversational AI Platform is purpose-built for the Vietnamese market, addressing critical gaps left by solutions like OpenAI, Coze, and Chatling.ai. It cuts costs and provides context-aware answers using your data, unlike OpenAI's pricey, generic responses and tough integration. It offers seamless QR code payments and scalability, bypassing Coze's local payment and growth issues. And it ensures fast replies and easy customization, fixing Chatling.ai's slow responses and rigid setup.

- Superior Information Retrieval: Tailored to Vietnamese context, providing accurate, locally-relevant responses that reduce costs and improve operational efficiency.
- Seamless Payment Integration: Supports QR codes, fitting Vietnam's e-wallet culture for easy monetization.
- Effortless Customization: Non-technical users can adjust AI responses without coding, while technical users can fine-tune bots for better accuracy and track response workflows step-by-step.
- Dual Operation Modes: Simple for users, advanced for admins, ensuring maximum usability for all skill levels.
- New Advantage (Predicted): Multilingual Support for Regional Dialects: Reaches diverse Vietnamese users, increasing adoption in rural areas.

Competitive Edge & Market Potential: By specifically addressing unmet needs in the Vietnamese SME market, our platform delivers clear advantages for both technical and non-technical users:

- Cost-Effective: Uses task-specific AI to lower expenses, unlike OpenAI's high-cost model.
- Plug-and-Play Scalability: Easy to deploy and integrate, unlike Coze's scalability issues, fitting businesses of all sizes.
- Always Improving: Automated quality checks ensure top performance, addressing Chatling.ai's slow response problem.
- Local Fit: Built for Vietnam with local language and payment support, solving integration challenges faced by others.
- Booming AI Market: Vietnam's digital transformation creates high demand, especially in key sectors like banking and e-commerce.
- Automation Imperative: Businesses need customer service automation to stay competitive, and we deliver it better.
- Regional Growth: Scalable design ready for expansion across Southeast Asia.
- New Advantage (Predicted): Cultural Context Awareness: Understands Vietnamese nuances (e.g., holiday greetings), enhancing user engagement.

6. Software Product Vision

The Conversational AI Platform, powered by Retrieval-Augmented Generation (RAG), equips businesses with intelligent AI Agents that deliver real-time, accurate, and context-rich responses, surpassing traditional chatbots. It empowers companies in Vietnam's fast-digitalizing market to enhance engagement, streamline operations, and drive growth with unmatched efficiency.

Handles All Kinds of Data, Always Up-to-Date: From internal files you upload to live external data like market updates or product feeds, our platform processes it all seamlessly and in real-time. Your bot stays sharp with the latest info think instant stock updates during Tet sales or real-time banking policies for a customer's query.

Faster Responses, Lower Costs: We've fine-tuned the system to retrieve and reply at lightning speed while keeping resource use lean. Compared to traditional setups, it slashes wait times and expenses, letting you handle more queries without breaking the bank. For instance, sales teams can pull product details in seconds, no more digging through manuals, speeding up deals, and cutting down on staff workload.

Less Manpower, More Impact: By automating data lookups and customer interactions, the platform reduces the need for human intervention in repetitive tasks. This means fewer staff hours spent on grunt work, lower costs, and consistent answers every time. Your team can focus on what matters, like closing deals, while customers get quick, reliable service.

Personalization Without the Hassle: Customize your bot's vibe to match your brand, no tech skills needed. Want a friendly tone for your e-commerce shop or a pro voice for finance clients? You decide how it speaks, making every interaction feel tailored and on-point.

This platform saves time, lowers costs, and improves productivity by providing quick and accurate answers from any data source. It grows easily across industries, reduces the need for staff, and customizes interactions, helping businesses in Vietnam innovate, keep customers happy, and stay ahead in a digital world.

7. Project Scope & Limitations

Our Conversational AI Platform aims to deliver a smart, adaptive web application for businesses, hosted on the cloud for maximum scalability and accessibility. The intuitive interface ensures a smooth experience for all users.

7.1 Major Features

Figure 7.1.1. Major Features Tree

Feature ID	**Feature and Description**
	End-user
FE-01	Conversational Interaction (Chat AI): A real-time AI chatbot providing dynamic, context-aware responses with features like embedding questions, agentic flows (e.g., guiding users through processes), rewriting chat messages for clarity, and viewing chat history for reference. Supports FAQs, transaction guidance, and more.
FE-02	Live Chat: Enables real-time human support when the AI cannot resolve a query, ensuring seamless user assistance.
	Business Owner
FE-03	Analytics Dashboard: Provides business owners with detailed analytics on user interactions, chatbot performance, engagement metrics, and financial reports to monitor and optimize operations.
FE-04	Bot Management: Allows business owners to create and customize chatbots, configure settings like persona, channel, and automation, and test bot performance. Includes viewing bot history and ensuring compliance with security and privacy standards.

FE-05	Benchmark Bot: Enables business owners to benchmark bot performance, compare metrics, and view detailed benchmark history to ensure the bot meets business goals.
FE-06	Live Chat: Enable real-time human support when needed.
FE-07	Data Source Management: Data managers can add, update, or remove external knowledge sources, including uploading documents in multiple formats or pulling data directly from websites for seamless updates.
Admin	
FE-08	Model Management: Admins can control active AI models, manage embeddings, track token usage (in/out), monitor costs, and select model types for optimal performance.
FE-09	Plan Management: Admins can oversee subscription plans, ensuring proper allocation and upgrades for users and organizations.R
FE-10	Invoice Management: Admins can manage billing and invoicing processes, ensuring accurate financial operations for the platform.
FE-11	User Management: Admins can handle user accounts, assign roles, and manage permissions to ensure secure and smooth platform operations.

Table 7.1.1. Major features of the project

7.2 Limitations & Exclusions

For the MVP, we focus on essential features to build a strong foundation. The following are out of scope for now but planned for future releases:

Limitations ID	Feature and Description
LI-01	Limited Channel Integrations: Only basic channels are supported; integrations with platforms like Facebook Messenger, Zalo, WhatsApp, Slack, and Microsoft Teams are not yet available.
LI-02	No Fine-Tuning Support: Models are used in their default configurations without support for fine-tuning or custom training based on user data.
LI-03	Limited UI Customization: The system does not yet support deep customization of user interfaces, such as theme changes or layout personalization.
LI-04	No Human Agent Handoff: Conversations are solely between users and AI; human agents cannot currently take over ongoing chats.
LI-05	Single Data Source Restriction: Bots can retrieve information from only one data source at a time; multi-datasource retrieval is not yet supported.
LI-06	Limited Service Packages: Service plans are not yet diversified to accommodate varying needs like bot quantity, storage limits, or SLA levels.

LI-07	Limited Admin Role Segmentation: Role and permission management are not yet segmented by bot or organization; admin privileges are currently global.
LI-08	No Native Mobile Apps (iOS/Android): The platform is accessible only via web browsers; native mobile applications are not available.
LI-09	No Voice or Speech-to-Text Features: Voice-related functionalities, including speech recognition and AI voice interaction, are not yet integrated.
LI-10	No Editable Model Configurations: Users cannot modify model settings such as temperature, token limits, or sampling parameters through the interface.

Table 7.2.1. Limitations & Exclusions

II. Project Management Plan

1. Overview

1.1 Cost & Time Estimations

Total estimate effort				**237**	**29/12 - 28/4**
I	**Sprint 1: Project Initiation & Ideation (week 1-2)**			**16**	**29/12 - 13/1**
1	Project Ideation	Discuss and select a feasible and meaningful project idea	Medium	3	11/1 - 13/1
2	Report 1: Project Introduction	Write Project Introduction report (RP1) including objectives, scope, and team members	Simple	3	29/12 - 31/12
3	Initial Requirements Gathering	Gather initial requirements from users or supervisors	Medium	2	1/1 - 2/1
4	Report 3: SRS v0.9 (Overall Requirement)	Create a preliminary version of the Software Requirements Specification (RP3)	Complex	6	3/1 - 8/1
5	Research (Technology Stack, Tools, etc.)	Research and propose suitable technology stack and project tools	Medium	2	9/1 - 10/1
II	**Sprint 2: System Design & Initial Prototyping (weeks 3-4)**			**28**	**13/1 - 03/02**
1	Report 2: Project Management Plan v1.0	Develop Project Management Plan (RP2) including timeline, resources, and assignments	Medium	4	*13/1 - 16/1*
2	Conduct User Interviews	Conduct interviews with users to understand their	Medium	3	17/1 - 19/1

		needs			
3	System Architecture Design	Design the overall architecture of the system (frontend, backend, APIs)	Complex	3	20/1 - 22/1
4	System Design & DevOps Setup	Set up development environment and CI/CD pipelines	Complex	2	23/1 - 24/1
5	Vertical Prototypes (Code Demo)	Build vertical prototypes demonstrating key functionalities	Complex	8	25/1 - 1/2
6	Report 4: SDS v1.0 (High Level Design Specifications)	Write the Software Design Specification (RP4) with high-level designs	Medium	6	25/1 - 30/1
7	Set up Development Environment	Install and configure tools for development	Simple	2	2/2 - 3/2
III	**Sprint 3: Detailed Design & Proof of Concept (weeks 5-6)**			**37**	**5/2 - 18/2**
1	Report 5: Test Planning (ref Report5_Test Documentation)	Create a comprehensive Test Plan (RP5) for the project	Medium	5	5/2 - 9/2
2	Report 3: SRS v1.0 (Detailed Requirements for Iteration 1)	Write the detailed requirements for Iteration 1	Complex	6	5/2 - 10/2
3	Report 4: SDS v1.0 (Detailed Design for Iteration 1)	Update the design specification for Iteration 1	Complex	6	10/2 - 16/2
4	Design Database Schema	Design the database schema and data structure	Complex	4	11/2 - 14/2
5	UI/UX Design	Design the user interface and user experience	Medium	6	11/2 - 16/2
6	Vertical Prototypes (POC)	Build technical proof-of-concept prototypes	Complex	10	11/2 - 18/2
IV	**Sprint 4: Implementation Iteration 1 (weeks 7-8)**			**44**	**19/2 - 3/3**
1	Code Package & Unit Testing Report	Develop and unit test the software modules	Complex	10	19/2 - 28/2
2	Report 5.1: Unit Test Cases	Write test cases for unit testing	Medium	4	19/2 - 22/2
3	Report 5.1: Unit Test Report	Report the results of unit testing	Medium	4	26/2 - 29/2
4	Report 5.2: Integration Test Cases	Write test cases for integration testing	Medium	4	22/2 - 25/3
5	Report 5.2: Integration	Report the results of	Medium	4	29/2 - 3/3

	Test Report	integration testing			
6	Report 3: SRS v1.0 (Detailed Requirements for Iteration 2)	Update detailed requirements for Iteration 2	Medium	4	24/2 - 27/2
7	Report 4: SDS v1.0 (Detailed Design for Iteration 2)	Update the design specification for Iteration 2	Complex	6	27/2 - 2/3
8	Software Package Version 1	Build and deliver the first version of the software	Complex	8	23/2 - 3/3
V	**Sprint 5: Implementation Iteration 2 (weeks 9-10)**			**44**	**4/3 - 17/3**
1	Code Package & Unit Testing Report	Develop and unit test the software modules	Complex	10	4/3 - 13/3
2	Report 5.1: Unit Test Cases	Write test cases for unit testing	Medium	4	4/3 - 7/3
3	Report 5.1: Unit Test Report	Report the results of unit testing	Medium	4	10/3 - 13/3
4	Report 5.2: Integration Test Cases	Write test cases for integration testing	Medium	4	8/3 - 11/3
5	Report 5.2: Integration Test Report	Report the results of integration testing	Medium	4	14/3 - 17/3
6	Report 3: SRS v1.0 (Detailed Requirements for Iteration 3)	Update detailed requirements for Iteration 3	Medium	4	9/3 -13/3
7	Report 4: SDS v1.0 (Detailed Design for Iteration 3)	Update the design specification for Iteration 3	Complex	6	14/3 - 17/3
8	Software Package Version 2	Build and deliver the second version of the software	Complex	8	4/3 - 17/3
VI	**Sprint 6: Implementation Iteration 3 (weeks 11-12)**			**34**	**18/3 - 31/3**
1	Code Package & Unit Testing Report	Develop and unit test the software modules	Complex	10	18/3 - 26/3
2	Report 5.1: Unit Test Cases	Write test cases for unit testing	Medium	4	18/3 - 21/3
3	Report 5.1: Unit Test Report	Report the results of unit testing	Medium	4	24/3 - 27/3
4	Report 5.2: Integration Test Cases	Write test cases for integration testing	Medium	4	22/3 - 25/3
5	Report 5.2: Integration Test Report	Report the results of integration testing	Medium	4	28/3 - 31/3
6	Software Package	Build and deliver the third	Complex	8	22/3 - 31/3

	Version 2	version of the software			
VII	**Sprint 7: Verification & Finalization (week 13-14)**			**19**	**1/4 - 14/4**
1	System Test Report	Report results from full system testing.	Medium	5	1/4 - 5/4
2	User Guides	Write user manuals and guides	Medium	7	6/4 - 12/4
3	Final Software Package	Deliver the final version of the software	Medium	7	8/4 - 14/4
VIII	**Sprint 8: Closing (week 15)**			**15**	**15/4 - 28/4**
1	Final Project Report	Write the final comprehensive project report	Medium	6	15/4 - 20/4
2	Presentation File	Prepare the final presentation slides	Simple	5	21/4 - 25/4
3	Project Retrospective	Summarize project lessons learned	Simple	**2**	26/4 - 27/4
4	Documentation Review and Finalization	Review and finalize all project documentation	Medium	**2**	27/4 - 28/4

Table 1.1.1. Work Breakdown Structure

1.2 Project Objectives

#	Metric	Unit	Planned	Actual	Notes / References
1	Effort Usage	Person-day	237 person-days	230 person-days	Reduced effort due to part-time contributions, and review team workload.
2	Review Defects	No of defects	5 defects	7 defects	Slightly above target, additional review sessions scheduled.
3	Unit Test Defects	No of defects	10 defects	8 defects	Fewer issues were found; early bug fixes improved unit quality.
4	Integration Test Defects	No of defects	15 defects	12 defects	Good performance overall; focus on critical integration points.
5	System Test Defects	No of defects	20 defects	18 defects	Minor issues remain; further refinements are planned before release.

Table 1.2.1. Project Objectives

1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans

1	Team member’s health	Very High	Low	Activate emergency response plan: boost the progress of other tasks and provide support for the responsibilities of the recuperating member.
2	Insufficient time spent on the capstone project of team members	High	High	Each member regularly reports any abnormal events to the whole team to quickly facilitate. Make a clear commitment on the day of the project's commencement.
3	Misalignment between project requirements and actual needs	High	Low	Adopt the Agile methodology and narrow a sprint down to 1 week to quickly adapt to changes and to receive stakeholder feedback.
4	Wrong budget estimation	Medium	High	Establish a clear and realistic budget plan for deployment, and server,.. and evaluate it regularly.
5	Scopes are expanded beyond expectation	Medium	Medium	Establish a clear scope and regularly review it to ensure that it remains under control. Any proposed changes should be carefully reviewed before being implemented.
6	Dependency and Compatibility of open-sourced services	Medium	Medium	Use APIs/wrappers to decouple open-source tools from core systems. Test integrations rigorously in staging first. Prioritize projects with corporate backing or large communities for reliability.
7	Operation Complexity of Using Microservice Architecture	Medium	High	Implement centralized monitoring and logging by integrating tools such as Grafana, Kibana,… to gain real-time insights into system performance and troubleshoot issues efficiently.

Table 1.3.1. Project Risks

2. Management Approach

2.1 Quality Management

2.1.1 Principles

Our project quality management follows several guiding principles:

- Well-defined, measurable quality goals that align with strategic directions.
- Team members must commit to quality through actions and dedication. Leader leads by example, set quality targets, and ensure members have what they need to succeed.
- Core workflows are researched to analyze inputs, outputs, interactions, and flow. Bottlenecks and inefficiencies are thus identified for improvement opportunities.

2.1.2 Testing Process

Testing method	Target
Unit Test	Isolate each part of the program and show that the individual parts are correct
Integration Test	Conducted to evaluate the compliance of a system or component with specified functional requirements.
System Test	Conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.
Usability Test	Witness real users perform tasks on a prototype to evaluate the app's ease of use, efficiency, and overall user satisfaction.

Table 2.1.2.1. Testing Process

2.1.3 Defect Classification

Defect Label	Note	Action
URGENT	Defects that prevent essential functionality from working or cause a severe performance issue. The system is unusable until it is addressed.	Fix immediately before any further development. Assign the highest priority.
HIGH	Causes major disruption of functions or features. Workarounds may exist but the functionality is compromised.	Fix at the earliest in the current development cycle. Address before release.
MEDIUM	Annoying but not serious discrepancies that do not impact core functionality. Cosmetic or minor procedural bugs.	Schedule fixes based on impact and effort. It can be postponed to future releases depending on risk assessment.
LOW	Very minor aesthetic or cosmetic discrepancies with minimal or no impact on use such as typos, missing labels, etc.	Defer fix to future releases. No immediate action is required but record for future reference.

Table 2.1.3.1. Defect Classification

2.1.4 Reviewing

- Conduct regular design and code reviews by independent teams to identify issues early. Static code analysis tools can also help.
- Perform inspections and walkthroughs on requirements, design documents, and other work products.

2.2 Project Training Plan

Some conventions must be followed by all members during the development process, as well as new technologies adopted. For those factors, the team assigns one member with the most proficient expertise in that field to host training sessions for the members involved.

Training Area	Participants	When, Duration
Git flow	All members	07/12
RAG flow	All member	15/21 - 27/12
Nest.js framework	All members	22/12
Figma design	DungHM, LongPV, DucNM	07/01
Docker, devops basic	DucTA, DucNM, AnhNMT	11/01

Table 2.2.1. Project Training Plan

3. Project Deliverables

#	Deliverables	Due Date	Notes
1	Project Introduction	31/01/2025	Provide an overview of the project: background, objective, and scope.
2	Project Plan	16/01/2025	Outline project schedule and milestones. A risk management plan and communication plan are also included
3	Software Requirement Specification (SRS)	13/03/2025	Detail the functional & non-functional requirements and use case description
4	Software Design Description (SDD)	17/03/2025	Include architectural design, database design, and app prototype
5	Product Release 1	03/03/2024	Launch the first official usable product to collect feedback
6	Product Release 2	17/03/2025	Launch the 1.1 version of the product, with added functionalities compared to the first version
7	Product Release 3	31/03/2025	Launch the 1.2 version of the product, with added functionalities compared to the second version
9	Software Test Document (STD)	05/04/2025	Detail the result of different types of testing: UT, IT, AT
10	Software User Guides	25/04/2025	Provide clear instructions on how to use the developed software, with concise language and illustrated screenshots and diagrams
11	Final Project Report	27/04/2025	Includes all the previous project documents in combined, source code, and thesis presentation slide

Table 3.1. Project Deliverables

4. Responsibility Assignments

Overall, the team consists of five members, supervised by lecturer KienNT. Our team is made up of two roles: the Team Leader, and the Development team. The development team is then divided into 4 smaller teams that correspond to a specific area, each of which has persons in charge (PIC) and supporting members

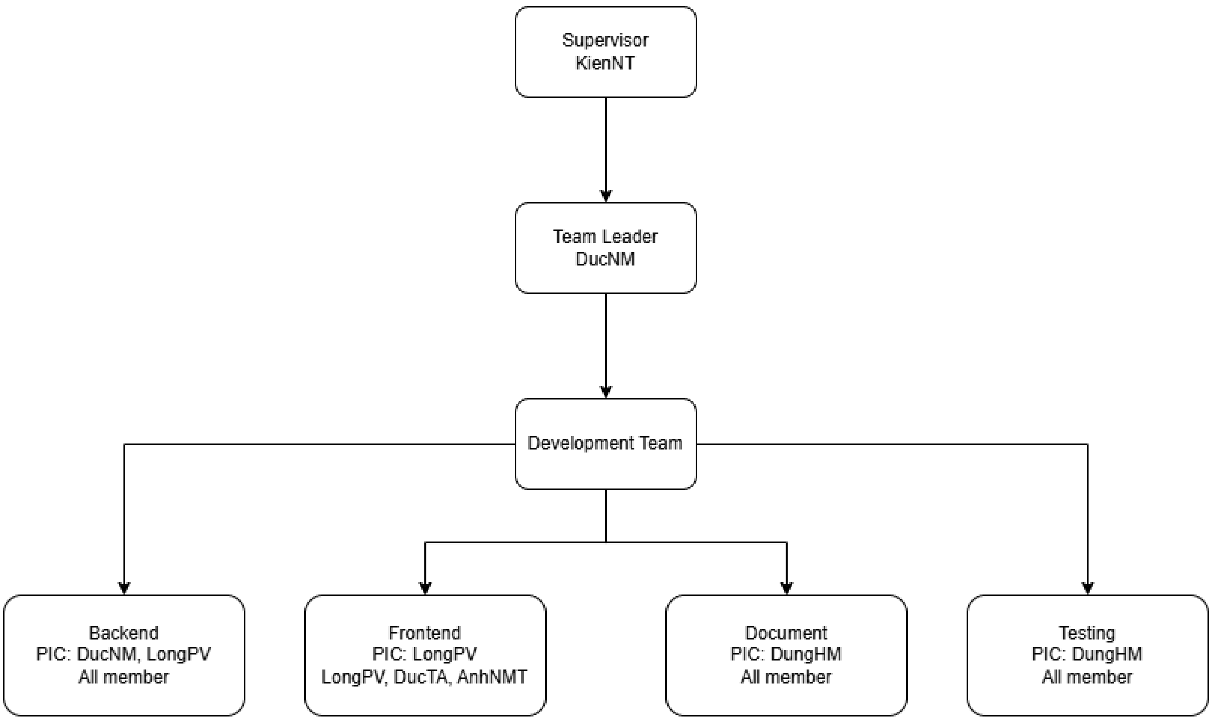


Figure 4.1. Team Structure

RACI Chart: R~Responsible, A~Accountable, C~Consulted, I~Informed

Work Package	**LongPV**	**DucNM**	**DucTA**	**DungHM**	**AnhNMT**
Project Planning & Tracking	C	R, A	C	A	C
Prepare Project Introduction Document	R	A	I	I	C
Prepare SRS Document (Overview Part)	C	A	C	R	C
Prepare SRS Document (User Requirements)	A	R	C	C	R

Prepare SDD	R	R, A	R	R	R
Prepare Test Documents and Report	I	A	I	R	I
Prepare User Guides	R	R, A	R	R	R
Code Backend	R, A	R, A	R	R	R
Code Frontend	R, A	C	R	C	R
Prepare Final Report (combining previous documents and presentation slide)	R	R, A	R	R	R

Table 4.1. Responsibility assignments

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Team Check-in (Daily, 15 min)	Team members	Answer 3 questions: what did I work on, tomorrow's planned work and issues (if any)	21:00 everyday	Google Meet
Weekly meeting with the supervisor	Team members and supervisor	Check current progress	20:30 every Saturday	Google Meet
Work Planning Session	Team members	Align on priorities for the coming period	09:00 every Sunday	Offline
Team Improvement Discussion	Team members	Reflect on what's working and what could be better	At the end of each interation	Google Meet
Documentation & Async Communication	Team members	Record key decisions and information without excessive formality	Ongoing, as needed	Google Meet, In person
Issue Resolution	Team members	Solve unexpected problems quickly	As needed	Google Meet, In-person

Table 5.1. Project Communications

6. Configuration Management

6.1 Document Management

Our team stores and manages project documents using Google Drive. This cloud-based platform facilitates real-time collaboration where teams can edit files simultaneously. Google Drive also retains previous document revisions.

Google Drive contains three main categories. The "Documents" section contains deliverables and additional documents for the project. It unites all the components into one comprehensive collection. The "Management" area includes information on the team and its operations during the product development phase. The last part is "Testing" to contain Testing reports.

In Google Drive, we create project documents using Google Docs and Google Sheets. This allows the entire team to coordinate their work virtually together.

Link to Driver: click here

Tool	Activity
Google Drive	Store all documents
Googles Docs	Document Report, User Guide
Google Sheet	Project Tracking, Project Weekly, Issues Report
Canva	Slide Presentation
Figma	Design Components, Screens
Draw.io / Lucid Chart	Diagrams

Table 6.1.1. Document Management

6.2 Source Code Management

We manage project source codes & their changes/versions in GitHub organization. Also, we strictly follow the Git workflow described in the following diagram:
Link to GitHub: click here

6.3 Tools & Infrastructures

Category	Tools / Infrastructure
Development	Next.js, Nodejs, FastAPI
Data platform	PostgreSQL, MongoDB, Milvus
Tech platform	Sonar (Check source code), Docker (Launching applications), RabbitMQ (Manage message queue), Redis (Caching data), Prometheus & Grafana (Monitoring system), Kibana (Check log system), Elastic search (searching data), LiteLLM (Manage Model)
IDEs/Editors	Pycharm, Visual Studio Code
Diagramming	Draw.io, Lucid Chart

Designing	Adobe Figma
Testing tools	Postman, Junit
Documentation	Google Docs/Sheets/Drive
Version Control	GitHub (Source Code)
Deployment server	AWS (Amazon Web Services), On-premises server
Project management	Clickup (Schedule, Tasks, Defects), GitHub (Issues)
Collaborations	Google Meet

Table 6.3.1. Tools and infrastructures

III. Software Requirement Specification

1. Overall Requirements

1.1 Context Diagram

The Conversational AI Platform is a software system designed to improve communication between businesses and their customers through chatbot management, conversation handling, and document processing. It provides role-specific functionalities for customers, business owners, and admins, enabling seamless interactions, efficient management, and comprehensive reporting. The platform aims to enhance customer engagement while offering tools for managing chat conversations, user roles, and billing processes.

The context diagram outlines the system's boundaries and its interactions with external entities, including customers, business owners, and administrators. It illustrates data flows such as message exchanges, user management, payment processing, and reporting. This version of the platform is designed to evolve, with plans for integrating additional features and connections in future releases.

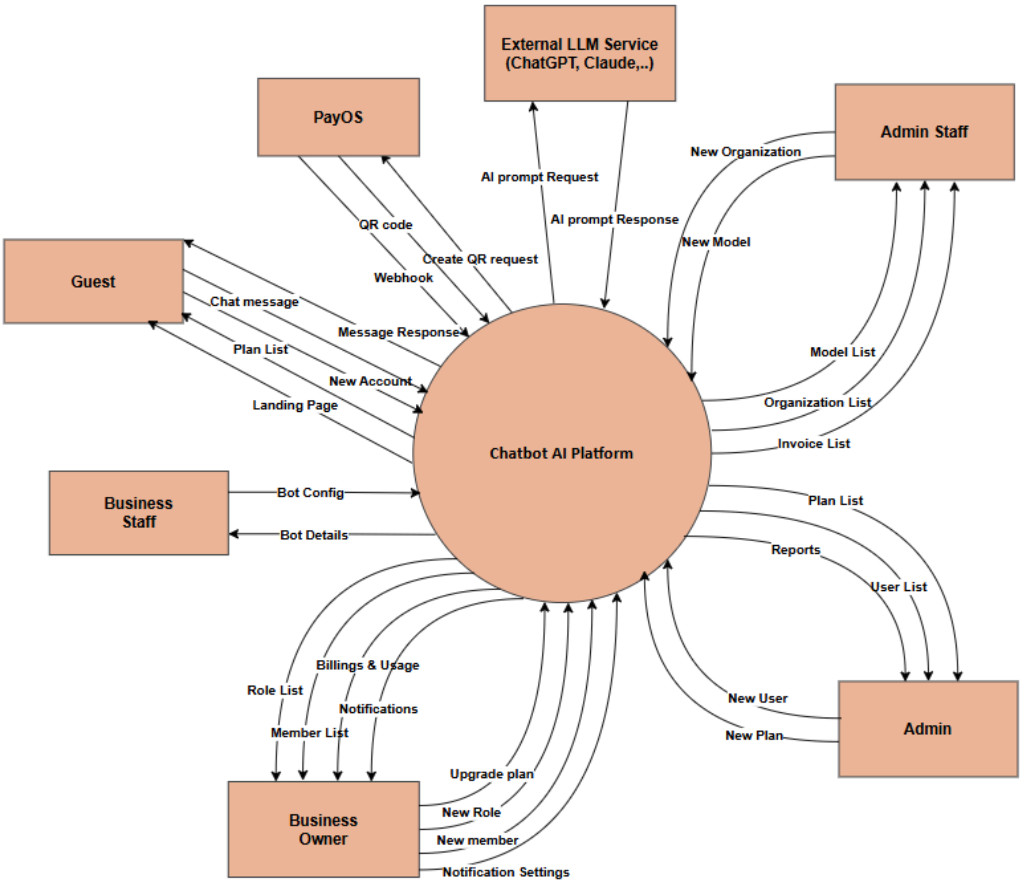


Figure 1.1.1 Context Diagram

1.2 Main Workflows

1.2.1 Setup an organization

1.2.1.1 Setup an organization workflow

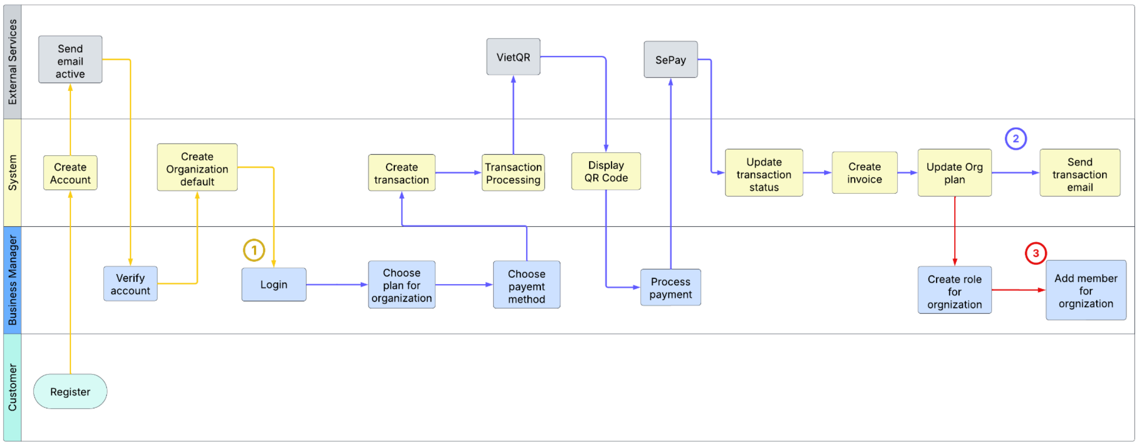


Figure 1.2.1.1.1 Organization Workflow

1.2.1.2 Detail flow

1. **User Registration and Organization Creation:** A user signs up for the system, confirms their email, and gets a default organization.
    - The user registers by providing their details (e.g., email and password).
    - The system sends a confirmation email to the user.
    - The user clicks a link in the email to verify their account.
    - Once verified, the system automatically creates a default organization for the user, linked to their account.
2. **Payment Flow:** The user selects a subscription plan, completes payment through a third-party service, and their organization's plan is updated.
    - The user chooses a plan (e.g., basic, premium) from the system's pricing options.
    - The system creates a transaction and redirects the user to a third-party payment provider to process the payment.
    - After the user completes the payment, the third-party service sends a callback to the system confirming the payment's success.
    - The system generates an invoice and updates the user's organization with the selected plan, granting access to the plan's features.
3. **Add Members to Organization:** After successfully purchasing a plan, business owner can invite members and assign roles.
    - The organization owner (BO) accesses the organization's settings in the system.
    - The BO creates roles (e.g., admin, developer, sale) with specific permissions for the organization.
    - The BO invites members to join the organization by sending email invitations.
    - Invited members accept the invitation and are added to the organization with their assigned roles, allowing them to collaborate within the system.

1.2.2 Bot Setup

1.2.2.1 Bot Setup workflow

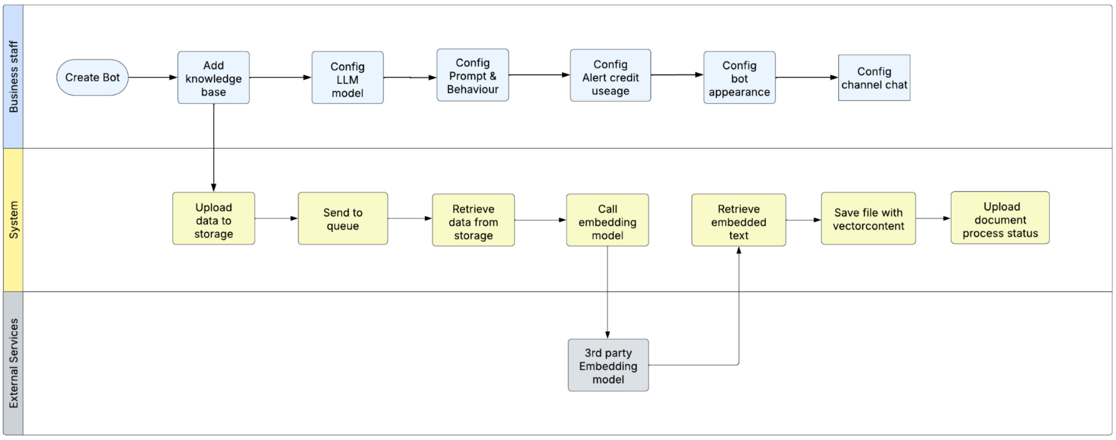


Figure 1.2.2.1.1 Bot setup workflow

1.2.2.2 Detail flow

1. Create Bot: Business staff initiate the bot creation process.
2. Add Knowledge Base:
    a. Upload data to secure storage with format validation
    b. Add files to processing queue to manage system load
    c. Retrieve documents from storage as processing begins
    d. Call embedding model to extract and segment text
    e. Utilize third-party embedding services when needed
    f. Process text into vector embeddings for semantic understanding
    g. Save vectorized content in specialized databases
    h. Update document processing status with real-time feedback
3. The system then processes this information in the background.
4. Configure LLM Model: Business staff select the appropriate language model based on their specific needs (speed, cost, or accuracy).
5. Configure Prompt & Behavior: Business staff define the bot's behavior, tone, and response patterns by setting up specific prompts.
6. Configure Alert Credit Usage: Business staff set up notifications for credit consumption to manage costs.
7. Configure Bot Appearance: Business staff customize the visual aspects and interface of the chatbot.
8. Configure Channel Chat: Business staff set up integration with various communication platforms (website, Zalo, Telegram, Slack).

1.2.3 Chat

1.2.3.1 Chat workflow

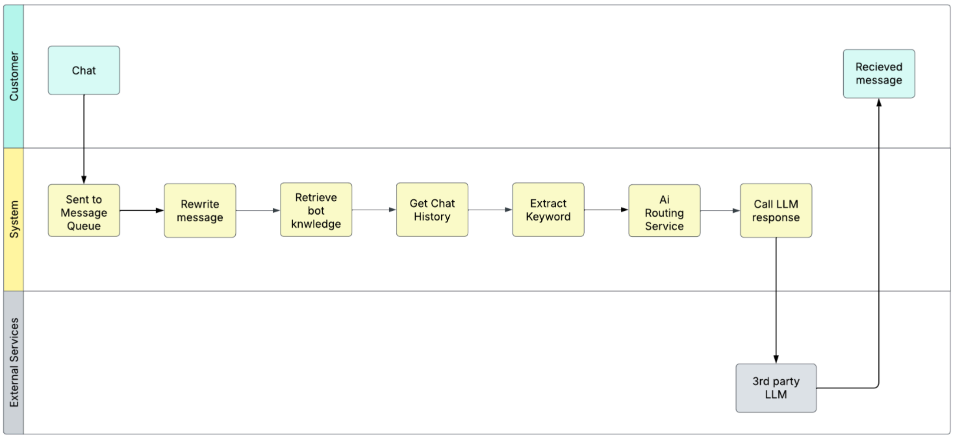


Figure 1.2.3.1.1 Chat workflow

1.2.3.2 Detail flow

1. User Initiates Chat: Customer starts a conversation with the bot.
2. Message Processing:
    a. Messages are sent to a queue for management
    b. The system rewrites/processes the message
    c. Retrieves relevant bot knowledge
    d. Gets chat history for context
    e. Extracts keywords from the message
    f. Routes through AI service
    g. Calls LLM response generation
    h. May utilize third-party LLM services as needed
3. Response Delivery: The system delivers the processed response back to the customer.

1.3 User Requirements

1.3.1 Actors

#	Actor	Description
1	Guest	A user who is not logged into the system, with limited access to public information.
2	Business Staff	Employees working for a business, responsible for managing bots.
3	Business Owner	The owner of the business, with access to and control over all business operations
4	Admin Staff	Administrative staff responsible for supporting and managing system operations, with more limited permissions compared to Admin
5	Admin	The system administrator with the highest level of access, managing the entire system and assigning roles to other users.

Table 1.3.1.1 Actors

1.3.2 Use Cases

ID	Use Case	Feature	Use Case Description
1	View Landing Page	Common	Displays the homepage with an overview of the system's services.
2	Register An Account	Common	Allows a visitor to create an account.
3	Login To The System	Common	Authenticates a user and provides access to the system.
4	Change Password	Common	Allows users to update their password.
5	Reset Password	Common	Resets the user's password if forgotten.
6	Logout	Common	Logs out the user from the system.
7	View/Edit Profile	Common	Allows users to view and edit their profile details.
8	Chat	Common	Enables real-time chat functionality.
9	View Bot List	Business Staff	Displays a list of available bots.
10	Add New Bot	Business Staff	Adds a new chatbot to the system.
11	Update Bot Details	Business Staff	Modify detailed information about a specific bot.
12	Activate/Deactivate Bot	Business Staff	Changes the active status of a bot.
13	View Data Source List	Business Staff	Displays available data sources.
14	Add New Data Source	Business Staff	Adds a new data source for bots.
15	Delete Data Source	Business Staff	Deletes a specific data source.
16	Config Bot's Persona	Business Staff	Configures the bot's persona.
17	Test Bot	Business Staff	Tracks each step in a bot's conversation flow, measuring response time and returned data for testing and validation.
18	View Test History	Business Staff	Access and review the results of previously conducted tests
19	Config Bot Channel	Business Staff	Configures integrations like Telegram, Slack, etc., enabling the bot to connect with external platforms.

20	Setup Automation	Business Staff	Allows users to turn the bot’s automation mode on or off, and configure tools the bot can use such as accessing external data (e.g. Excel) or managing transaction logs.
21	Detect customer insight	Business Staff	Allows users to view user contacts based on insights.
22	View Live Chat	Business Staff	Allows users to monitor and review customer chat interactions in real-time or historically.
23	Benchmark Bot	Business Staff	Evaluates the bot's performance based on various criteria.
24	View Benchmark History	Business Staff	Displays a list of previously executed bot benchmark runs, including model details, configuration settings, and timestamps.
25	View Benchmark Details	Business Staff	Shows detailed information about a specific benchmark run, including input data, model responses, scoring metrics.
26	Compare Benchmark Results	Business Staff	Allows users to select and compare multiple benchmark runs side-by-side, highlighting differences in model performance, accuracy, and configuration settings to support data-driven evaluations and improvements.
27	View Dashboard	Business Owner	Provides an overview of key benchmarking statistics, recent activity, and performance trends through interactive charts and summaries.
28	View Organization List	Business Owner	Displays a list of registered organizations.
29	Add New Organization	Business Owner	Allows the admin to register a new organization.
30	Update Organization Details	Business Owner	Modify detailed information about a specific organization.
31	Activate/Deactivate Organization	Business Owner	Changes the active status of an organization.
32	View Member List	Business Owner	Displays a list of registered members.
33	Add New member	Business Owner	Allows the admin to register a new member.
34	Delete Member	Business Owner	Changes the active status of a member.
35	View Role List	Business Owner	Displays a list of roles.

36	Add New Role	Business Owner	Allows the business owner to define a new role.
37	Update Role Details	Business Owner	Modify details of a specific role
38	Activate/Deactivate Role	Business Owner	Changes the active status of a role.
39	View Plan & Usage	Business Owner	Displays billing information and usage statistics.
40	Upgrade Plan	Business Owner	Enables users to upgrade their subscription plan and payment
41	Setup Credit Alerts	Business Owner	Configure an email address to receive automatic alerts when their credit balance reaches a specified threshold
42	View Model List	Admin Staff	Displays a list of available models.
43	View Model Details	Admin Staff	Displays detailed information about a specific model.
44	View Invoice List	Admin Staff	Displays a list of invoices.
45	Add New Invoice	Admin Staff	Allows admins to add a new invoice.
46	Update Invoice Details	Admin Staff	Modify detailed information about a specific invoice.
47	Change Invoice Status	Admin Staff	Changes the active status of a model.
48	View User List	Admin	Displays a list of system users.
49	Add New User	Admin	Allows the admin to create a new user account.
50	Update User Details	Admin	Modify detailed information about a specific user.
51	Activate/Deactivate User	Admin	Changes the active status of a user account.
52	View Plan List	Admin	Displays a list of available subscription plans.
53	Add New Plan	Admin	Allows admins to define a new subscription plan.
54	Update Plan Details	Admin	Modify details of a specific subscription plan.
55	Activate/Deactivate Plan	Admin	Changes the active status of a subscription plan.

Table 1.3.2.1 Use Cases

1.3.3 Use Case Diagram

1.3.3.1 UCs for Guest

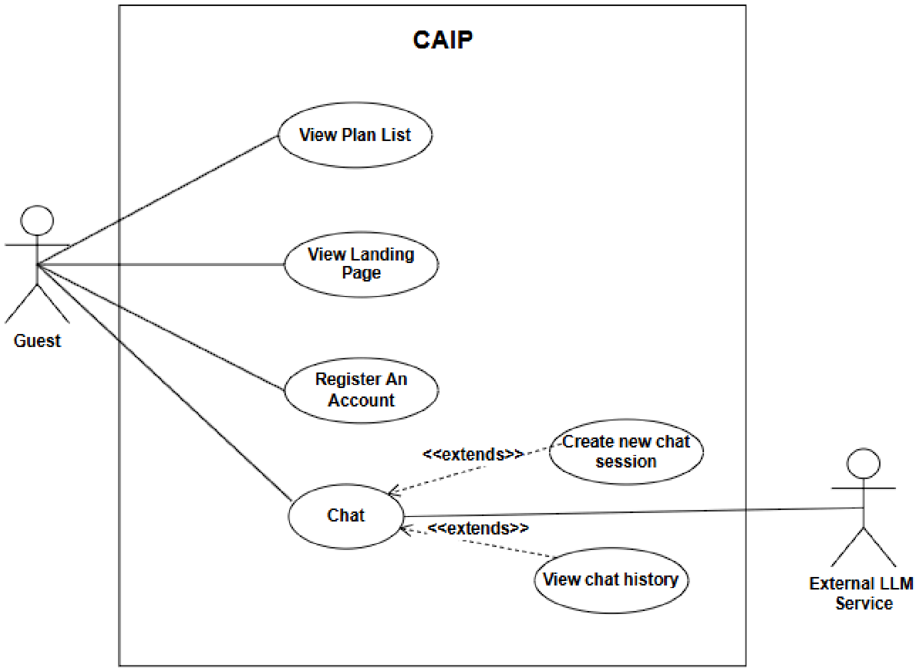


Figure 1.3.3.1.1 UCs for Guest

1.3.3.2 UCs for Registered user

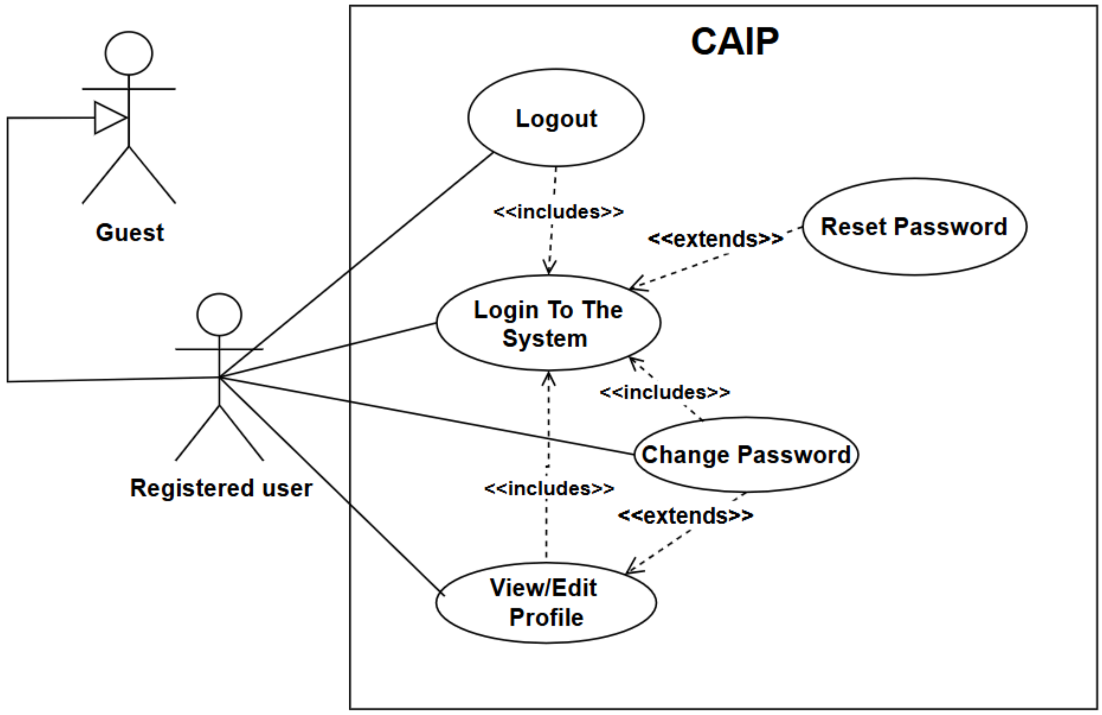


Figure 1.3.3.2.1 UCs for Registered user

1.3.3.3 UCs for Business Staff

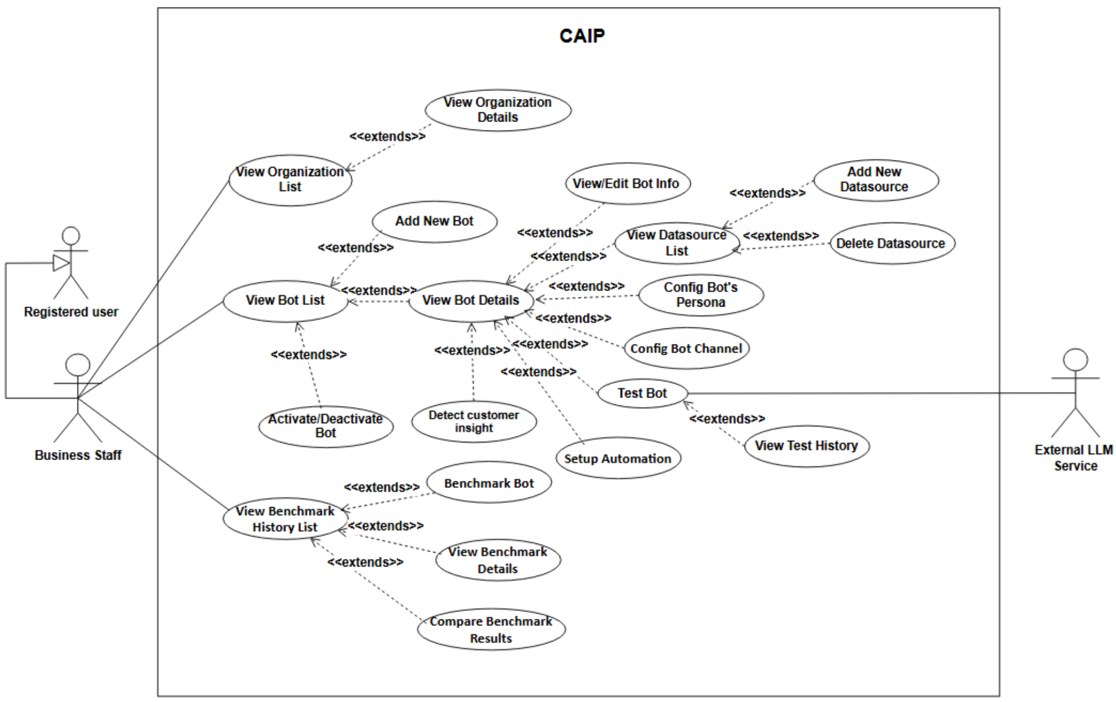


Figure 1.3.3.3.1 UCs for Business Staff

1.3.3.4 UCs for Business Owner

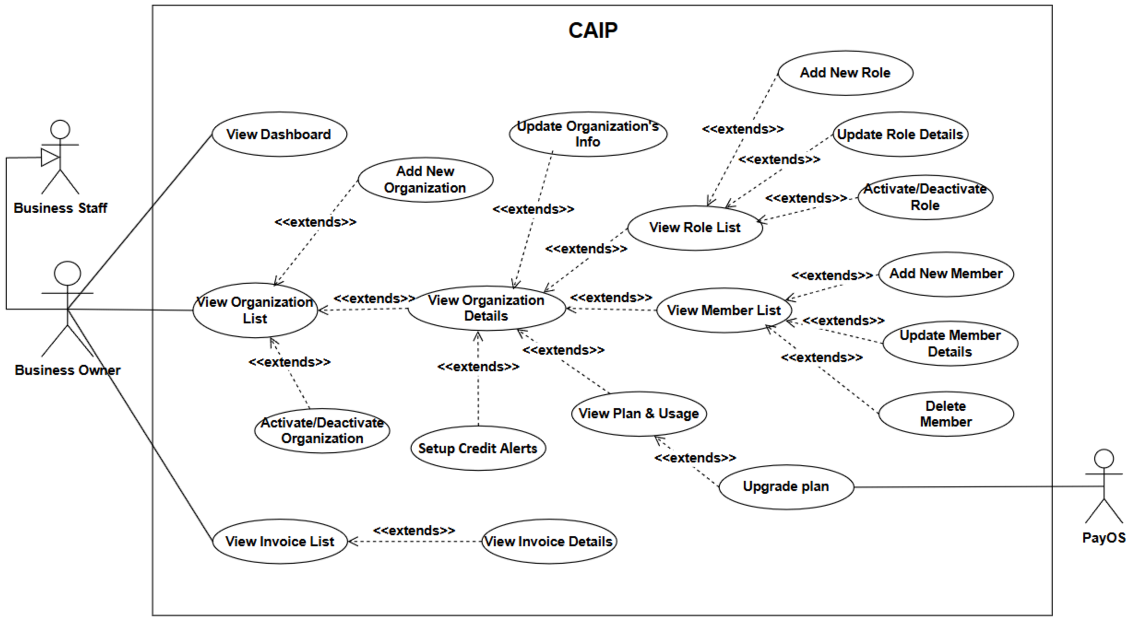


Figure 1.3.3.4.1 UCs for Business Owner

1.3.3.5 UCs for Admin Staff

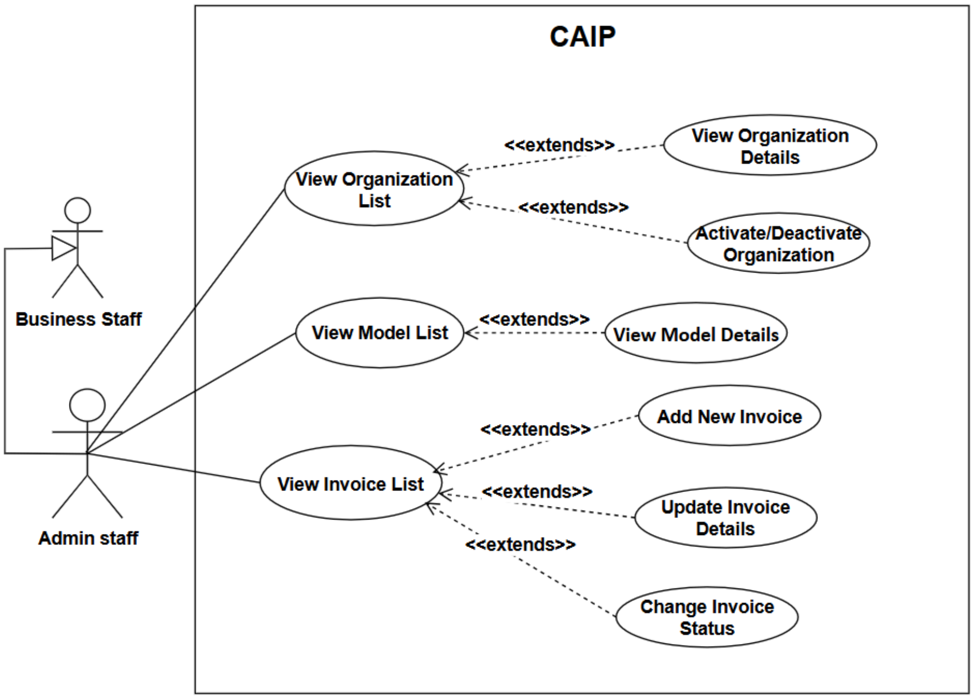


Figure 1.3.3.5.1 UCs for Admin Staff

1.3.3.6 UCs for Admin

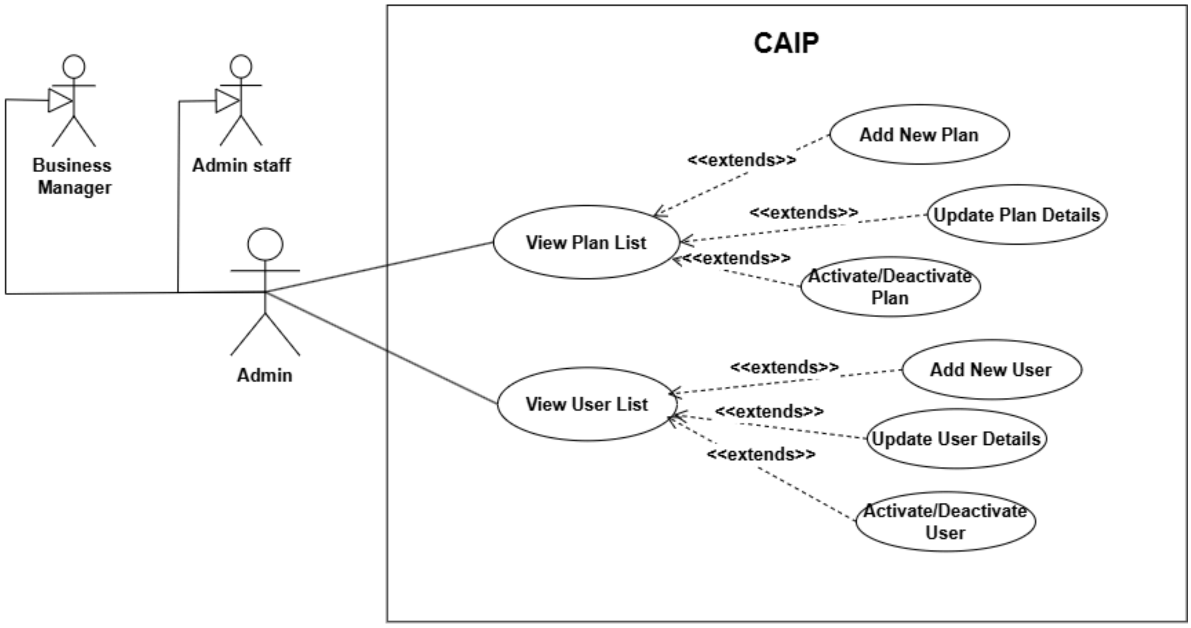


Figure 1.3.3.6.1 UCs for Admin

1.4 System Functionalities

1.4.1 Screens Flow

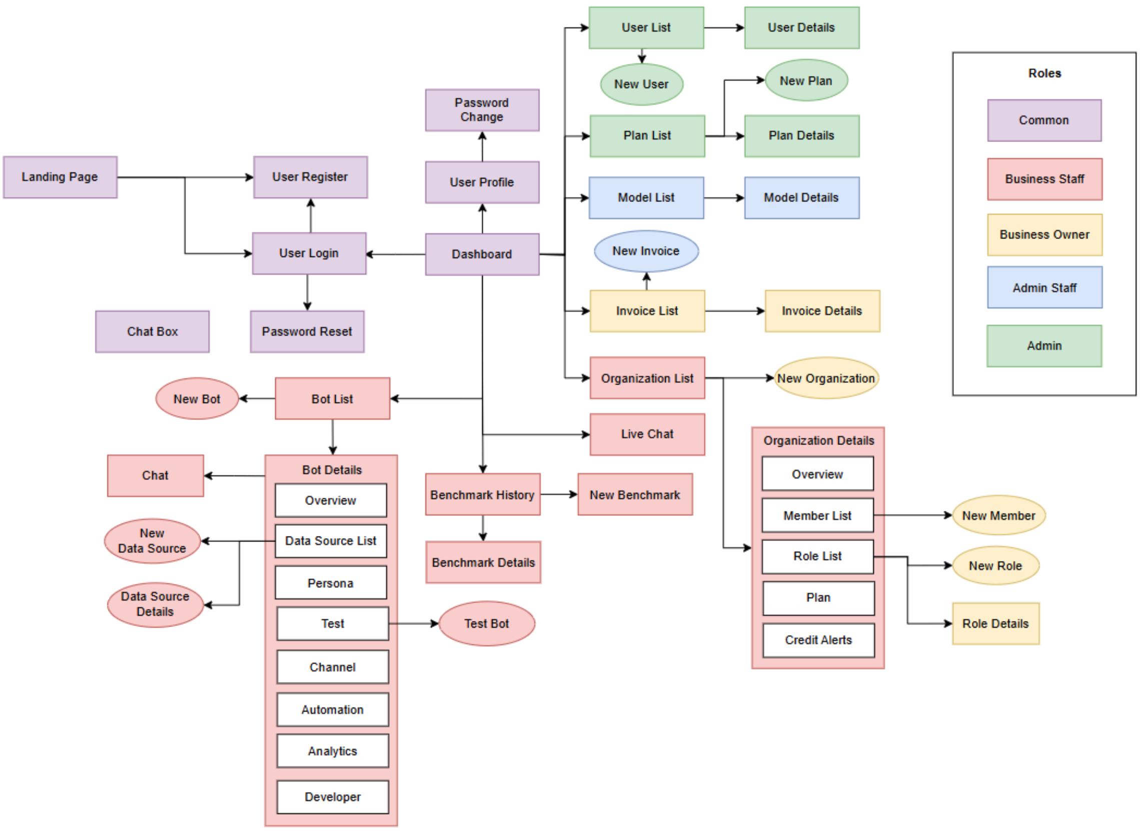


Figure 1.4.1.1 Screens Flow

1.4.2 Screen Authorization

Screen	Guest	Business Staff	Business Owner	Admin Staff	Admin
Landing Page	X	X	X	X	X
Chat Box	X	X	X	X	X
User Register	X	X	X	X	X
User Login		X	X	X	X
Password Reset		X	X	X	X
Password Change		X	X	X	X
User Profile		X	X	X	X
Dashboard		X	X	X	X
Bot List		X	X	X	X
Bot Details		X	X	X	X
New Bot		X	X	X	X
Live Chat		X	X	X	X
Chat		X	X	X	X
Test Bot		X	X	X	X
New Data Source		X	X	X	X
Data Source Details		X	X	X	X
Organization List		X	X	X	X
Organization Details		X	X	X	X
Benchmark History		X	X	X	X
New Benchmark		X	X	X	X
Benchmark Details		X	X	X	X
New Organization			X	X	X
New Member			X	X	X
New Role			X	X	X
Role Details			X	X	X
Invoice List			X	X	X
Invoice Details			X	X	X
Model List				X	X
Model Details				X	X
New Invoice				X	X
User List					X
New User					X
User Details					X
Plan List					X
New Plan					X
Plan Details					X

Table 1.4.2.1 Screen Authorization

1.4.3 Non-UI Functions

#	Feature	System Function	Description

1	Datasource Management	Process uploaded data	Automatically ingests and processes knowledge data received from external sources or system integrations in the background, without requiring user interaction.
2	Datasource Management	Persist Processed Data	Saves processed data and original files into designated storage systems, such as cloud-based S3 buckets or on-premise storage, ensuring data integrity and accessibility for future retrieval.
3	Chat	Insight and process chat responses	Utilizes multiple LLM models to analyze chat responses for personal information (e.g., email, phone numbers). Backend systems automatically evaluate the responses and notify the business with relevant insights or alerts.

Table 1.4.3.1 Non-UI Functions

1.5 Entity Relationship Diagram

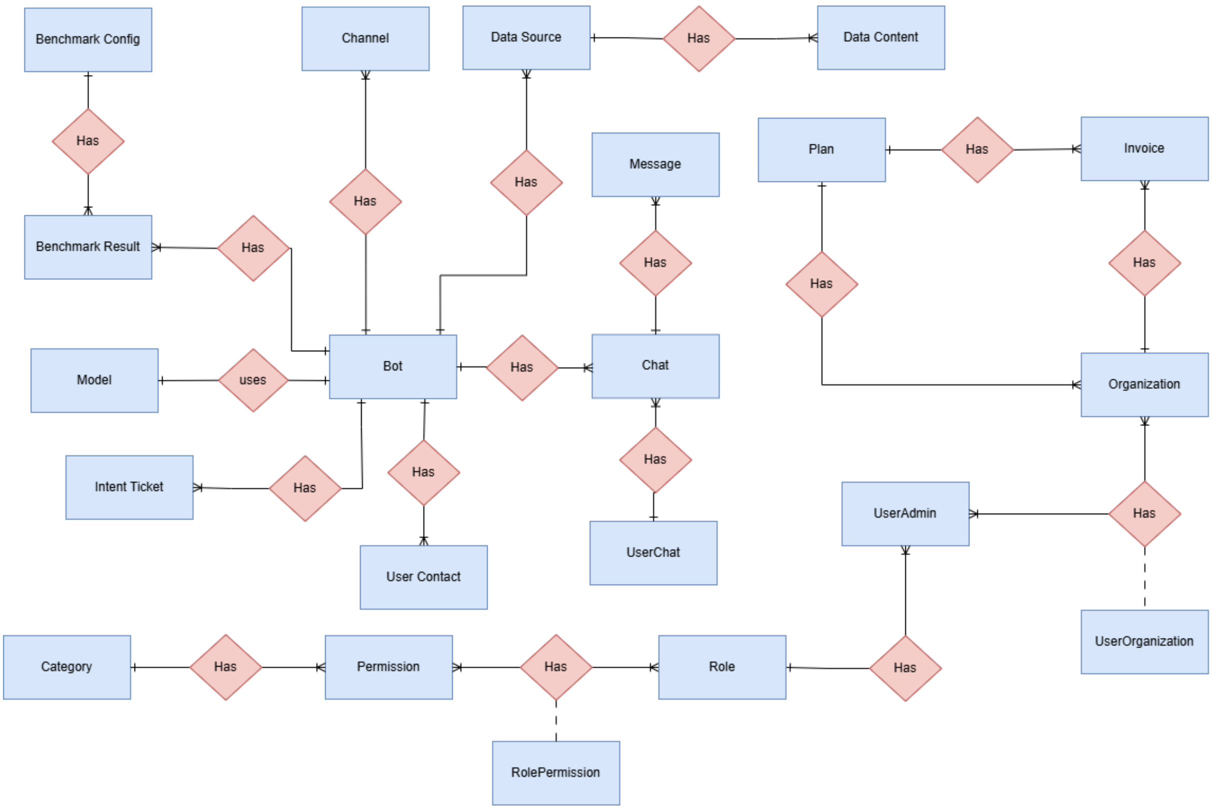


Table 1.5.1 Entity Relationship Diagram

2. Use Case Specifications

2.1 Common Feature

2.1.1 View landing page

UC ID and Name:	**UC-1_View landing page**

Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Admin Staff, Business Owner, Business Staff, Customer, Guest	Secondary Actors:	None
Trigger:	This screen triggers when a user enters the website		
Description:	As a user, I want to view the landing page to get an overview of the system's features and offerings.		
Preconditions:	PRE-1. Users enter the website		
Postconditions:	POST-1. The website display sliders, information, plans		
Normal Flow:	**1.0 View landing page** 1. Displaying the company logo and/or name 2. Providing a navigation menu to help users explore different sections of the site 3. Highlighting featured content or promotions 4. Providing access to user account information, if applicable.		
Alternative Flows:	1. If the user is not logged in, the home page might provide a call-to-action button to encourage them to create an account.		
Exceptions:	N/A		

Table 2.1.1.1 View landing page

2.1.2 Register an account

UC ID and Name:	**UC-2_Register an account**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Admin Staff, Business Owner, Business Staff, Customer, Guest	Secondary Actors:	None
Trigger:	Click a button Register		
Description:	As a user, I want to register an account so that I can access platform features and services.		
Preconditions:	PRE-1. The user email/phone does not register		
Postconditions:	POST-1. The user has an account to login into system		
Normal Flow:	**2.0 Register an account** 1. User selects the account registration function on the main screen. 2. The system displays a text box allowing the user to create an fullname, username, password. 3. User fills in the account name and password he wants to create. **4. The system checks the characters entered in the username and password fields.** 5. User selects “Register”. **6. The system checks the account name and password with the data on the system. If it exists, go to the registration page..** 7. The system checks the email domain with the data on the system. If it does not exist, add the account and password to the system.		

	8.1 The user receives the OTP from the inputted phone. 8.2 The user receives a link to verify from inputted email. 9.1 The web moves to enter the OTP screen. 9.2 User clicks the links then redirects to his/her dashboard. 10. The user enters the OTP. 11. The user clicks the “Confirm” button. 12 The system moves to the his/her dashboard.
Alternative Flows:	**2.1 Register an account** **4. Username containing special characters** 1. The system displays the message "Invalid username or password" below the text box of the registered name. 2. User re-enters the username. 3. Go back to step 4. **2.2 Register an account** **6. The registered name matches the data on the system** 1. The system displays the message "Registered name is already in use". 2. User re-enters the registered name and password. 3. Go back to step 6. 7. The system checks the email domain with the data on the system. If it does not exist, add the account and password to the system. 1. The system displays the message "Email domain is not allowed". 2. User re-enters the registered name and password. 3. Go back to step 6.
Exceptions:	N/A
Business Rules:	BR1, BR2

Table 2.1.2.1 Register an account

2.1.3 Login to the system

UC ID and Name:	**UC-3_Login to the system**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Admin Staff, Business Owner, Business Staff, Customer, Guest	Secondary Actors:	None
Trigger:	User clicks Login button		
Description:	As a user, I want to be able to log into the system so that I can use the system’s authenticated features and access my personalized account.		
Preconditions:	PRE-1. User account has been created & authorized		
Postconditions:	POST-1. User logs in the system successfully		
Normal Flow:	**3.0 Login to the system** 1. User accesses the User Login screen 2. User types in the login details or choose other login options (see 3.1) 3. User clicks the Login button 4. System validates the login details (see 3.0.E1)		

	5. System allows user to access 6. System tracks user's success login to the Activity Log 7. System accesses the DashBoard
Alternative Flows:	***3.1 Google Login*** 1. User chooses to login system using Google account 2. System redirects the user to the Google's Login screen 3. User types in the Google account details and chooses to login 4. Google validates user's login information successfully and redirect him/her back to the system 5. Return to step 5 of normal flow.
Exceptions:	***3.0.E1 System can't authenticate the user*** 1. The Error Message screen is shown to the user 2. User cancels the logging in *=> UC stops, change to UC-1_View Landing Page* 3. User clicks "Forgot Password?" link *=> change to UC-5_Reset Password* 4. User clicks "Register" link *=> change to UC-2_Register* ***3.0.E2 User has been blocked*** 1. The system show show the message " Your account has been blocked"
Business Rules:	BR1, BR2

Table 2.1.3.1 Login to the system

2.1.4 Change Password

UC ID and Name:	**UC-4_Change password**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Subject manager, Class Manager, Team Leader, Student	Primary Actor:	None
Trigger:	When users click on the button "Change Password"		
Description:	As a user, I want to change my account password to maintain account security or update it to a new one.		
Preconditions:	PRE-1. Users login successfully		
Postconditions:	POST-1. Users can change passwords.		
Normal Flow:	**4.0 Change password** 1. User selects "Change Password" **2. The system displays a text box allowing the user to enter the old password (1 time) and enter the old password (2 times).** 4. Click "Save Changes" and the user has changed the new password for the system.		
Alternative Flows:	**4.1 User entered wrong old password** 1. The system displays the message "Need to enter the correct old password". 2. After entering, continue the sequence.		

	4.2 User entered wrong new password 1. The system displays the message “New password does not match”. 2. The user must enter the new password twice, making sure that the two passwords match. 3. After entering, continue the sequence.
Exceptions:	N/A
Business Rules:	BR1, BR2

Table 2.1.4.1 Change Password

2.1.5 Reset Password

UC ID and Name:	**UC-5_Reset password**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Admin Staff, Business Owner, Business Staff, Customer, Guest	Secondary Actors:	None
Trigger:	When users click on the “Forgot Password” button in the Login screen		
Description:	As a user, I want to reset my password if I forget it so that I can regain access to my account.		
Preconditions:	PRE-1. The user account is active		
Postconditions:	POST-1. The user can log in with the new password		
Normal Flow:	**5.0 Reset password** 1. The user clicks “Forgot password?” 2. The user enters an email/phone. 3. The user clicks the “Continue” button. 4. The system checks the email that the user entered. **5. The user receives the OTP from inputted email/phone.** 6. The web moves to enter the OTP screen. **7. The user enters the OTP.** 8. The user clicks the “Confirm” button. 9. The system checks the input OTP that the user entered. 10. The web moves to reset the password screen. 11. The user enters the new password and confirms the new password, then clicks the “Save” button. 12. The system checks the valid password that the user entered. 15. When the user clicks OK, the dialog is closed and the web returns to the home page		
Alternative Flows:	**4.1 User did not receive the confirmation code** 1. The system displays the message “Confirmation code sent”. 2. User re-enters the confirmation code. 3. Go back to step 6. **4.2 User entered wrong confirmation code** 1. The system displays the message "Invalid confirmation code". 2. User re-enters the confirmation code. 3. Go back to step 8.		

Exceptions:	N/A
Business Rules:	BR1, BR2

Table 2.1.5.1 Reset Password

2.1.6 Logout

UC ID and Name:	**UC-6_Logout**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Admin Staff, Business Owner, Business Staff, Customer, Guest	Secondary Actors:	None
Trigger:	When users click on the button “Logout”		
Description:	As a user, I want to log out of my account to end my session and protect my account from unauthorized access.		
Preconditions:	PRE-1. Users login successfully		
Postconditions:	POST-1. Users logout of the system.		
Normal Flow:	**6.0 Logout** 1. User selects "Logout” 2. User logout of the system successfully		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.1.6.1 Logout

2.1.7 View/Edit Profile

UC ID and Name:	**UC-7_View/Edit profile**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Amin, Admin Staff, Business Owner, Business Staff, Customer, Guest	Secondary Actors:	None
Trigger:	User clicks “My profile” button		
Description:	*As a user, I want to view my profile so that i can confirm and update information*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. User see profile successfully		
Normal Flow:	**7.0 View/Edit profile** 1. User selects "My profile” 2. The system displays the profile of the user. 3. User edits his/her own profile. 4. Click “Save Changes” and the user has updated his/her profile for the system.		
Alternative Flows:	**7.1 Full name contains special characters** 1. The system displays the message "Invalid fullname" 2. User re-enters the full name.		

	3. Go back to step 4 of normal flow
Exceptions:	N/A
Business Rules:	None

Table 2.1.7.1 View/Edit Profile

2.1.8 Chat

UC ID and Name:	**UC-8_Chat**		
Created By:	DungHM	Date Created:	22/Jan/2025
Primary Actor:	Guest	Secondary Actors:	External LLM Service
Trigger:	User use Chat box in website		
Description:	*As a user, I want to chat with a bot so that I can quickly get answers to my questions, receive assistance, or complete tasks without needing to contact a human agent.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. The user receives automated responses or assistance from the chatbot.		
Normal Flow:	**8.0 Chat** 1. User selects the "Chat with Bot" option from the application. 2. The chatbot welcomes the user and offers help or suggestions. 3. User types a query or request in the chat input field. 4. The system processes the input and generates an appropriate response. 5. The chatbot displays the response in the conversation thread. 6. User continues the conversation or completes the interaction by ending the chat.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	BR20		

Table 2.1.8.1 Chat

2.2 Business Staff Feature

2.2.1 View Bot List

UC ID and Name:	**UC-9_View bot list**		
Created By:	DungHM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks “Bots” from the sidebar		
Description:	*As a business staff, I want to view a list of available bots so that I can monitor, manage, and select bots for specific tasks or interactions.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Show a list of all users in the system and their information.		
Normal Flow	**9.0 List bots**		

	1. User clicks the “User” in the sidebar 2. System show a list of all users
Alternative Flows:	None
Exceptions:	N/A
Business Rules:	BR21

Table 2.2.1.1 View Bot List

2.2.2 Add New Bot

UC ID and Name:	**UC-10_Add New Bot**		
Created By:	DungHM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks *Add new chatbot* button in Bot list screen		
Description:	*As a business staff, I want to add a new bot with custom configuration, So that the system can automate tasks and improve efficiency.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Add a new bot to the system successfully		
Normal Flow	**10.0 Add New Bot** 1. User clicks the “Add newBot” button. 2. The web displays a form for the admin to fill out the bot's details. 3. User fills out the form and clicks the “Add” button. 4. A new bot has been created to the system.		
Alternative Flows:	**10.1. Name of a new bot existing in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.2.1 Add New Bot

2.2.3 Update Bot Details

UC ID and Name:	**UC-11_Update Bot Details**		
Created By:	DungHM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks View details button from Bot list screen		
Description:	*As a business staff, I want to be able to update the details of an existing bot so that I can ensure its configuration remains accurate and optimized for current requirements.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The bot to be updated exists in the system		

Postconditions:	POST-1. Bot details has been updated successfully
Normal Flow	**11.0 Update Bot Details** 1. User clicks “View Details” of the bot list screen. 2. The system displays the bot's details. 3. User edits the bot's details . 4. Click “Update” and the user has updated the bot's details for the system.
Alternative Flows:	**11.1. Update name that existed in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.
Exceptions:	None
Business Rules:	None

Table 2.2.3.1 Update Bot Details

2.2.4 Activate/Deactivate Bot

UC ID and Name:	**UC-12_Activate/Deactivate Bot**		
Created By:	DungHM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks Status toggle in Bot List screen		
Description:	*As a business staff, I want to be able to deactivate/activate a bot so that I can control whether the bot is operational or temporarily disabled.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The bot to be updated exists in the system		
Postconditions:	POST-1. Activate/deactivate a bot successfully		
Normal Flow	**12.0 Activate/Deactivate bot** 1. User clicks status toggle to activate/deactivate a bot. 2. The system displays a confirmation form. 3. Click “Confirm” to activate/deactivate a bot. 4. A bot has been activated/deactivated successfully		
Alternative Flows:	None		
Exceptions:	None		
Business Rules:	None		

Table 2.2.4.1 Activate/Deactivate Bot

2.2.5 View Data Source List

UC ID and Name:	**UC-13_View data source list**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None

Trigger:	Clicks “Knowledge base” from the sidebar
Description:	*As a business staff, I want to view a list of base data sources so that I can manage or verify the information used by the bot for interactions.*
Preconditions:	PRE-1. User has logged the system
Postconditions:	POST-1. Show a list of all data sources for a specific bot in the system.
Normal Flow	**13.0 View data source list** 1. User clicks the “Knowledge” in the tab in “Bot Details” screen 2. System shows a list of all data sources.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules:	N/A

Table 2.2.5.1 View Data Source List

2.2.6 Add New Data Source

UC ID and Name:	**UC-14_Add New Data Source**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks *Add new data source* button in Data source list screen		
Description:	*As a business staff, I want to add a new base data source so that the bot can use it for better responses and interactions.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Add a new data source to the system successfully		
Normal Flow	**14.0 Add New Data Source** 1. User clicks the “Add new data source” button. 2. The web displays a form for the admin to fill out the data source's details. 3. User fills out the form and clicks the “Add” button. 4. A new data source has been added to the system.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.6.1 Add New Data Source

2.2.7 Delete Data Source

UC ID and Name:	**UC-15_Delete Data Source**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None

Trigger:	Clicks *Delete* button in Data source list screen
Description:	*As a business staff, I want to delete an existing base data source so that it is no longer used by the bot.*
Preconditions:	PRE-1. User has logged the system
Postconditions:	POST-1. The selected data source is successfully removed from the system.
Normal Flow	**15.0 Delete Data Source** 1. User navigates to the data source list. 2. User selects a data source to delete. 3. The system prompts for confirmation. 4. The system deletes the selected data source. 5. The system confirms the successful deletion and updates the list of data sources.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules:	N/A

Table 2.2.7.1 Delete Data Source

2.2.8 Config Bot's Persona

UC ID and Name:	**UC-16_Config Bot's Persona**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Persona" tab in "Bot Details" screen		
Description:	*As a business staff, I want to configure the bot settings so that I can customize its behavior.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Bot configurations are successfully updated.		
Normal Flow	**16.0 Config Bot's Persona** 1. User clicks the "Persona" tab in "Bot Details" screen 2. System displays the current bot configuration. 3. User updates bot information 4. User saves the configuration changes. 5. System confirms successful configuration.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.8.1 Config Bot's Persona

2.2.9 Test Bot

UC ID and Name:	**UC-17_Test Bot**

Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	External LLM Service
Trigger:	User navigates to the "Test" tab in "Bot Details" screen		
Description:	*As a business staff, I want to know the details behind the response so that I can ensure it behaves correctly before releasing it to customers.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Detailed information behind the response is shown.		
Normal Flow	**17.0 Test Bot** 1. User clicks the "Test" tab in the "Bot Details" screen. 2. User clicks the "Debug Chat Flow" button. 3. User types a query or request in the chat input field. 4. The system processes the input and generates an appropriate response. 5. Click "Transaction Log" above the response. 6. The chatbot displays detailed info about the response.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.9.1 Test Bot

2.2.10 View Test History

UC ID and Name:	**UC-18_View Test History**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Test" tab in the "Bot Details" screen.		
Description:	*As a business staff, I want to access and review the results of previously conducted tests so that I can analyze performance, track progress*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Show a list of all previously conducted tests for a specific bot in the system.		
Normal Flow	**18.0 View Test History** 1. User clicks the "Test" tab in "Bot Details" screen 2. System shows a list of previously conducted tests.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.10.1 View Test History

2.2.11 Config Bot Channel

UC ID and Name:	**UC-19_Config Bot Channel**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the “Channel” tab in “Bot Details” screen.		
Description:	*As a business staff, I want to configure the bot settings so that I can customize its behavior.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Bot configurations are successfully updated.		
Normal Flow	**19.0 Config Bot Channel** 1. User clicks the “Channel” tab in “Bot Details” screen 2. System displays the current bot integration info. 3. User updates bot integration info 4. User saves the configuration changes. 5. System confirms successful configuration.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.11.1 Config Bot Channel

2.2.12 Setup Automation

UC ID and Name:	**UC-20_Setup Automation**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Appearance" settings page.		
Description:	*As a business staff, I want to configure the bot settings so that I can customize its behavior.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Bot configurations are successfully updated.		
Normal Flow	**20.0 Setup Automation** 1. User clicks the “Automation” in the tab in “Bot Details” screen 2. System displays the current bot configuration. 3. User updates bot information 4. User saves the configuration changes. 5. System confirms successful configuration.		
Alternative Flows:	N/A		
Exceptions:	N/A		

Business Rules:	N/A

Table 2.2.12.1 Setup Automation

2.2.13 Detect Customer Insight

UC ID and Name:	**UC-21_Detect Customer Insight**		
Created By:	DucNM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Appearance" settings page.		
Description:	*As a business staff, I want to view potential customer contacts based on insights so that I can reach out to them and drive sales.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. System show list of potential customer contact based on insights		
Normal Flow	**21.0 Detect Customer Insight** 1. User clicks the “Analytics” in the tab in “Bot Details” screen 2. User enter list of insights 3. System shows a list of potential customer contacts based on insights.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.13.1 Detect Customer Insight

2.2.14 View Live Chat

UC ID and Name:	**UC-22_View Live Chat**		
Created By:	DungHM	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks “Chat Tracking” from the sidebar		
Description:	*As a business staff, I want to view live chat so that I can monitor conversations and assist if needed*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. The user receives automated responses or assistance from the chatbot.		
Normal Flow:	**22.0 View Live Chat** 1. User selects the “ChatTracking” from the sidebar. 2. System show list of conversations 3. Users choose a specific conversation. 4. System shows details of the chosen conversation.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	None		

Table 2.2.14.1 View Live Chat

2.2.15 Benchmark Bot

UC ID and Name:	**UC-23_Benchmark Bot**		
Created By:	LongPV	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	External LLM Service
Trigger:	User clicks the “New Benchmark” button in the “Benchmark History” screen.		
Description:	*As a business staff, I want to benchmark the bot's performance to assess its response time, accuracy, and efficiency under various conditions.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Benchmark results are generated and presented to the user.		
Normal Flow	**23.0 Benchmark Bot** 1. User clicks the “New Benchmark” button in the “Benchmark History” screen. 2. User selects the model to benchmark. 3. System displays benchmarking options. 4. System runs benchmark tests on the bot. 5. System displays benchmark results.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.15.1 Benchmark Bot

2.2.16 View Benchmark History

UC ID and Name:	**UC-24_View Benchmark History**		
Created By:	LongPV	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Evaluation" from the sidebar.		
Description:	*As a Business Staff, I want to view the benchmark history so that I can review past evaluation data and performance metrics.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Benchmark history is displayed to the user.		
Normal Flow	**24.0 View Benchmark History** 1. User clicks on "Evaluation" from the sidebar menu. 2. System loads and displays the Benchmark History page.		
Alternative Flows:	N/A		

Exceptions:	N/A
Business Rules:	N/A

Table 2.2.16.1 View Benchmark History

2.2.17 View Benchmark Details

UC ID and Name:	**UC-25_View Benchmark Details**		
Created By:	LongPV	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Details" from the Benchmark History screen.		
Description:	*As a Business Staff, I want to view detailed information about previously conducted benchmarks so that I can analyze results and make informed decisions for future actions.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Benchmark details are displayed to the user.		
Normal Flow	**25.0 View Benchmark Details** 1. User clicks on the "Details" from the Benchmark History screen. 2. System retrieves and displays a detailed view of selected benchmarks, including data such as criteria, results, dates, and any associated scores.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.2.17.1 View Benchmark Details

2.2.18 Compare Benchmark Results

UC ID and Name:	**UC-26_Compare Benchmark Results**		
Created By:	LongPV	Date Created:	23/Jan/2025
Primary Actor:	Business Staff, Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User selects benchmark results from the list to compare.		
Description:	*As a Business Staff, I want to compare multiple benchmark results so that I can choose an appropriate model and prompt for my bot.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Comparison of selected benchmark results is displayed to the user.		
Normal Flow	**26.0 Compare Benchmark Results** 1. User navigates to the Benchmark History screen and selects multiple benchmark results from the list to compare. 2. System processes the selected benchmarks and generates a comparison view that shows key differences, trends, and metrics.		

	3. Users can analyze the data side-by-side, and make data-driven decisions.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules:	N/A

Table 2.2.18.1 Compare Benchmark Results

2.3 Business Owner Feature

2.3.1 View Dashboard

UC ID and Name:	**UC-27_View Dashboard**		
Created By:	DucNM	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	User navigates to the dashboard page.		
Description:	*As a Business Owner, I want to view the dashboard to monitor bot performance, data insights, and system statistics.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Dashboard information is successfully displayed.		
Normal Flow	**27.0 View Dashboard** 1. User navigates to the dashboard page. 2. System processes and visualizes data into charts, tables, and metrics. 3. System displays real-time and historical insights on the dashboard. 4. User reviews the dashboard information.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.3.1.1 View Dashboard

2.3.2 View Organization List

UC ID and Name:	**UC-28_View Organization List**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	User navigates to the organization list page.		
Description:	*As a Business Owner, I want to view a list of organizations to manage or select them as needed.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Organization list is successfully displayed.		
Normal Flow	**28.0 View Organization List** 1. User navigates to the organization list page. 2. System show a list of all organization		

Alternative Flows:	None
Exceptions:	N/A
Business Rules:	None

Table 2.3.2.1 View Organization List

2.3.3 Add New Organization

UC ID and Name:	**UC-29_Add New Organization**		
Created By:	AnhNMT	Date Created:	18/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks *Add new organization* button in Organization list screen		
Description:	*As a Business Owner, I want to add a new organization to manage users and bots under a structured hierarchy.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Add a new organization to the system successfully		
Normal Flow	**29.0 Add New Organization** 1. User clicks the "Add new organization" button. 2. The web displays a form for the admin to fill out the organization's details. 3. User fills out the form and clicks the "Add" button. 4. A new organization has been created to the system.		
Alternative Flows:	**29.1. Name of a new organization existing in the system** 1. Fill other name that isn't existed in system 2. Return to step 4 of normal flow.		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.3.3.1 Add New Organization

2.3.4 Update Organization Details

UC ID and Name:	**UC-30_Update Organization Details**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks View details button from Organization list screen		
Description:	*As a Business Owner, I want to be able to update the details of an existing organization to keep information accurate and up to date.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The organization to be updated exists in the system		
Postconditions:	POST-1. Organization details has been updated successfully		
Normal Flow	**30.0 Update Organization Details** 1. User clicks "View Details" of the organization list screen. 2. The system displays the organization's details.		

	3. User edits the organization details . 4. Click “Update” and the user has updated the organization's details for the system.
Alternative Flows:	**30.1. Update name that existed in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.
Exceptions:	None
Business Rules:	None

Table 2.3.4.1 Update Organization Details

2.3.5 Activate/Deactivate Organization

UC ID and Name:	**UC-31_Activate/Deactivate Organization**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks Status toggle in Organization List screen		
Description:	*As a Business Owner, I want to be able to activate or deactivate an organization to control its availability and access to system resources.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The organization to be updated exists in the system		
Postconditions:	POST-1. Activate/deactivate a organization successfully		
Normal Flow	**31.0 Activate/Deactivate Organization** 1. User clicks status toggle to activate/deactivate an organization . 2. The system displays a confirmation form. 3. Click “Confirm” to activate/deactivate an organization. 4. An organization has been activated/deactivated successfully.		
Alternative Flows:	None		
Exceptions:	None		
Business Rules:	None		

Table 2.3.5.1 Activate/Deactivate Organization

2.3.6 View Member List

UC ID and Name:	**UC-32_View Member List**		
Created By:	DucNM	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to “Members” tab in the “Bot Details” screen.		
Description:	*As a Business Owner, I want to view the member list to see members of the organization.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Member list is successfully displayed.		
Normal Flow	**32.0 View Member List**		

	1. User clicks the “Member” tab in the “Bot Details” screen. 2. System show a list of all members		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.3.6.1 View Member List

2.3.7 Add New Member

UC ID and Name:	**UC-33_Add New Member**		
Created By:	DucNM	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks *Add member* button in member list screen		
Description:	*As a Business Owner, I want to add a new member to the system to expand the member list.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. A new member is successfully added and notified via email.		
Normal Flow	**33.0 Add New Member** 1. User clicks the “Invite member” button. 2. System displays a form with an email input field. 3. User enters the email address of the new member. 4. User submits the form. 5. System validates the email format and checks for duplicates. 6. System sends a notification email to the new member with account activation details. 7. System confirms the successful addition of the member.		
Alternative Flows:	**33.1. Email of a new member existing in the system** 1. Fill other email that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.3.7.1 Add New Member

2.3.8 Delete Member

UC ID and Name:	**UC-34_Delete Member**		
Created By:	DucNM	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks Delete in Member List screen		
Description:	*As a Business Owner, I want to be able to delete a member so that I can manage the team structure and remove inactive or unauthorized users.*		
Preconditions:	PRE-1. User has logged the system		

Postconditions:	POST-1. Delete a member successfully
Normal Flow	**34.0 Delete Member** 1. User clicks the “Members” tab in the “Bot Details” screen. 2. User clicks Delete on the specific member. 3. The system displays a confirmation form. 4. Click “Confirm” to delete a member. 5. A member has been deleted successfully
Alternative Flows:	None
Exceptions:	None
Business Rules:	None

Table 2.3.8.1 Delete Member

2.3.9 View Role List

UC ID and Name:	**UC-35_View Role List**		
Created By:	LongPV	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	User navigates to the member list page.		
Description:	*As a Business Owner, I want to view the list of roles to manage or assign roles to members.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Role list is successfully displayed.		
Normal Flow	**35.0 View Role List** 1. User clicks the “Roles” tab in the “Bot Details” screen. 2. System show a list of all role		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.3.9.1 View Role List

2.3.10 Add New Role

UC ID and Name:	**UC-36_Add New Role**		
Created By:	LongPV	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks *Add new role* button in Role list screen		
Description:	*As a Business Owner, I want to add a new role to manage user permissions effectively.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. New role is successfully added to the system.		
Normal Flow	**36.0 Add New Role**		

	1. User clicks the “Add new role” button. 2. System displays a role creation form. 3. User enters the role name and sets permissions. 4. User submits the form. 5. System confirms the successful addition of the role.		
Alternative Flows:	**36.1. Name of a new role existing in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.3.10.1 Add New Role

2.3.11 Update Role Details

UC ID and Name:	**UC-37_Update Role Details**		
Created By:	LongPV	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks View details button from Role list screen		
Description:	*As a Business Owner, I want to be able to update role details to modify permissions or correct information.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The role to be updated exists in the system		
Postconditions:	POST-1. Role details has been updated successfully		
Normal Flow	**37.0 Update Role Details** 1. User clicks “View Details” of the role list screen. 2. The system displays the role's details. 3. User edits the role's details . 4. Click “Update” and the user has updated the role's details for the system.		
Alternative Flows:	**37.1. Update name that existed in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	None		
Business Rules:	None		

Table 2.3.11.1 Update Role Details

2.3.12 Activate/Deactivate Role

UC ID and Name:	**UC-38_Activate/Deactivate Role**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	Clicks Status toggle in Role List screen		
Description:	*As a user, I want to be able to activate or deactivate a role*		

Preconditions:	PRE-1. User has logged the system PRE-2. The role to be updated exists in the system
Postconditions:	POST-1. Activate/deactivate a role successfully
Normal Flow	**38.0 Activate/Deactivate Role** 1. User clicks status toggle to activate/deactivate a role. 2. The system displays a confirmation form. 3. Click "Confirm" to activate/deactivate a role. 4. A role has been activated/deactivated successfully.
Alternative Flows:	None
Exceptions:	None
Business Rules:	None

Table 2.3.12.1 Active/Deactive Role

2.3.13 View Plan & Usage

UC ID and Name:	**UC-39_View Plan & Usage**		
Created By:	LongPV	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Billing & Usage" page.		
Description:	*As a Business Owner, I want to view current plan so that I can monitor subscription details and ensure the plan aligns with business needs.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Billing and usage information is successfully displayed.		
Normal Flow	**39.0 View Plan & Usage** 1. User clicks the "Plan" tab in the "Bot Details" screen. 2. System displays current plan and usage metrics.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.3.13.1 View Plan & Usage

2.3.14 Upgrade Plan

UC ID and Name:	**UC-40_Upgrade plan**		
Created By:	LongPV	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Plan" tab in the "Bot Details" screen.		
Description:	*As a Business Owner, I want to be able to upgrade my current subscription plan to access additional features and resources.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Subscription plan is successfully upgraded.		

Normal Flow	**40.0 Upgrade plan** 1. User clicks the “Plan” tab in the “Bot Details” screen. 2. System displays available subscription plans with features and pricing. 3. User selects a plan. 4. System prompts the user to confirm the upgrade. 5. User confirms the action 6. System show an QR code in the screen 7. User scan QR code and payment 8. System confirms the successful upgrade
Alternative Flows:	None
Exceptions:	None
Business Rules:	None

Table 2.3.14.1 Upgrade plan

2.3.15 Setup Credit Alerts

UC ID and Name:	**UC-41_Setup Credit Alerts**		
Created By:	LongPV	Date Created:	25/Jan/2025
Primary Actor:	Business Owner, Admin	Secondary Actors:	None
Trigger:	User navigates to the “Plan” tab in the “Bot Details” screen.		
Description:	*As a Business Owner, I want to configure an email address to receive automatic alerts when the credit balance reaches a specified threshold, so that I can proactively manage and maintain sufficient credits.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Credit alert is successfully configured and active.		
Normal Flow	**41.0 Setup Credit Alerts** 1. User clicks the “Plan” tab in the “Bot Details” screen. 2. User selects or enters the email address to receive credit alerts. 3. User sets the credit balance threshold for the alert. 4. User saves the configuration. 5. System confirms that the credit alert has been set up successfully.		
Alternative Flows:	None		
Exceptions:	None		
Business Rules:	None		

Table 2.3.15.1 Setup Credit Alerts

2.4 Admin Staff Feature

2.4.1 View Model List

UC ID and Name:	**UC-42_View Model List**		
Created By:	AnhNMT	Date Created:	28/Jan/2025

Primary Actor:	Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Model List" page.		
Description:	*As an admin staff, I want to view a list of available models to select and manage them as needed.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. The list of models is displayed successfully.		
Normal Flow	**42.0 View Model List** 1. User navigates to the "Model List" page. 2. System displays the model list with relevant details		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.4.1.1 View Model List

2.4.2 View Model Details

UC ID and Name:	**UC-43_View Model Details**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Admin Staff, Admin	Secondary Actors:	None
Trigger:	Clicks View details button from Model list screen		
Description:	*As an admin staff, I want to be able to update the details of an existing model to adjust the bot's configuration and behavior.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The model to be updated exists in the system		
Postconditions:	POST-1. Model has been updated successfully		
Normal Flow	**43.0 View Model Details** 1. User clicks “View Details” of the model list screen. 2. The system displays the model's details.		
Alternative Flows:	None		
Exceptions:	None		
Business Rules:	None		

Table 2.4.2.1 View Model Detail

2.4.3 View Invoice List

UC ID and Name:	**UC-44_View Invoice List**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Admin Staff, Admin	Secondary Actors:	None
Trigger:	User navigates to the "Invoice List" page.		

Description:	*As an admin staff, I want to view a list of invoices to track payment history and manage billing.*
Preconditions:	PRE-1. User has logged the system
Postconditions:	POST-1. The list of invoices is displayed successfully.
Normal Flow	**44.0 View Invoice List** 1. User navigates to the "Invoice List" page. 2. System show a list of all invoices
Alternative Flows:	None
Exceptions:	N/A
Business Rules:	None

Table 2.4.3.1 View Invoice List

2.4.4 Add New Invoice

UC ID and Name:	**UC-45_Add New Invoice**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Admin Staff, Admin	Secondary Actors:	None
Trigger:	User clicks the "Add New Invoice" button.		
Description:	*As an admin staff, I want to be able to add a new invoice so that I can manage and track billing records accurately.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. A new invoice is successfully created and added to the invoice list.		
Normal Flow	**45.0 Add New Invoice** 1. User clicks the "Add New Invoice" button. 2. User fills in the required invoice information. 3. User clicks "Save" to create the new invoice. 4. System validates and adds the invoice to the list.		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.4.4.1 Add New Invoice

2.4.5 Update Invoice Details

UC ID and Name:	**UC-46_Update Invoice Details**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Admin Staff, Admin	Secondary Actors:	None
Trigger:	User selects an invoice from the "Invoice List" page.		
Description:	*As an admin staff, I want to be able to update the details of an existing invoice so that I can ensure billing information is accurate and up-to-date.*		
Preconditions:	PRE-1. User has logged the system		

Postconditions:	POST-1. The selected invoice's information is successfully updated.
Normal Flow	**46.0 Update Invoice Details** 1. User navigates to the "Invoice List" page. 2. User selects an invoice to update. 3. User clicks the "Edit" button. 4. User modifies the necessary invoice fields. 5. User clicks "Save" to apply the changes. 6. System validates and updates the invoice.
Alternative Flows:	None
Exceptions:	N/A
Business Rules:	None

Table 2.4.5.1 Update Invoice Details

2.4.6 Change Invoice Status

UC ID and Name:	**UC-47_Change Invoice Status**		
Created By:	AnhNMT	Date Created:	28/Jan/2025
Primary Actor:	Admin Staff, Admin	Secondary Actors:	None
Trigger:	User selects an invoice from the "Invoice List" page.		
Description:	*As an admin staff, I want to be able to change the status of an invoice (e.g., from pending to paid) so that I can accurately reflect the invoice's current state.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. The invoice's status is successfully updated.		
Normal Flow	**47.0 Change Invoice Status** 1. User navigates to the "Invoice List" page. 2. User selects an invoice to change. 3. User clicks the "Change Status" button. 4. User selects the new status from the available options. 5. User confirms the change. 6. System updates the invoice status.		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.4.6.1 Change Invoice Status

2.5 Admin Feature

2.5.1 View User List

UC ID and Name:	**UC-48_View User List**		
Created By:	DucTA	Date Created:	2/Feb/2025

Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks “User” from the sidebar		
Description:	*As an admin, I want to be able to view all users of the system so that I can manage all users.*		
Preconditions:	PRE-1. User account have admin role		
Postconditions:	POST-1. Show a list of all users in the system and their information.		
Normal Flow	**48.0 View User List** 1. User clicks the “User” in the sidebar 2. System show a list of all users		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.5.1.1 View User List

2.5.2 Add New User

UC ID and Name:	**UC-49_Add New User**		
Created By:	DucTA	Date Created:	2/Feb/2025
Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks *Add new user* button in User list screen		
Description:	*As an admin, I want to be able to add new users so that I can add new users like a new admin account, quickly create users without registering.*		
Preconditions:	PRE-1. User account have admin role		
Postconditions:	POST-1. Add a new user to the system successfully		
Normal Flow	**49.0 Add new user** 1. User clicks the “Add new user” button. 2. The web displays a form for the admin to fill out the user's details. 3. User fills out the form and clicks the “Add” button. 4. A new user has been added to the system.		
Alternative Flows:	**49.1. Email of a new user existing in the system or email domain isn’t appropriate.** 1. Fill other email that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.5.2.1 Add new user

2.5.3 View User Details

UC ID and Name:	**UC-50_Update User Details**		
Created By:	DucTA	Date Created:	2/Feb/2025

Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks View details button from User list screen		
Description:	*As a user, I want to be able to update the details of a user to keep their information accurate and up to date.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The user to be updated exists in the system		
Postconditions:	POST-1. User details has been updated successfully		
Normal Flow	**50.0 Update User Details** 1. User clicks “View Details” of the user list screen. 2. The system displays the user's details. 3. User edits the user's details . 4. Click “Update” and the user has updated the user's details for the system.		
Alternative Flows:	**50.1. Update name that existed in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	None		
Business Rules:	None		

Table 2.5.3.1 View User Details

2.5.4 Activate/Deactivate User

UC ID and Name:	**UC-51_Activate/Deactivate User**		
Created By:	DucTA	Date Created:	2/Feb/2025
Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks Status toggle in Bot List screen		
Description:	*As a user, I want to be able to deactivate/activate users so that I can control whether the user is operational or temporarily disabled.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The user to be updated exists in the system		
Postconditions:	POST-1. Activate/deactivate a user successfully		
Normal Flow	**51.0 Activate/Deactivate User** 1. User clicks status toggle to activate/deactivate a user. 2. The system displays a confirmation form. 3. Click “Confirm” to activate/deactivate a user. 4. A user has been activated/deactivated successfully		
Alternative Flows:	None		
Exceptions:	None		
Business Rules:	None		

Table 2.5.4.1 Activate/Deactivate User

2.5.5 View Plan List

UC ID and Name:	**UC-52_View Plan List**		
Created By:	DucTA	Date Created:	2/Feb/2025
Primary Actor:	Admin	Secondary Actors:	None
Trigger:	User navigates to the "Plan List" page.		
Description:	*As an admin, I want to manage the list of plans to add, update, or delete plans as needed.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Plan list is successfully displayed.		
Normal Flow	**52.0 View Plan List** 1. User navigates to the "Plan List" page. 2. System show a list of all plan		
Alternative Flows:	None		
Exceptions:	N/A		
Business Rules:	None		

Table 2.5.5.1 View Plan List

2.5.6 Add New Plan

UC ID and Name:	**UC-53_Add New Plan**		
Created By:	DucTA	Date Created:	2/Feb/2025
Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks *Add new plan* button in Plan list screen		
Description:	*As an admin, I want to add a new plan to offer different features and pricing options for users.*		
Preconditions:	PRE-1. User has logged the system		
Postconditions:	POST-1. Add a new plan to the system successfully		
Normal Flow	**53.0 Add New Plan** 1. User clicks the “Add new plan” button. 2. The web displays a form for the admin to fill out the plan's details. 3. User fills out the form and clicks the “Add” button. 4. A new plan has been created for the system.		
Alternative Flows:	**53.1. Name of a new plan existing in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	N/A		
Business Rules:	N/A		

Table 2.5.6.1 Add New Plan

2.5.7 View Plan Details

UC ID and Name:	**UC-54_Update Plan Details**		
Created By:	DucTA	Date Created:	2/Feb/2025
Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks View details button from Plan list screen		
Description:	*As a user, I want to be able to update the details of an existing plan to ensure accurate and relevant information.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The plan to be updated exists in the system		
Postconditions:	POST-1. Plan details has been updated successfully		
Normal Flow	**54.0 Update Plan Details** 1. User clicks “View Details” of the plan list screen. 2. The system displays the plan's details. 3. User edits the plan's details . 4. Click “Update” and the user has updated the plan's details for the system.		
Alternative Flows:	**54.1. Update name that existed in the system** 1. Fill other name that isn’t existed in system 2. Return to step 4 of normal flow.		
Exceptions:	None		
Business Rules:	None		

Table 2.5.7.1 Update Plan Details

2.5.8 Activate/Deactivate Plan

UC ID and Name:	**UC-55_Activate/Deactivate Plan**		
Created By:	DucTA	Date Created:	2/Feb/2025
Primary Actor:	Admin	Secondary Actors:	None
Trigger:	Clicks Status toggle in Plan List screen		
Description:	*As a user, I want to be able to activate or deactivate a plan to control its availability for users.*		
Preconditions:	PRE-1. User has logged the system PRE-2. The plan to be updated exists in the system		
Postconditions:	POST-1. Activate/deactivate a plan successfully		
Normal Flow	**55.0 Activate/Deactivate Plan** 1. User clicks status toggle to activate/deactivate a plan. 2. The system displays a confirmation form. 3. Click “Confirm” to activate/deactivate a plan. 4. A plan has been activated/deactivated successfully		
Alternative Flows:	None		
Exceptions:	None		

Business Rules:	None

Table 2.5.8.1 Activate/Deactivate Plan

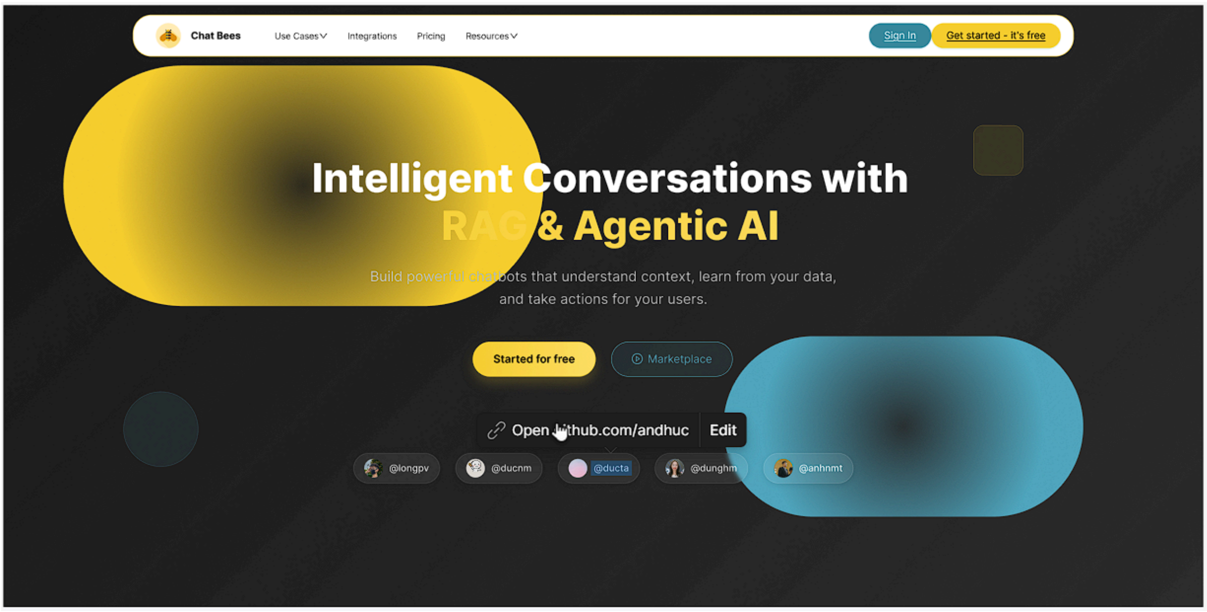
3. Functional Requirements

3.1 Common Feature

3.1.1 Landing Page

This screen allows users to know the introduction of the website.
On the screen, user can also

- Redirect to Sign In Form
- Redirect to Register Form

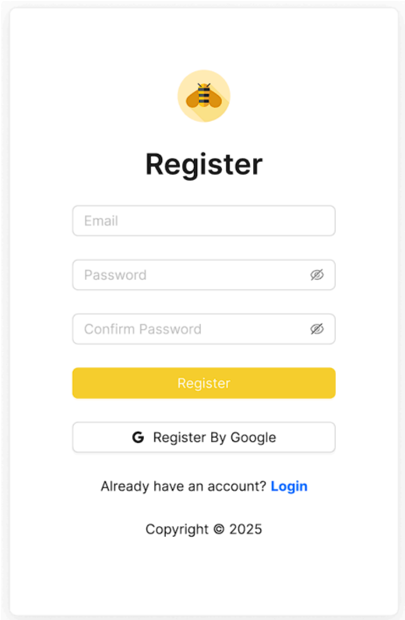


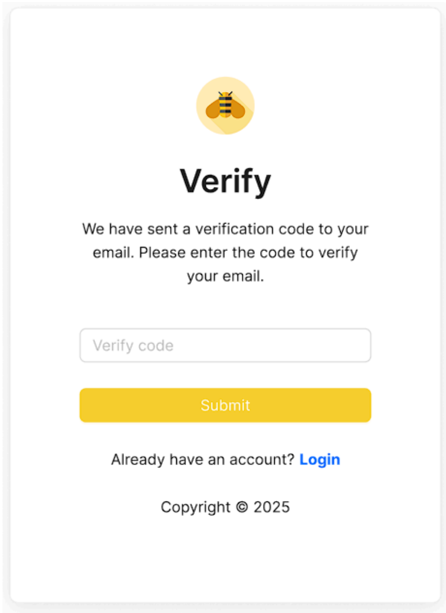
Field Description

Field Name	Field Type	Description
Sign In	Button	Click to redirect to Sign In Form
Get started - It's free	Button	Click to redirect to Register Form

3.1.2 User Register

This screen allows users to be authenticated to the system screens/functionalities.





Field Description

Field Name	Field Type	Description
Email	Text Box	Data type: valid email format (e.g., user@example.com). Required.
Password	Password Box	Data type: string, minimum 8 characters. Required.
Confirm Password	Password Box	Data type: string. Must match the Password field. Required.
Register	Button	Click to submit Register Form
Register By Google	Button	Click to sign up with google
Login	Hyperlink	Click to redirect to Login Page

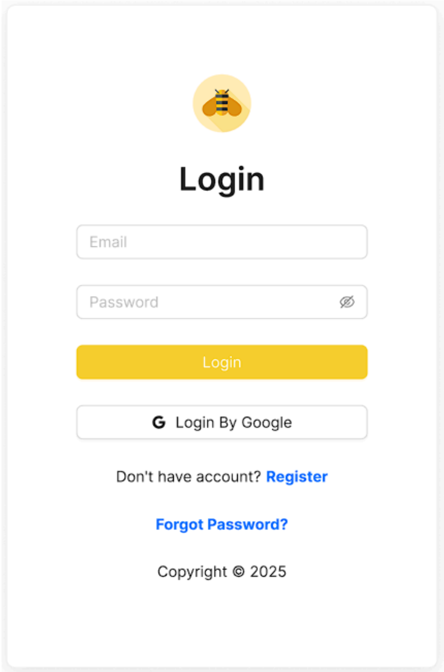
Field Description

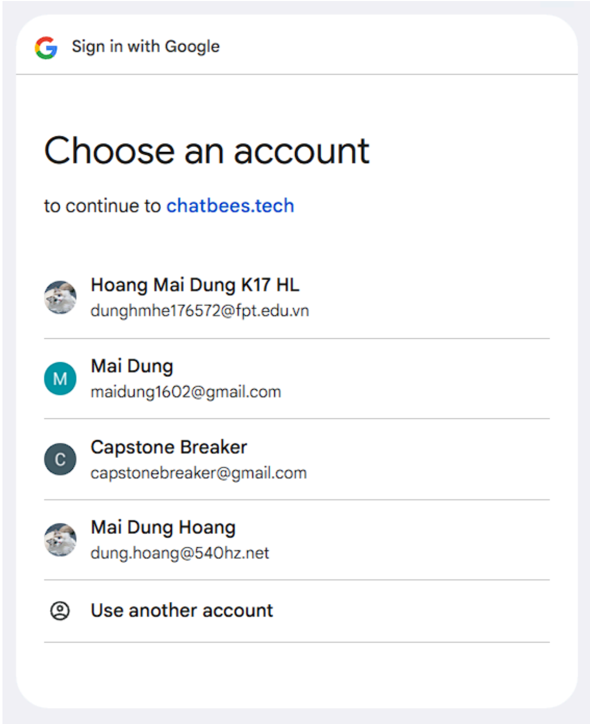
Field Name	Field Type	Description
Verify Code	Text Box	Data type: string,valid verify code.
Login	Hyperlink	Click to redirect to Login Page

3.1.3 User Login

This screen allows users to be authenticated to the system screens/functionalities.

S1_User Login screen ***S2_Select account to login (with Google)***



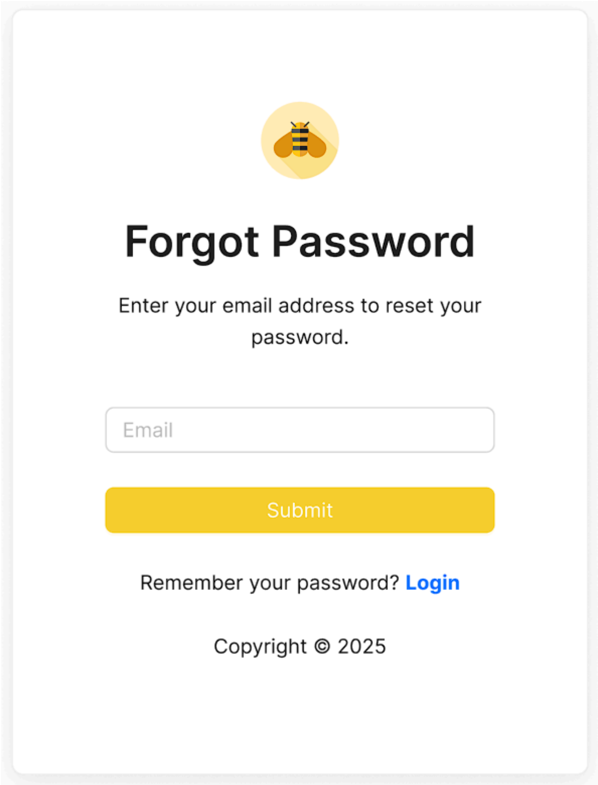


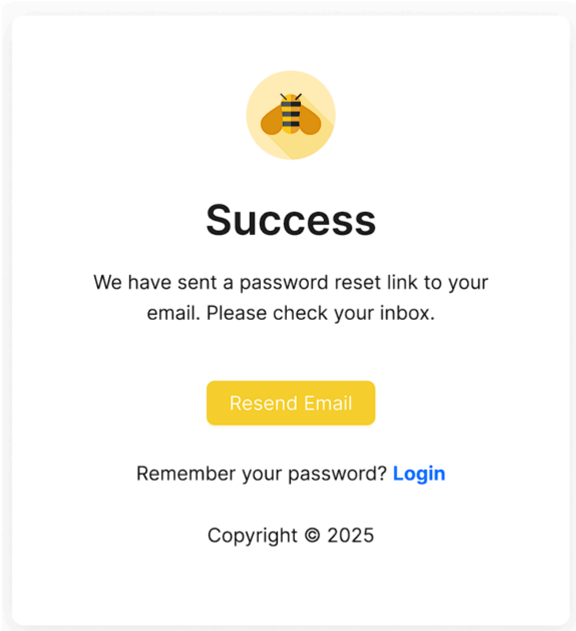
Field Description

Field Name	Field Type	Description
Email	Text Box	Data type: valid email format (e.g., user@example.com). Required.
Password	Password Box	Data type: string, minimum 8 characters. Required.
Login	Button	Click to submit Login Form
Login By Google	Button	Click to log in with google
Register	Hyperlink	Click to redirect to Register Page

3.1.4 Password Reset

This screen allows users to request a verification email to reset their password in case they've forgotten it.

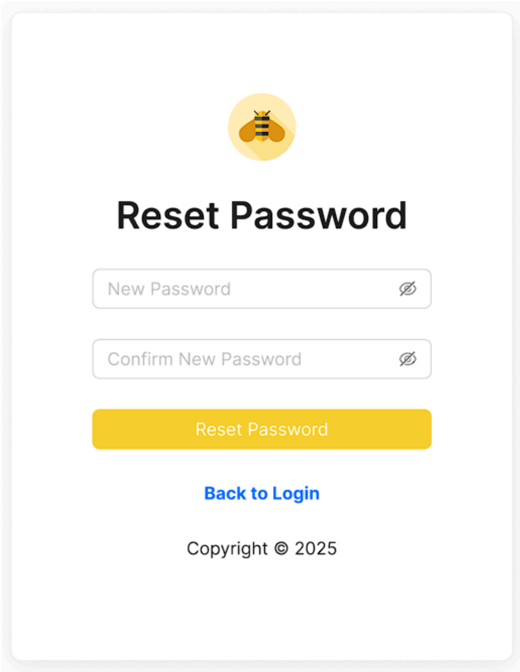




Field Description

Field Name	Field Type	Description
Email	Text Box	Data type: valid email format (e.g., user@example.com). Required.
Submit	Button	Click to submit Form
Login	Hyperlink	Click to redirect to Login Page

This screen allows users to reset a new password.

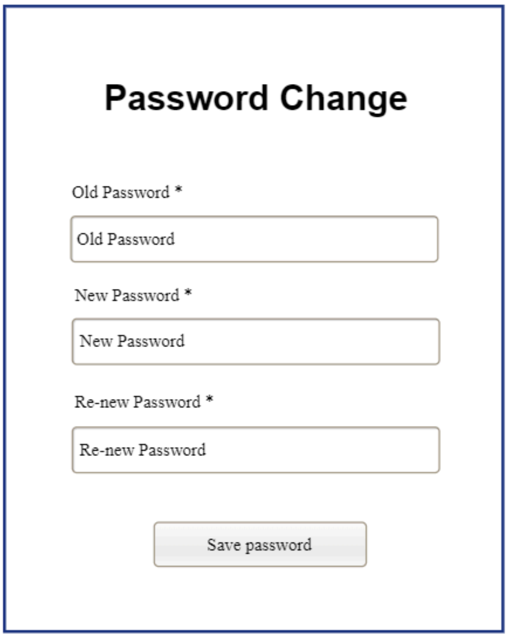


Field Description

Field Name	Field Type	Description
New Password	Password Box	Data type: string, minimum 8 characters. Required.
Confirm Password	Password Box	Data type: string. Must match the Password field. Required.
Reset Password	Button	Click to submit Password Reset Form
Back to Login	Hyperlink	Click to redirect to Login Page

3.1.5 Password Change

This screen allows users to change password.

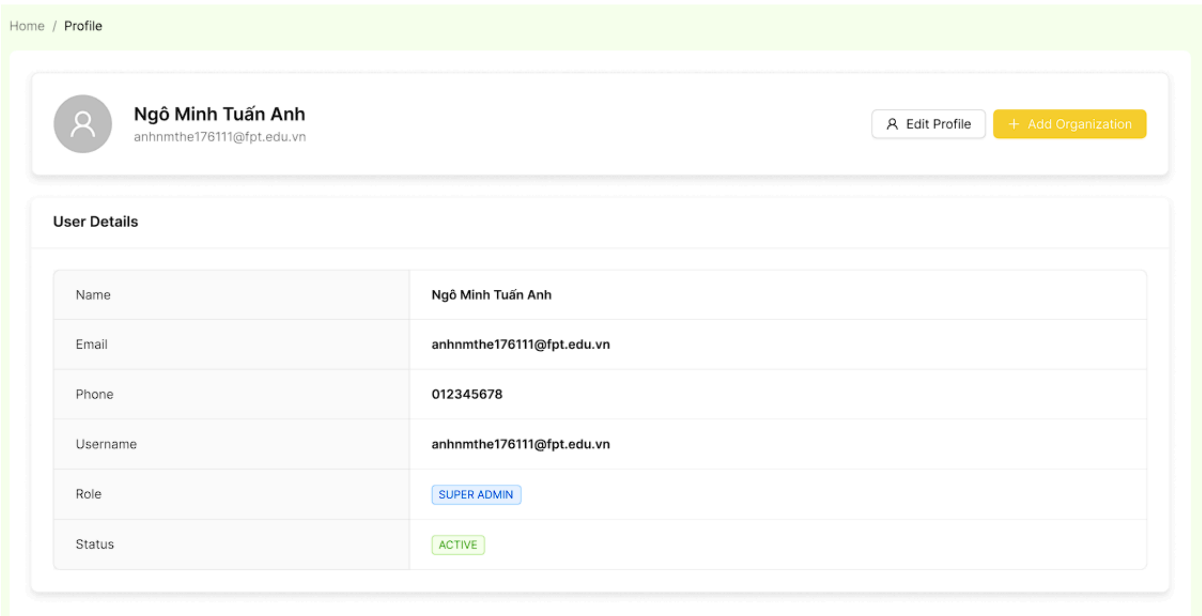


Field Description

Field Name	Field Type	Description
Old Password	Password Box	Data type: string, invalid old password. Required.
New Password	Password Box	Data type: string, minimum 8 characters. Required.
Confirm Password	Password Box	Data type: string. Must match the Password field. Required.
Save Password	Button	Click to submit Password Change Form

3.1.6 User Profile

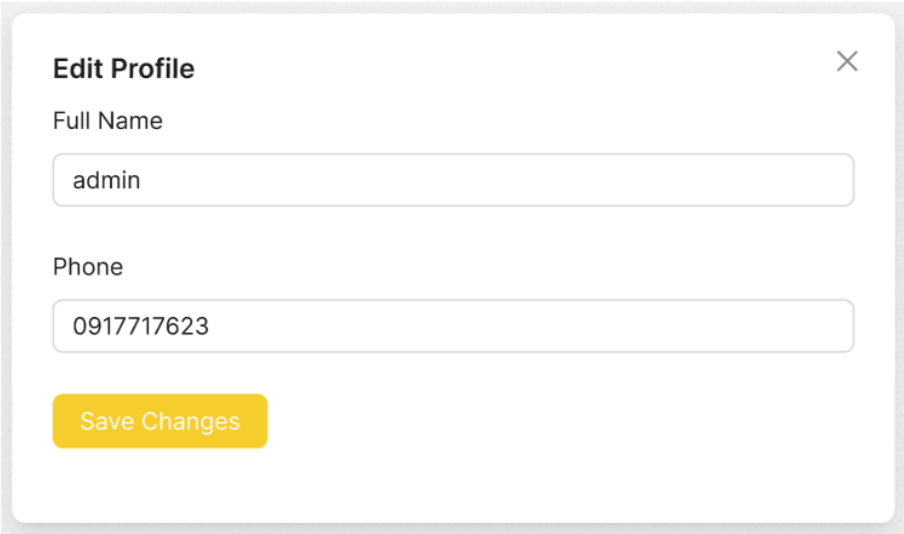
This screen allows users to view profiles.



Field Description

Field Name	Field Type	Description
Edit Profile	Button	Click to open edit Profile Form
Add Organization	Button	Click to open New Organization Form

This screen allows users to edit profiles.



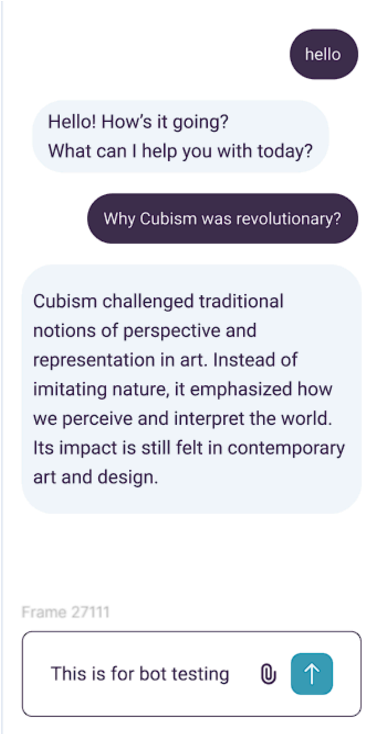
Field Description

Field Name	Field Type	Description
Full Name	Text Box	Data type: string, max length of 255 characters
Phone	Text Box	Data type: string, valid phone format.
Save Changes	Button	Click to submit Edit Profile Form

3.1.7 Chat Box

This screen allows users to chat.

-



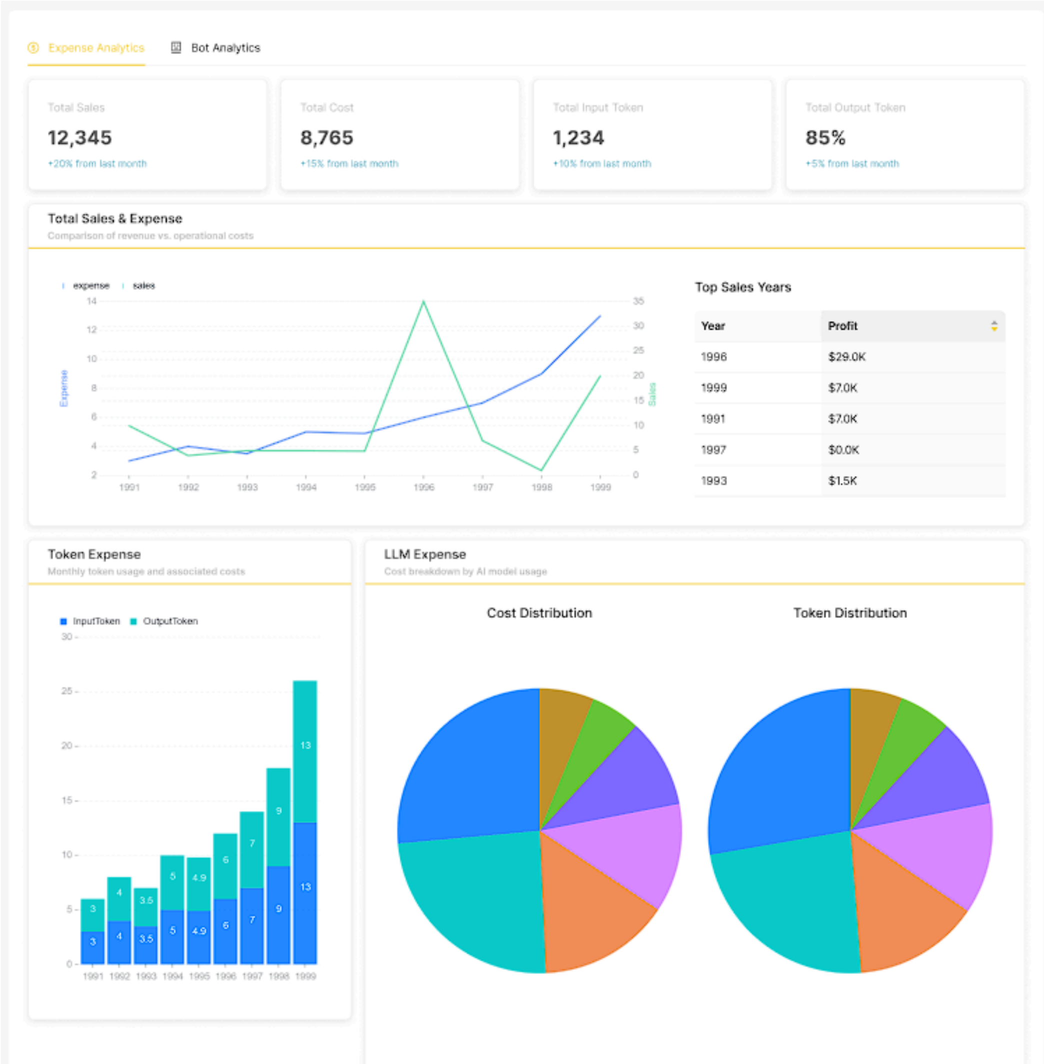
Field Description

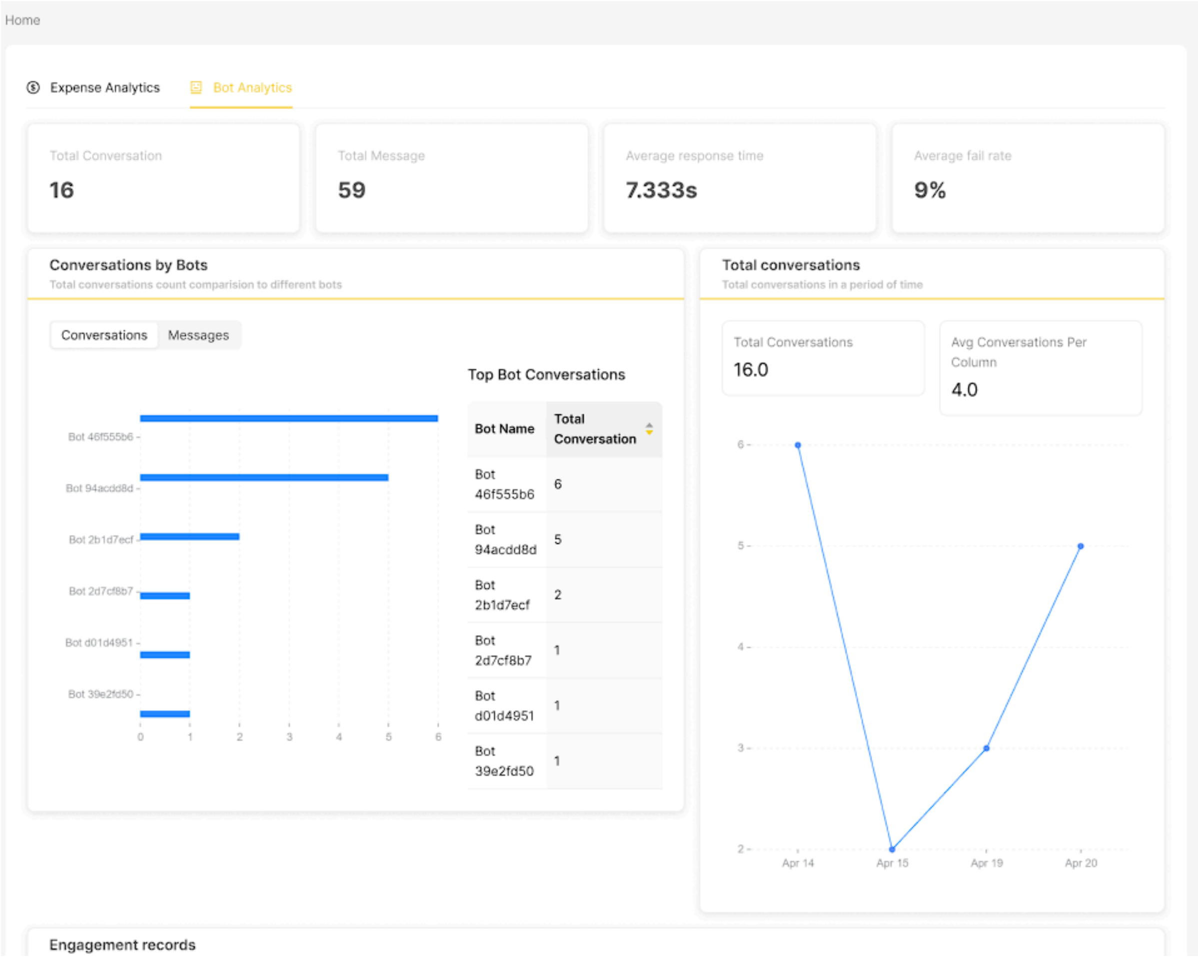
Field Name	Field Type	Description
input	Text Box	Data type: string, Text field where the user types a message
Send	Button	Click to send the message typed in the input field.

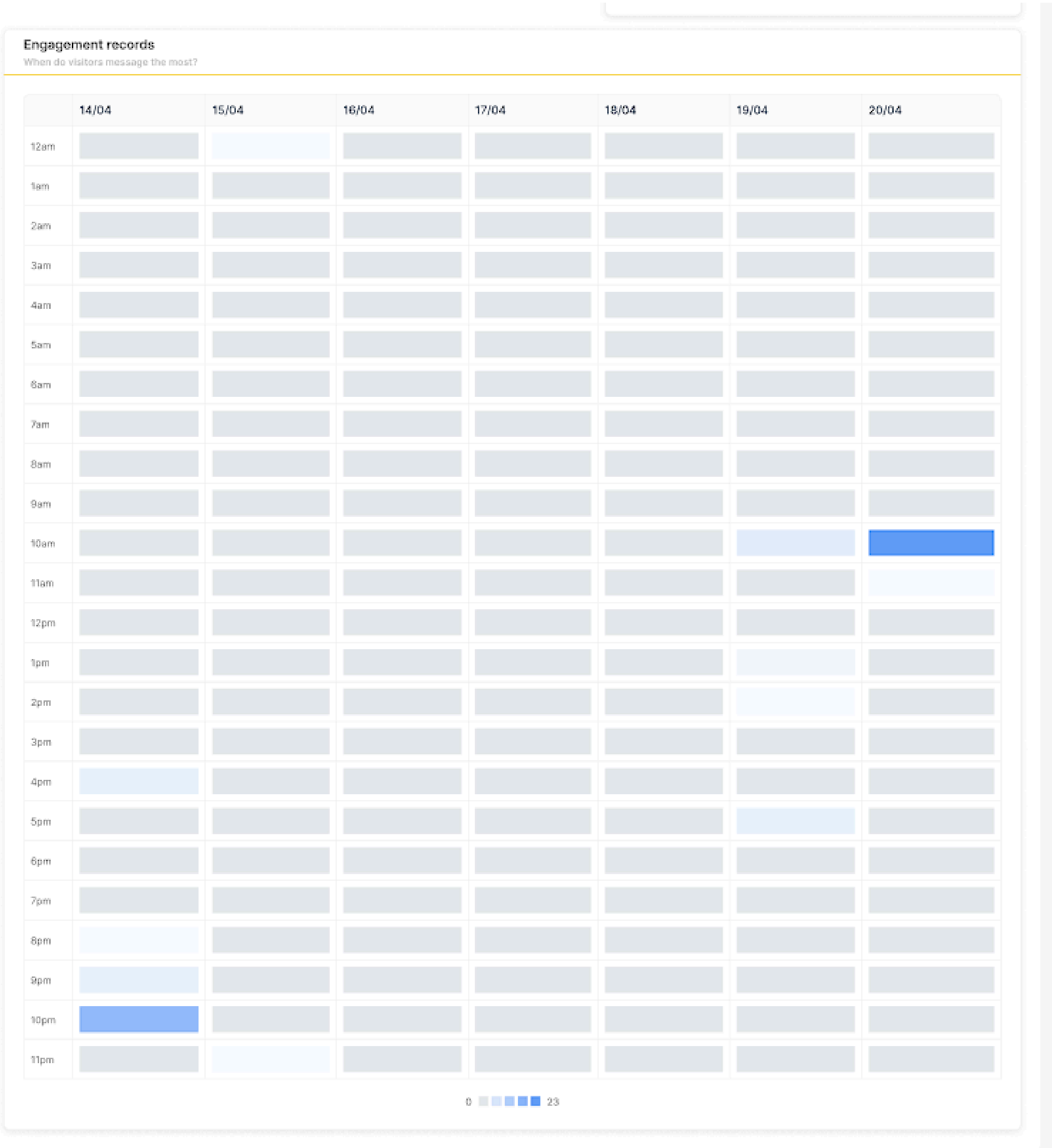
3.2 Business Staff Feature

3.2.1 Dashboard

This screen allows users to view the overview of the system, displaying essential information and key metrics to help users monitor and manage activities efficiently.




Home
Expense Analytics
Bot Analytics
Total Conversation
16
Total Message
59
Average response time
7.333s
Average fail rate
9%
Conversations by Bots
Total conversations count comparision to different bots
Conversations
Messages
Top Bot Conversations
Bot Name
Total Conversation
Bot 46f555b6
6
Bot 94acdd8d
5
Bot 2b1d7ecf
2
Bot 2d7cf8b7
1
Bot d01d4951
1
Bot 39e2fd50
1
Total conversations
Total conversations in a period of time
Total Conversations
16.0
Avg Conversations Per Column
4.0
Apr 14
Apr 15
Apr 19
Apr 20
Engagement records



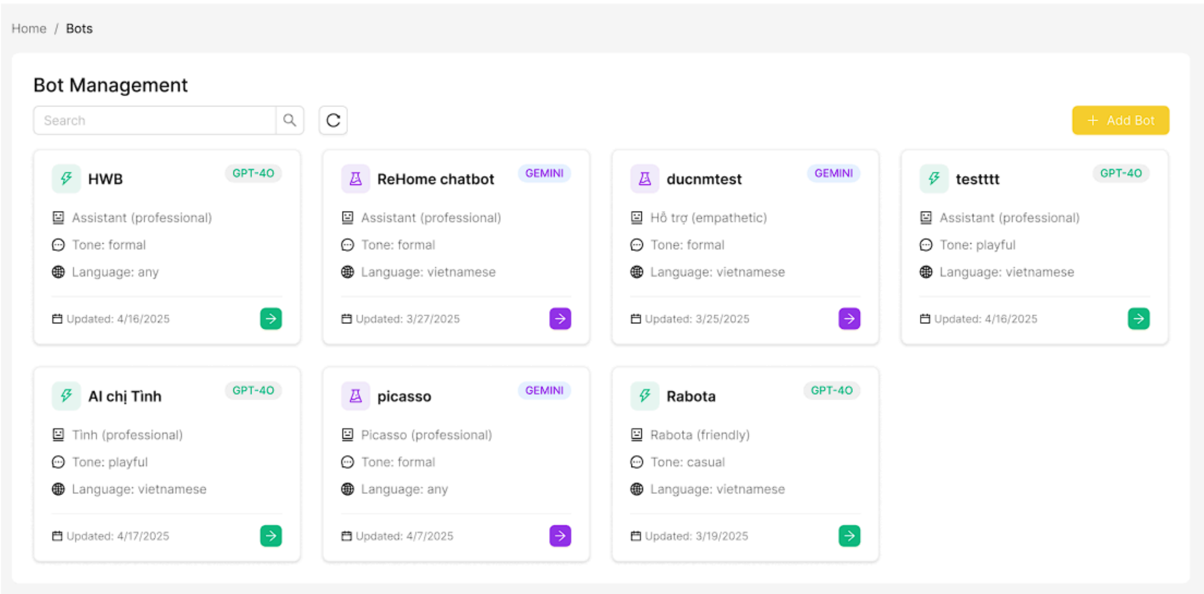
3.2.2 Bot List

This screen allows the users to:

- View Bot List: view list of bot.
- Search Bot: enter keyword(s) to search bot by name

On the screen, s/he can also

- Go to the New Bot screens for adding new by clicking Add bot.
- Go to the Bot Details screens for updating existing bot by clicking "->".

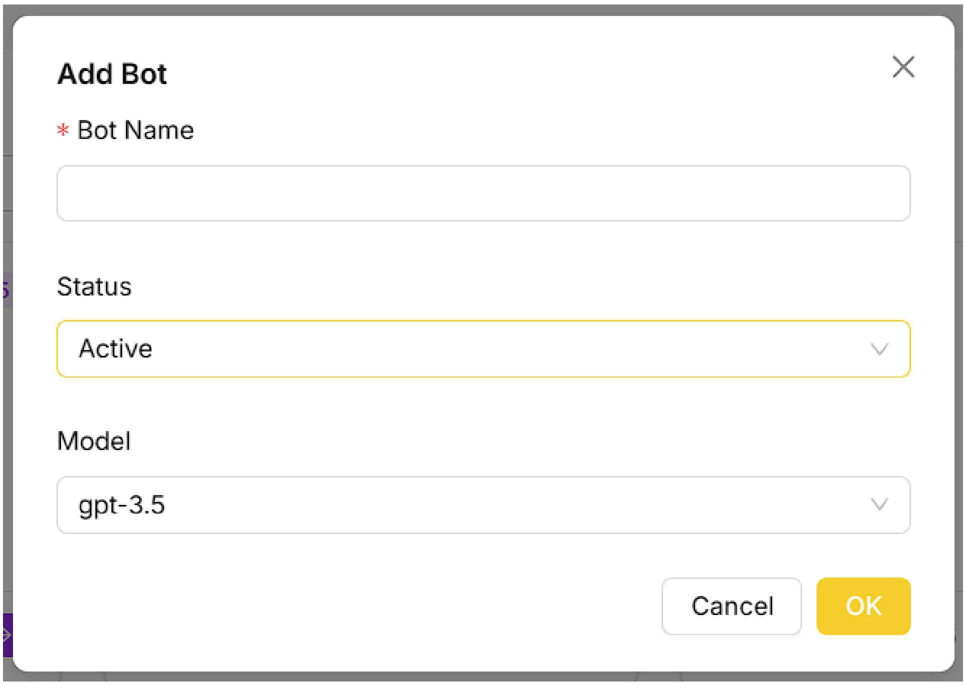


Field Description

Field Name	Field Type	Description
Search	Text Box	Data type: string. Allows users to search bot by name
Add bot	Button	Click to open the New Bot Page.

3.2.3 New Bot

This screen allows users to create a new bot



Field Description

Field Name	Field Type	Description
Bot Name	Text Box	Data type: string. Allows users to search bot by name
Ok	Button	Click to submit New Bot form

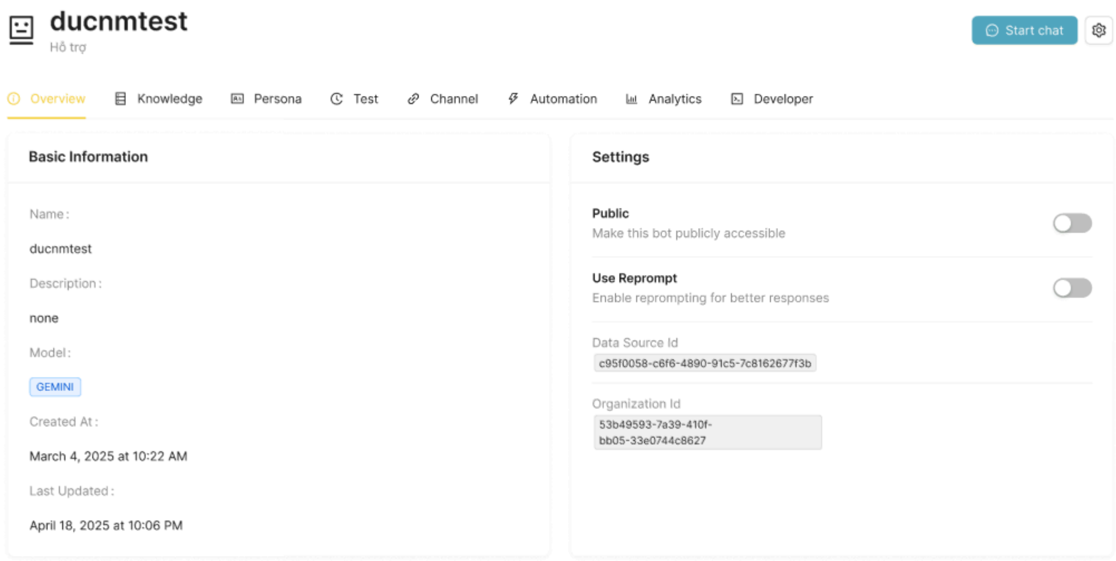
Cancel	Button	Closes the form.

3.2.4 Bot Details

This screen allows users to view bot details

3.2.4.1 Overview

This screen allows users to view overview detail of a bot



Field Description

Field Name	Field Type	Description
Start chat	Button	Click to open Chat Screen
Public	Toggle	Toggle to set the bot as public or private.
Use Reprompt	Toggle	Toggle to enable or disable the reprompt feature.

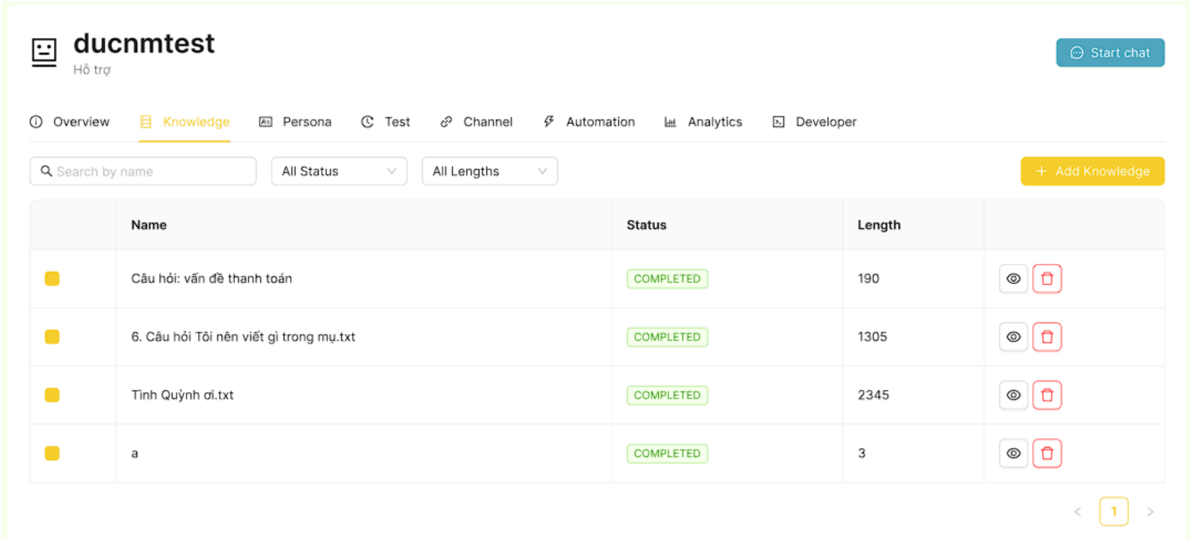
3.2.4.2 Knowledge

This screen allows the users to:

- View Data Source List: view list of datasource .
- Filter Data Source List: filter datasource by statuses, lengths
- Search Data Source: enter keyword(s) to search datasource by name

On the screen, s/he can also

- Go to the New Data Source screens for adding new by clicking Add Knowledge.
- Go to the Data Source Details screens for view datasource details by clicking Details.
- Delete Data Source: Delete datasource by clicking Delete

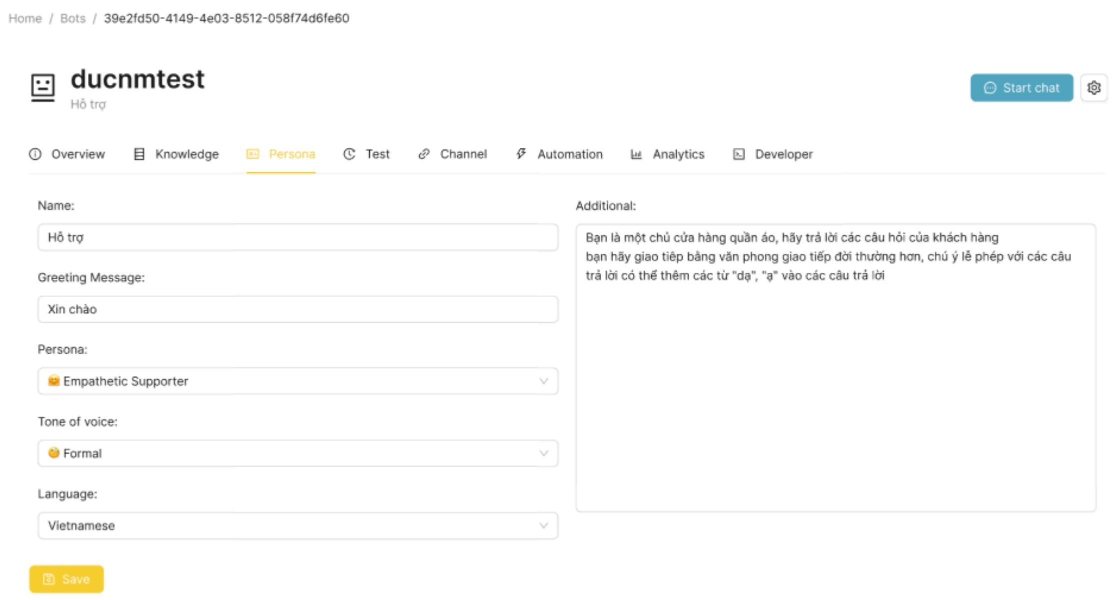


Field Description

Field Name	Field Type	Description
Search	Text Box	Data type: string. Allows users to search or filter data sources by name.
Status	Selection Box	Choose status to filter data source
Lengths	Selection Box	Choose lengths to filter datasource
Add	Button	Click to open New Datasource Page
Details	Button	Click to open the Datasource Details Page.
Delete	Button	Click to remove a datasource from the list.
List CheckBox	CheckBox	Click to choose data sources for the bot use.

3.2.4.3 Persona

This screen allows users to view personalist of a bot



Field Description

Field Name	Field Type	Description
Name	Text Box	Data type: string. The display name of the AI assistant.
Greeting Message	Text Box	Data type: string. The initial message the assistant sends to start a conversation.
Persona	Select box	Select the assistant’s personality or role (e.g., Professional Assistant).
Tone of voice	Select box	Choose the communication tone (e.g., Playful, Formal).
Language	Select box	Select the language the bot will use (e.g., Vietnamese).

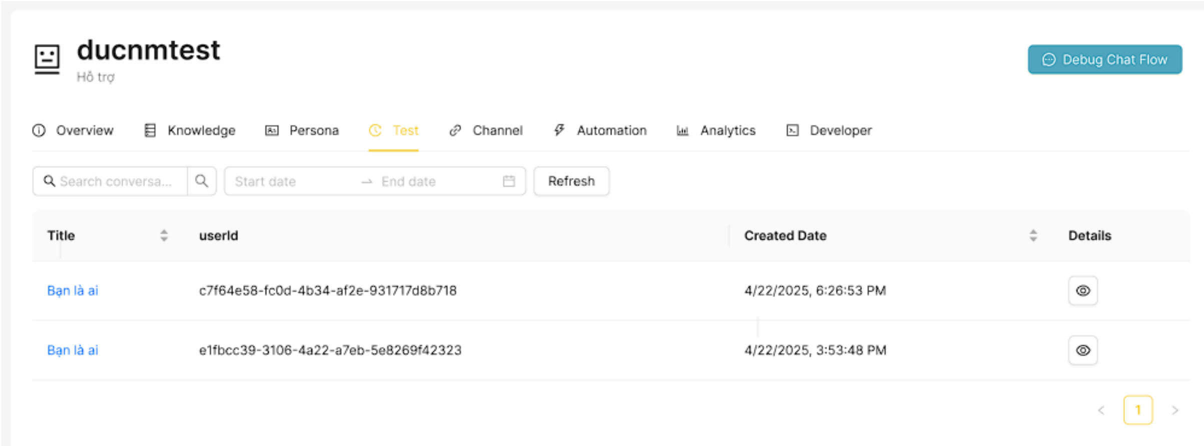
3.2.4.4 Test

This screen allows the users to:

- View Test Bot History List: view list of test bot history.
- Filter Test Bot History List: filter test bot history by start date, end date
- Search Data Source: enter keyword(s) to search test bot history by title

On the screen, s/he can also

- Go to the Test Bot screens for test bot by clicking Debug Chat Flow .
- Go to the Test Bot History Details screens for view test bot history by clicking Details.

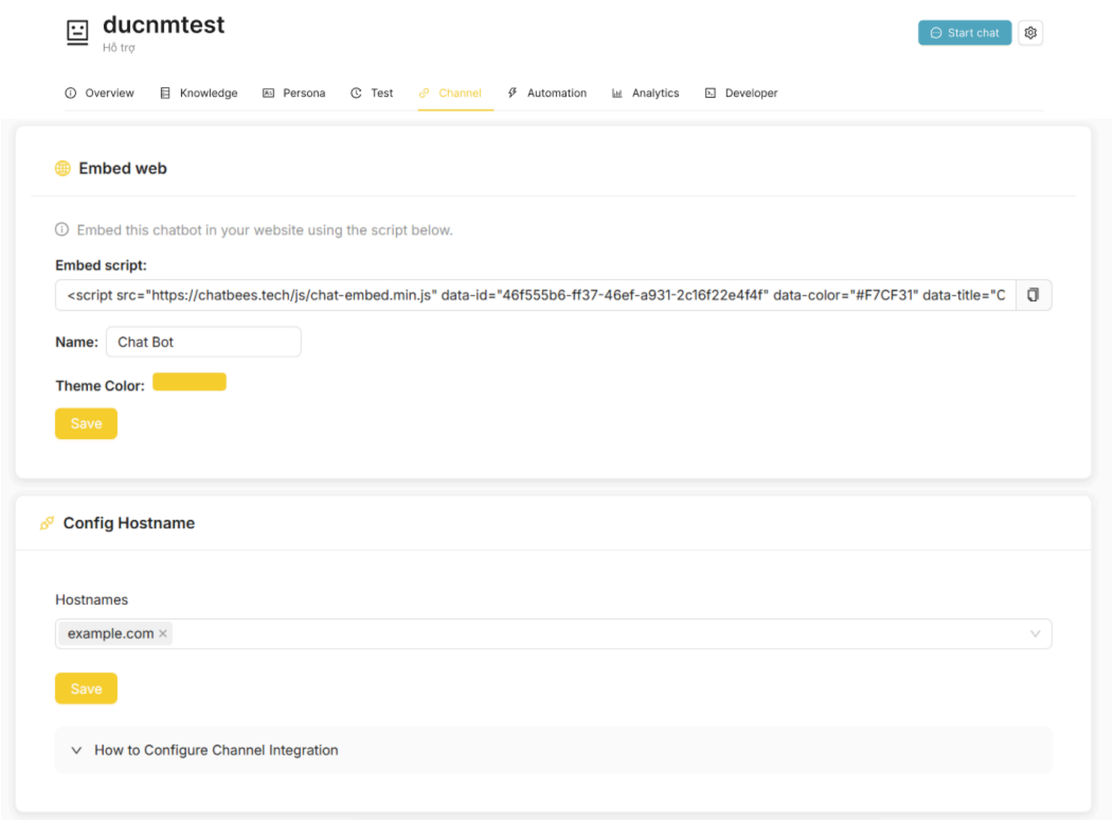


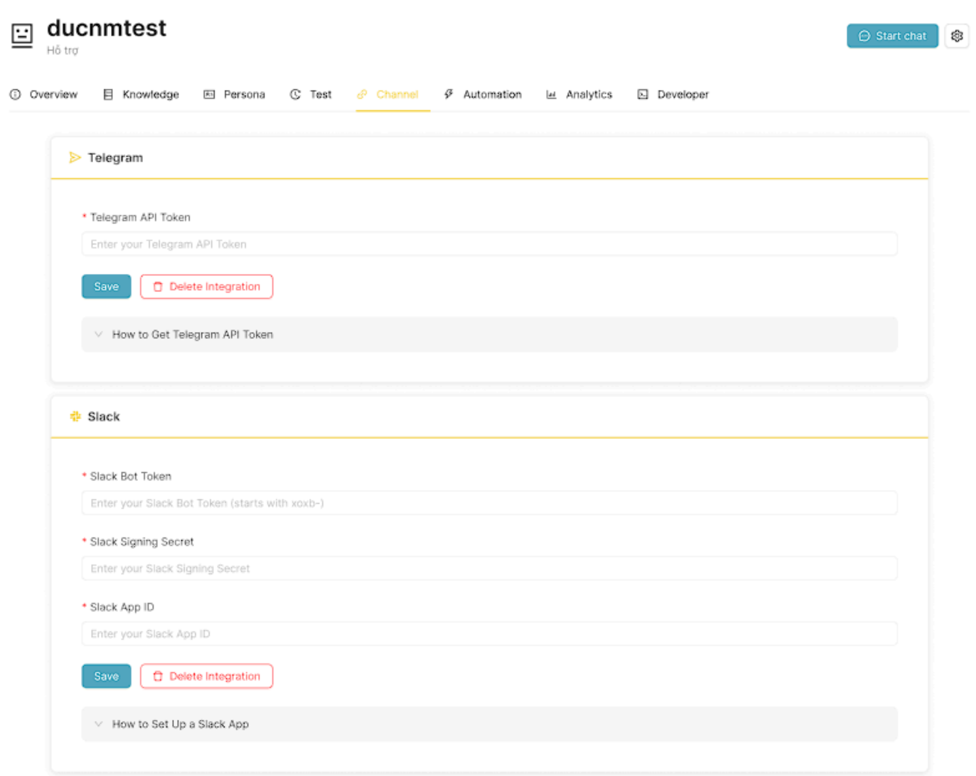
Field Description

Field Name	Field Type	Description
Debug Chat Flow	Button	Click to open Test Bot Page
Search	Text Box	Data type: string. Allows users to search or filter chat history by keywords.
Start date	Date Picker	Data type: date. Select the beginning date to filter chat history.
End date	Date Picker	Data type: date. Select the ending date to filter chat history.
Details	Button	Click to view details of conversation

3.2.4.5 Channel

This screen allows users to view and update integration of a bot

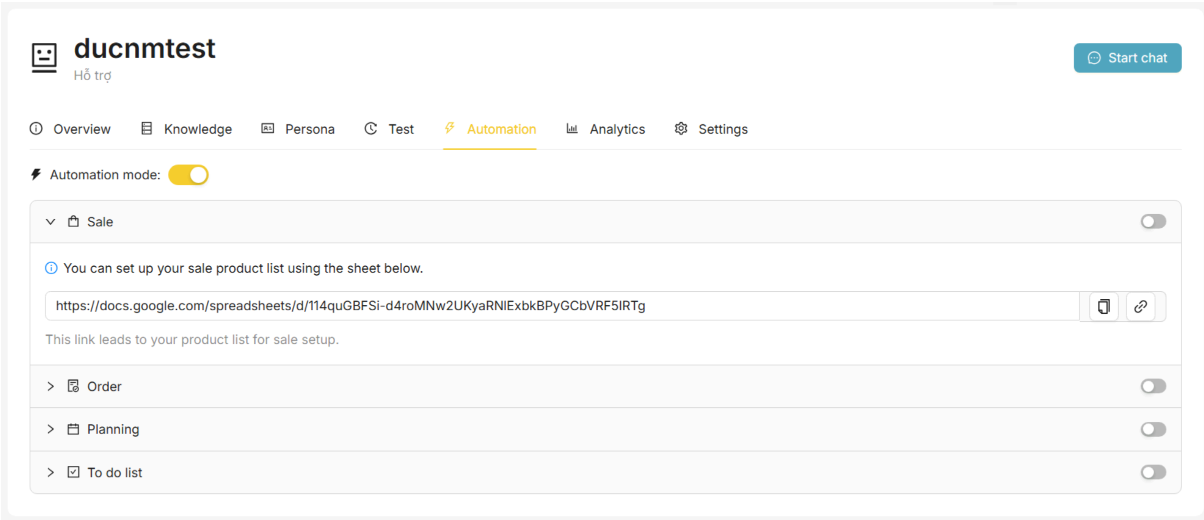

ducnmtest
Hỗ trợ
Start chat
Overview
Knowledge
Persona
Test
Channel
Automation
Analytics
Developer
Embed web
Embed this chatbot in your website using the script below.
Embed script:
<script src="https://chatbees.tech/js/chat-embed.min.js" data-id="46f555b6-ff37-46ef-a931-2c16f22e4f4f" data-color="#F7CF31" data-title="C
Name:
Chat Bot
Theme Color:
Save
Config Hostname
Hostnames
example.com
Save
How to Configure Channel Integration


ducnmtest
Hỗ trợ
Start chat
Overview
Knowledge
Persona
Test
Channel
Automation
Analytics
Developer
Telegram
* Telegram API Token
Enter your Telegram API Token
Save
Delete Integration
How to Get Telegram API Token
Slack
* Slack Bot Token
Enter your Slack Bot Token (starts with xoxb-)
* Slack Signing Secret
Enter your Slack Signing Secret
* Slack App ID
Enter your Slack App ID
Save
Delete Integration
How to Set Up a Slack App

Field Description

Field Name	Field Type	Description
Name	Text Box	Data type: string, enter the name of the chat box.
Theme Color	Color Picker	Select a color for the chat box's theme. The color can be chosen from a color palette
Hostnames	Text Box	Data type: string, input allowed hostnames for the integration.
Telegram API Token	Text Box	Data type: string, input the API token for Telegram integration.
Slack Bot Token	Text Box	Data type: string, input the bot token for Slack integration.
Slack Signing Secret	Text Box	Data type: string, input the signing secret for Slack integration.
Slack App Id	Text Box	Data type: string, input the signing secret for Slack integration.
Save	Button	Save the current integration settings.
Delete Integration	Button	Remove or delete the current integration.

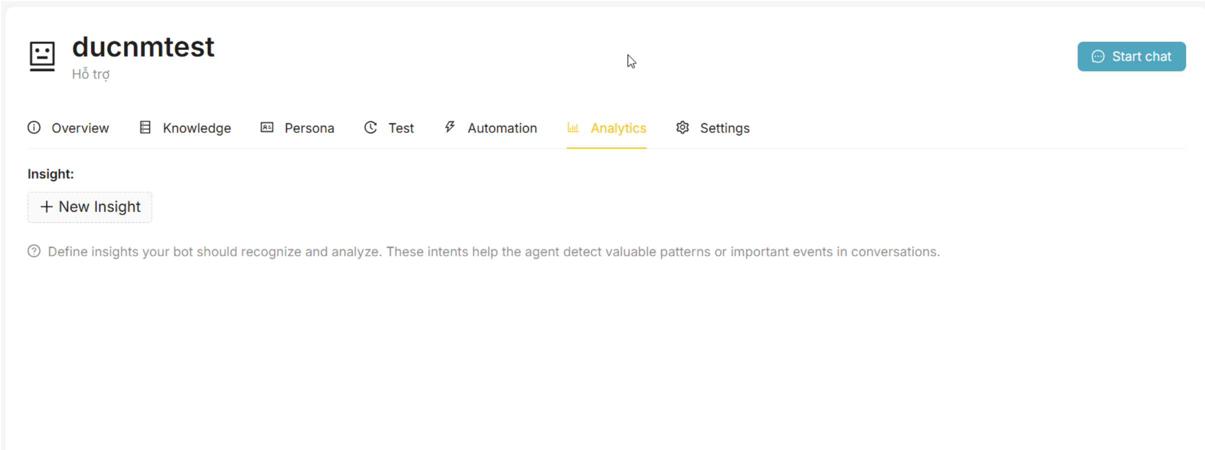
3.2.4.6 Automation



Field Description

Field Name	Field Type	Description
Automation Mode	Toggle	A switch to enable or disable the automation mode for the system.
Sale Toggle	Toggle	A switch to activate or deactivate sales automation features.
OrderToggle	Toggle	A switch to control the activation of order automation features.
Planning Toggle	Toggle	A switch to enable or disable planning functionalities.
To do list Toggle	Toggle	A switch to turn on or off the to-do list management feature.
Link Sheet	Text Box	Data type: string, a text input for entering or displaying the link to the product sheet.

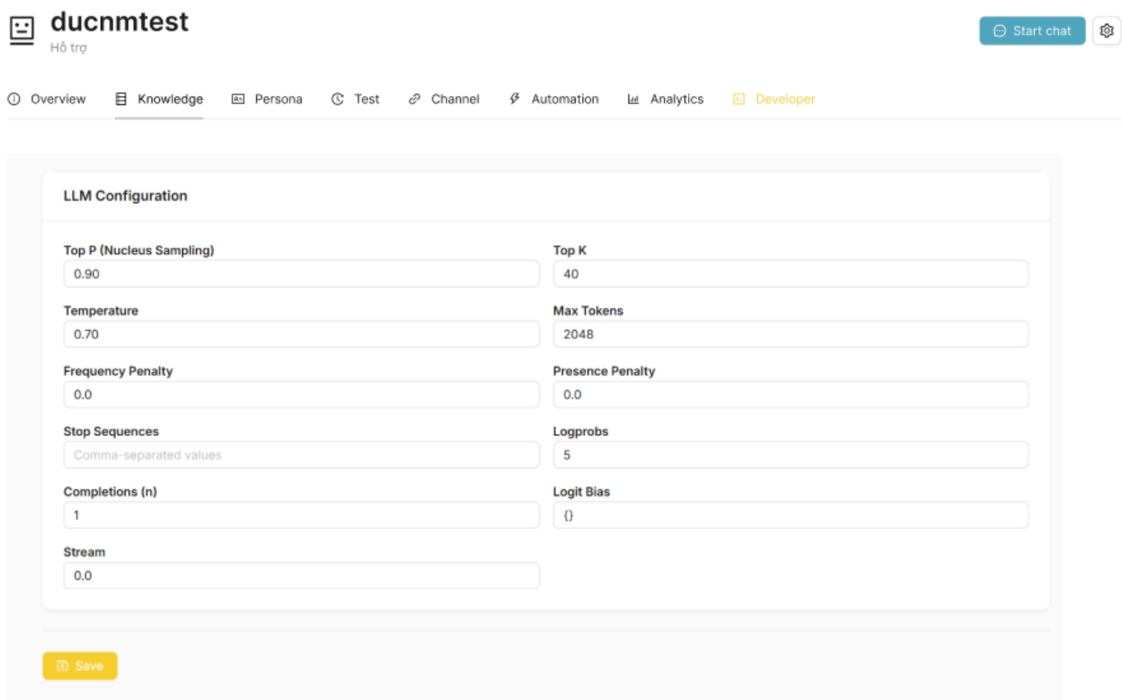
3.2.4.7 Analytics



Field Description

Field Name	Field Type	Description
New Insight	Button	Open text box to enter new insight

3.2.4.8 Developer



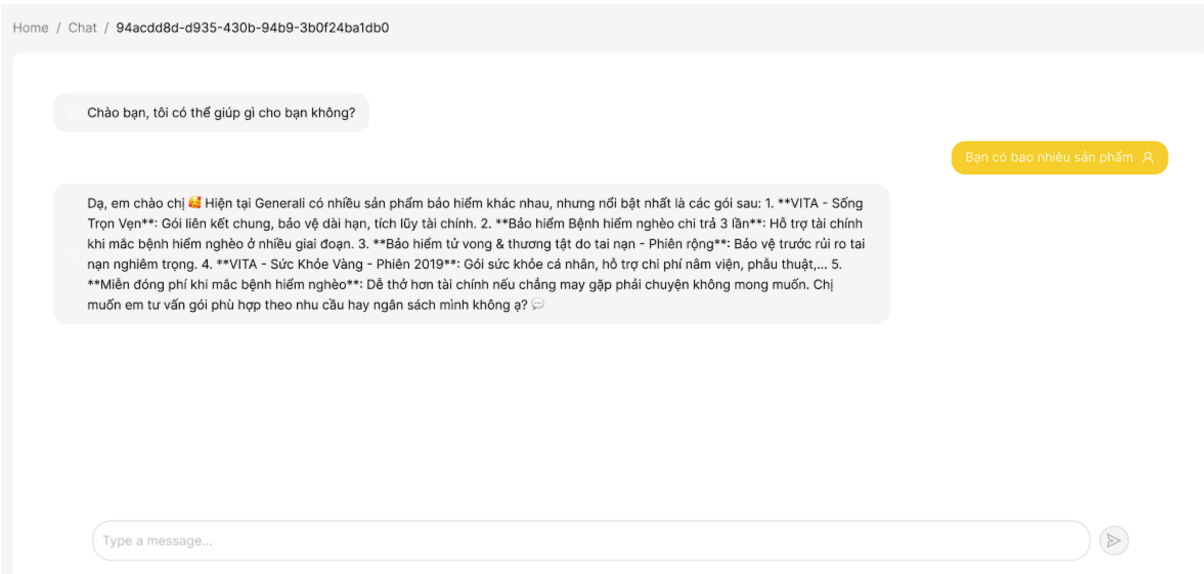
Field Description

Field Name	Field Type	Description
Top P	Number Box	Controls how many possible next words are considered.
Top K	Number Box	Limits choices to the top K most likely words.

Temperature	Number Box	Controls how random the output is.
Max Tokens	Number Box	Sets the maximum length of the response.
Frequency Penalty	Number Box	Decreases repeated words.
Presence Penalty	Number Box	Encourages talking about new topics.
Stop Sequences	Text Box	Tells the model when to stop writing.
Logprobs	Number Box	Number of most likely tokens to include log probabilities for.
Completions (n)	Number Box	Number of completions to generate per prompt.
Logit Bias	Text Box	If enabled (1), streams partial outputs as they are generated.
Stream	Text Box	Biases the model's likelihood to select certain tokens more or less. Example: {"50256": -100}.
Save	Button	Save the current developer settings.

3.2.5 Chat

This screen allows users to live chat and view chat history

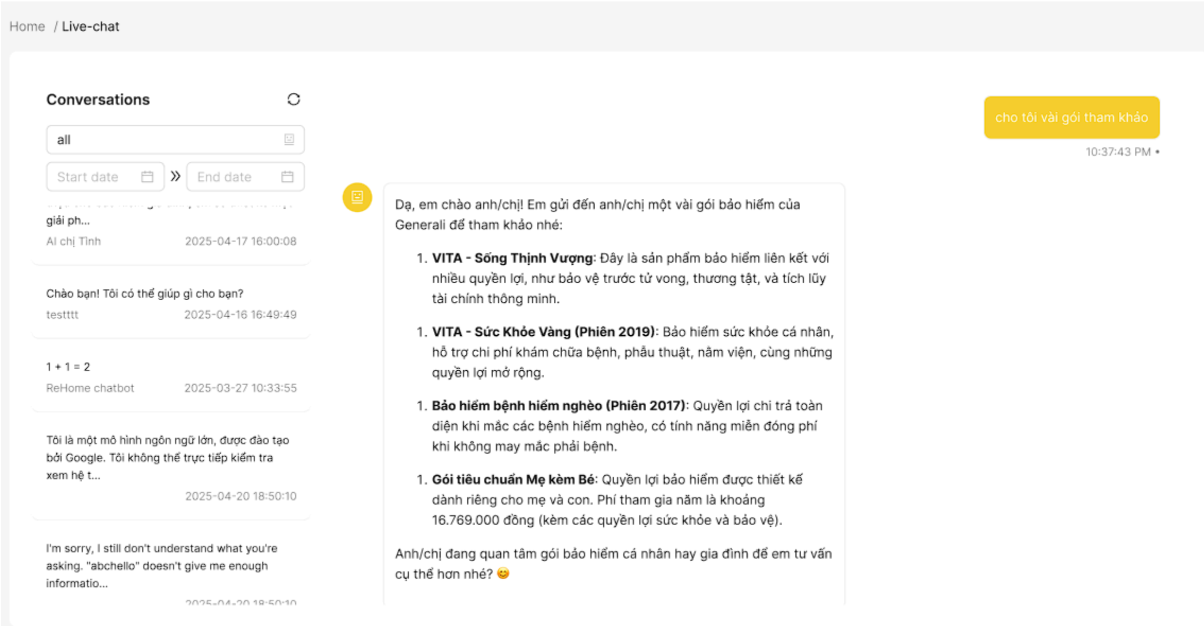


Field Description

Field Name	Field Type	Description
input	Text Box	Data type: string, Text field where the user types a message
Send	Button	Click to send the message typed in the input field.

3.2.6 Live Chat

This screen allows users to view chat tracking

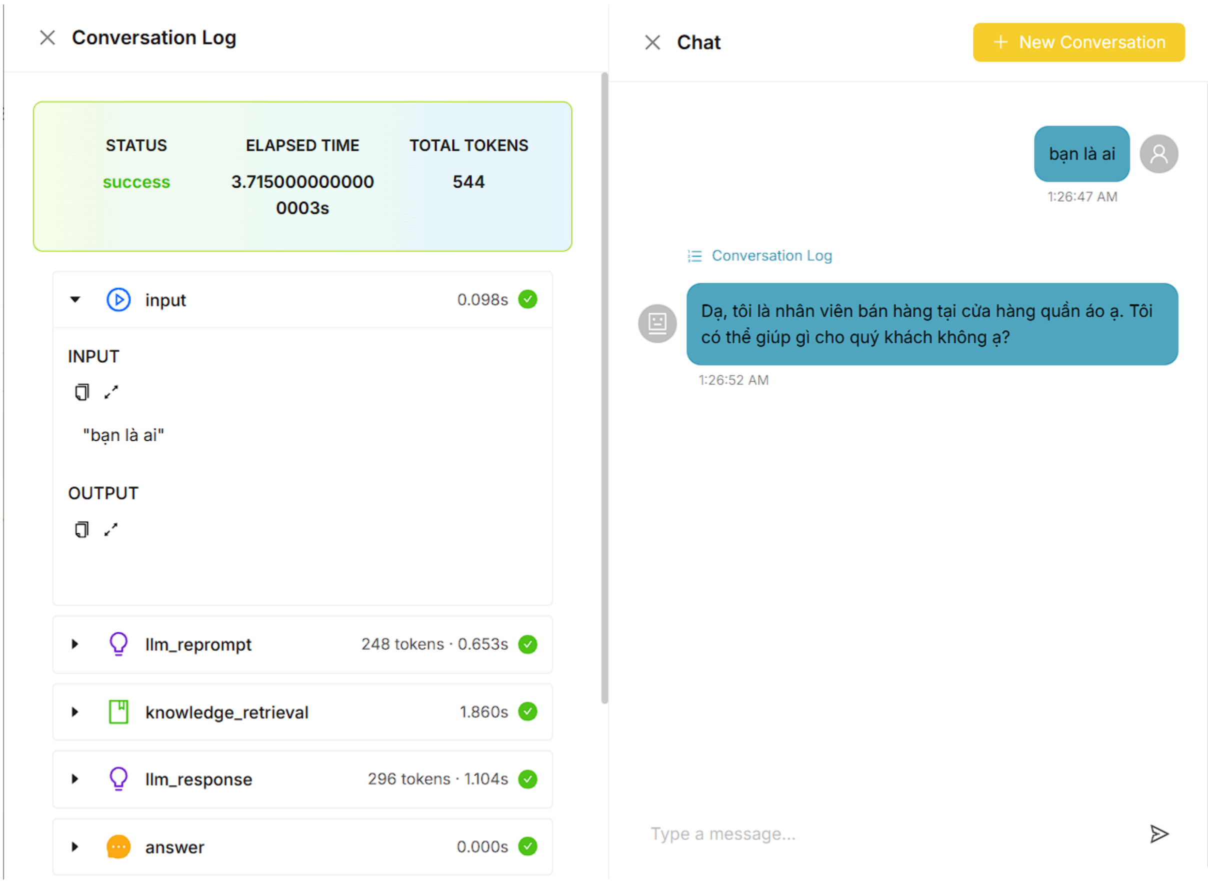


Field Description

Field Name	Field Type	Description
Bot	Selection Box	Initial data values: all bot. Choose a bot to filter chat history.
Start date	Date Picker	Data type: date. Select the beginning date to filter chat history.
End date	Date Picker	Data type: date. Select the ending date to filter chat history.

3.2.7 Test Bot

This screen allows users to test bot

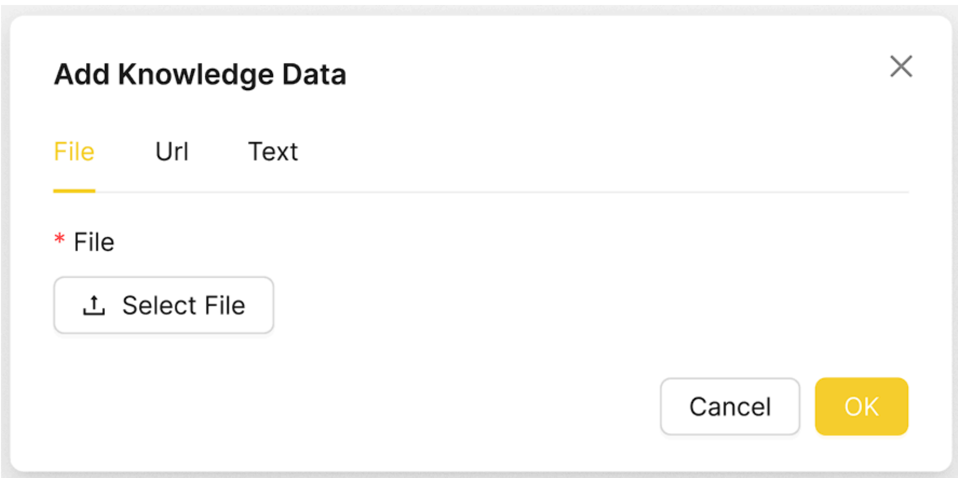


Field Description

Field Name	Field Type	Description
input	Text Box	Data type: string, Text field where the user types a message
Send	Button	Click to send the message typed in the input field.
New Conversation	Button	Click to create a new conversation

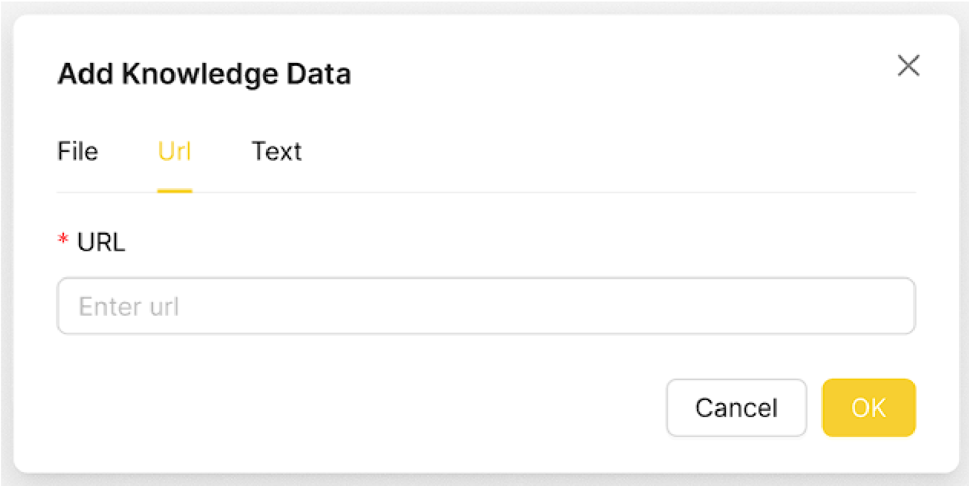
3.2.8 New Data Source

This screen allows users to add data source



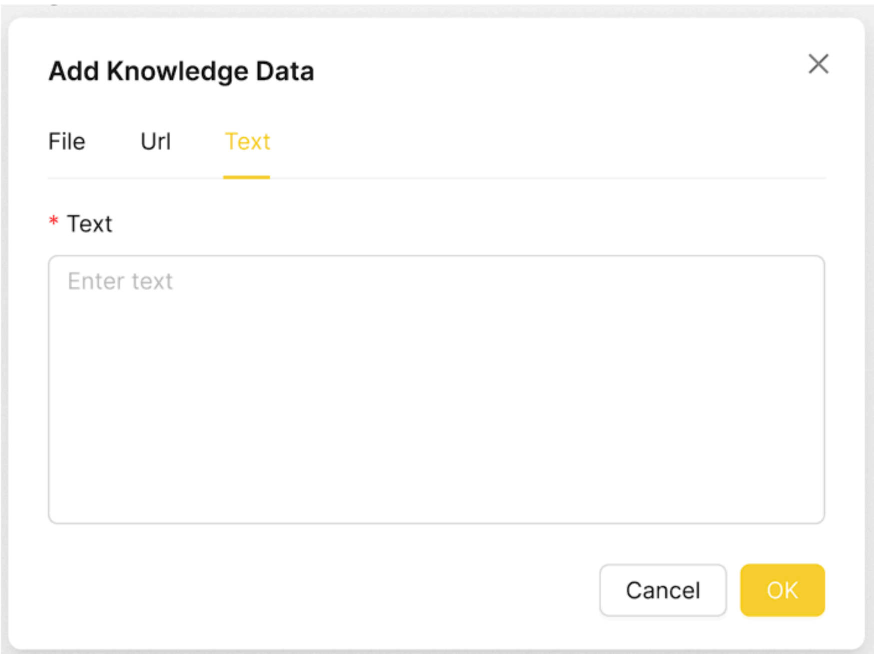
Field Description

Field Name	Field Type	Description
File	File Uploader	Data type: file. Required. Upload a file to be used as a knowledge source. Supported formats include .txt, .pdf, .doc, .docx, .csv, .xls, .xlsx



Field Description

Field Name	Field Type	Description
Url	Text Box	Data type: string. Required. Enter a valid web address to extract knowledge data from that webpage. Example: https://example.com/help

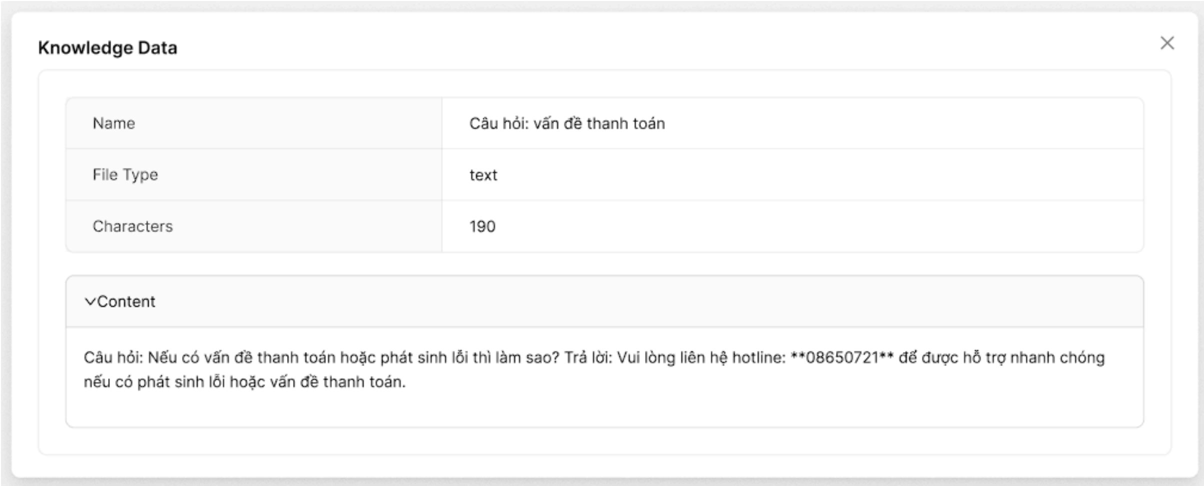


Field Description

Field Name	Field Type	Description
Text	Text Box	Data type: string..Required. Paste or write knowledge content directly in plain text format (e.g., instructions, FAQs).

3.2.9 Data Source Details

This screen allows users to view data source details



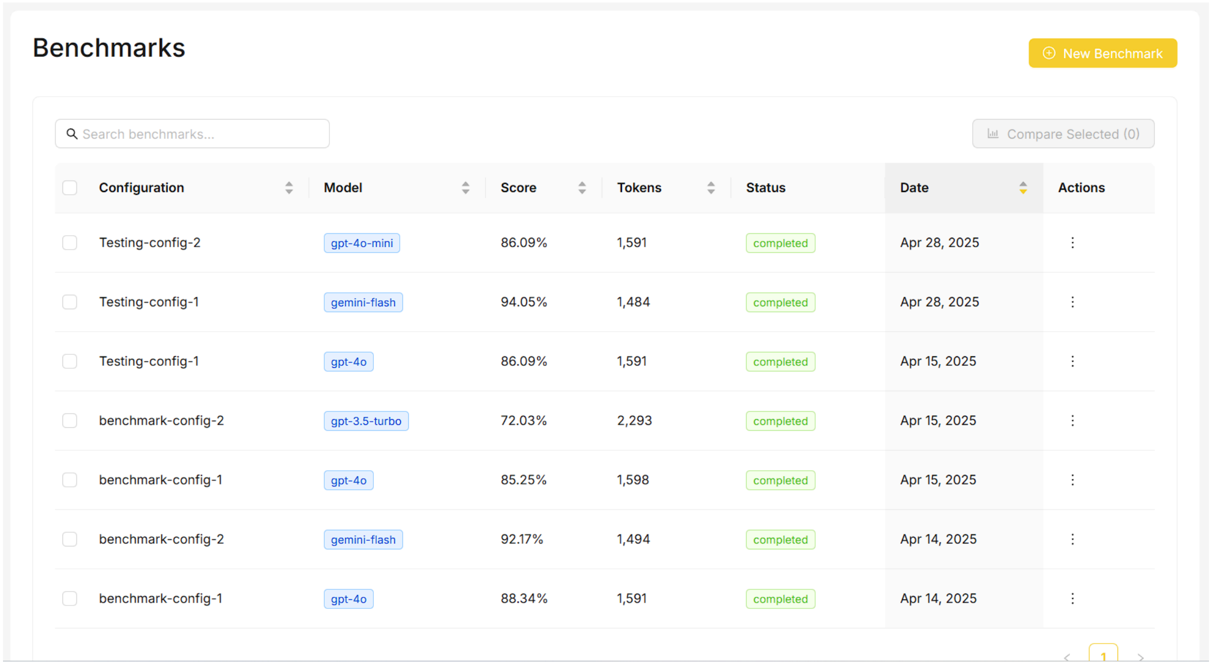
3.2.10 Benchmark History

This screen allows the users to:

- View Benchmark History List: view list of benchmark history.
- Search Benchmark: enter keyword(s) to search benchmark results by name

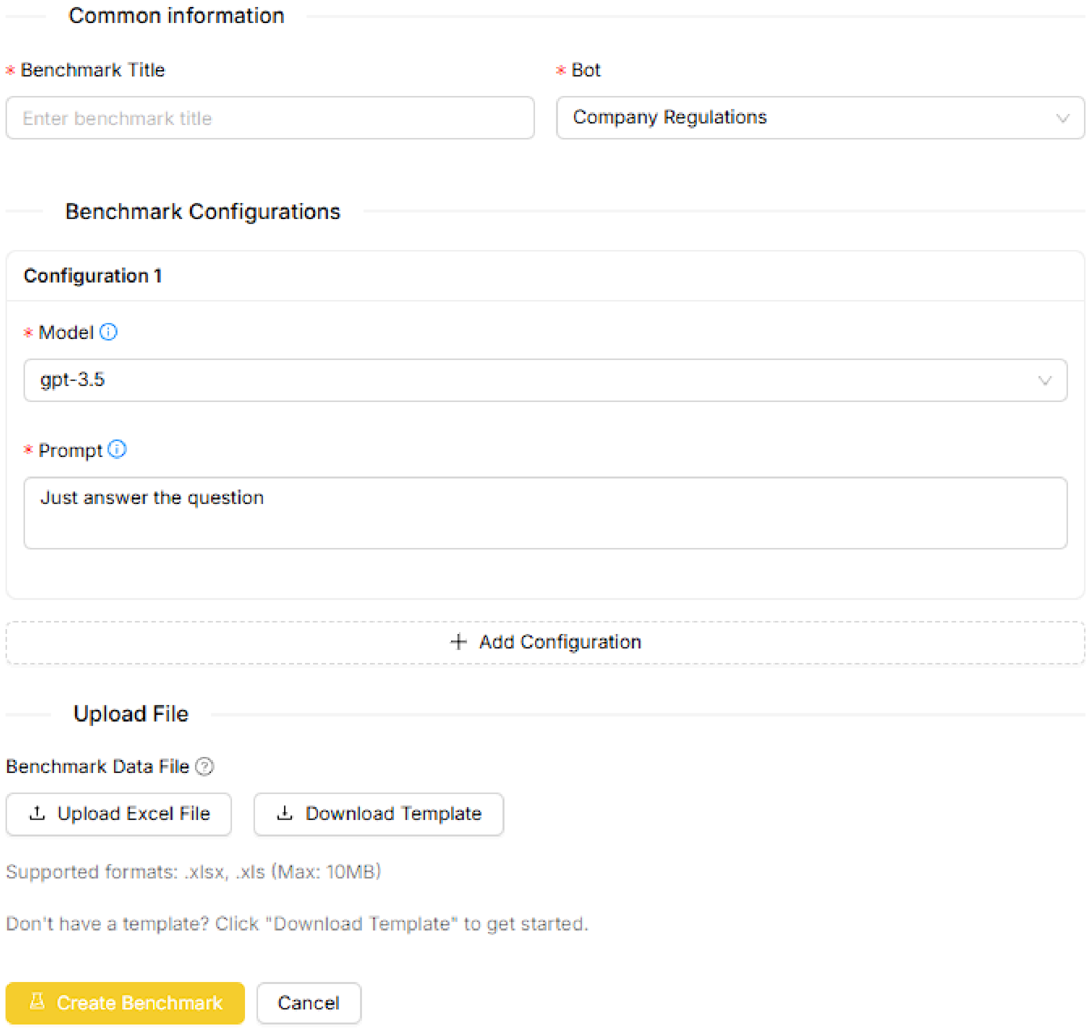
On the screen, s/he can also

- Go to the New Benchmark screens for adding new by clicking Add bot.
- Go to the Benchmark Details screens for view details of benchmark results.



Configuration	Model	Score	Tokens	Status	Date	Actions
Testing-config-2	gpt-4o-mini	86.09%	1,591	completed	Apr 28, 2025	⋮
Testing-config-1	gemini-flash	94.05%	1,484	completed	Apr 28, 2025	⋮
Testing-config-1	gpt-4o	86.09%	1,591	completed	Apr 15, 2025	⋮
benchmark-config-2	gpt-3.5-turbo	72.03%	2,293	completed	Apr 15, 2025	⋮
benchmark-config-1	gpt-4o	85.25%	1,598	completed	Apr 15, 2025	⋮
benchmark-config-2	gemini-flash	92.17%	1,494	completed	Apr 14, 2025	⋮
benchmark-config-1	gpt-4o	88.34%	1,591	completed	Apr 14, 2025	⋮

3.3.11 New Benchmark

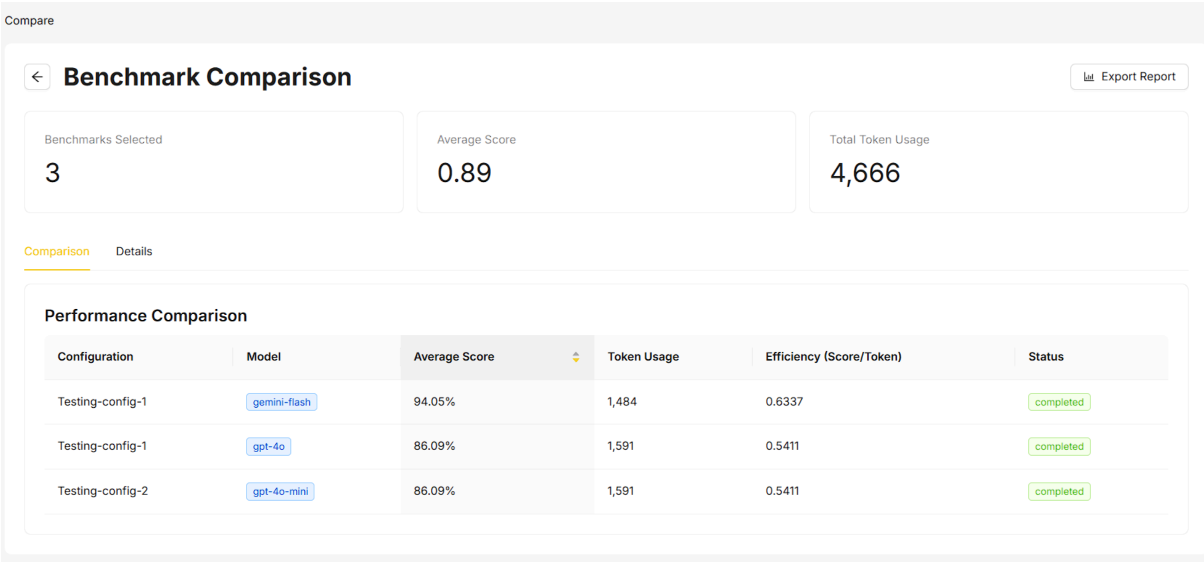


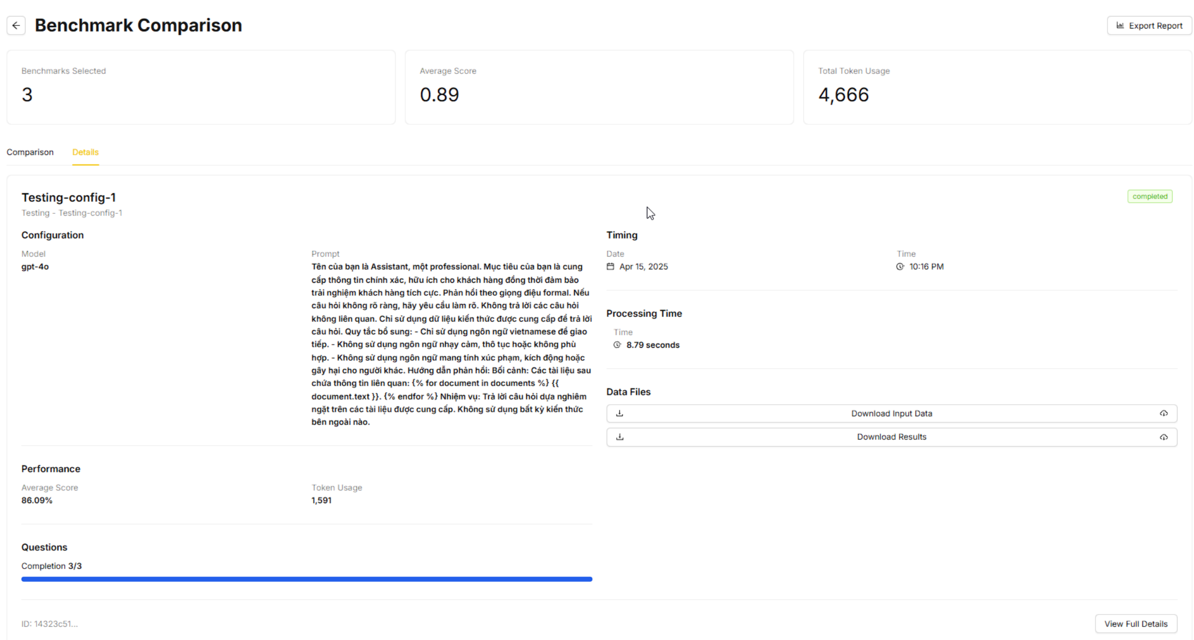
Field Description

Field Name	Field Type	Description
Title	Text Box	Data type: string. Required. Text for test title
Bot	Selection Box	Initial data values: first bot in the list. Choose a bot to run benchmark test.
LLM Model	Selection Box	Initial data values: first LLM model in the list. Choose a LLM model to run a benchmark test.
Prompt	Text Box	Data type: string..Required. Text for test prompt running test benchmark
File	File Uploader	Data type: file. Required. Upload a file to be used as a question source for test evaluation. Supported formats include .txt, .pdf, .doc, .docx, .csv, .xls, .xlsx

3.2.12 Benchmark Details

This screen allow the user to view and compare bot evaluation tests





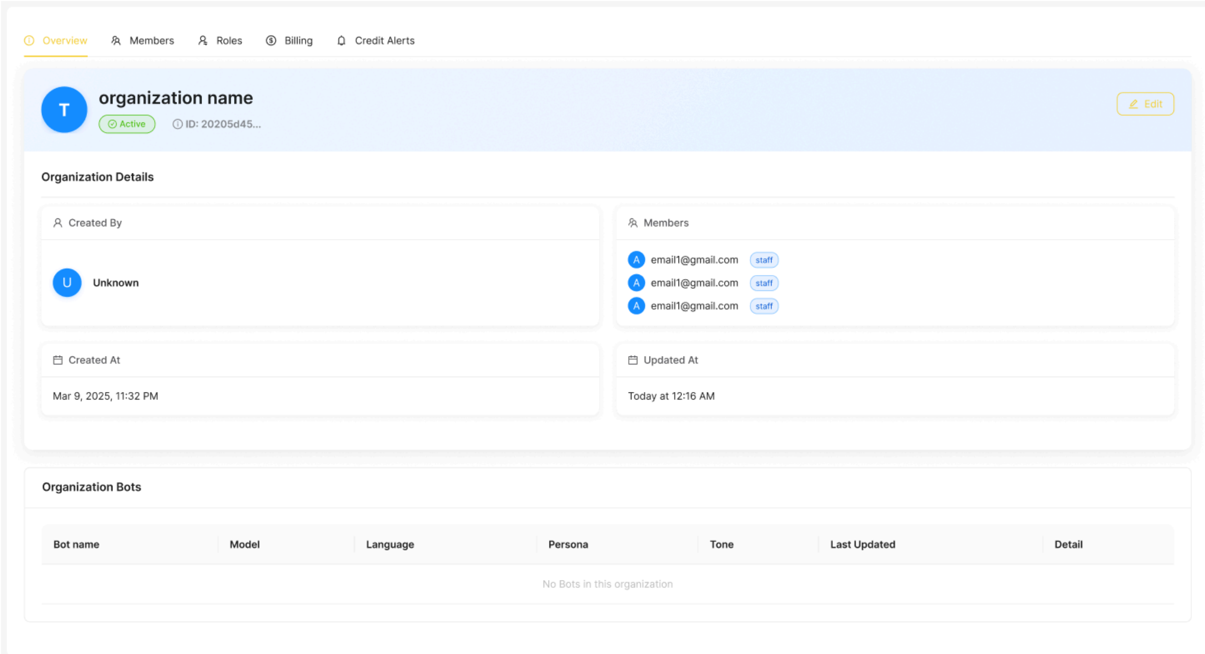
3.3 Business Owner Feature

3.3.1 Organization Details

This screen shows details of an organization

3.3.1.1 Overview

This screen shows overview of an organization



Field Description

Field Name	Field Type	Description
Edit	Button	Click to open Edit Organization Form

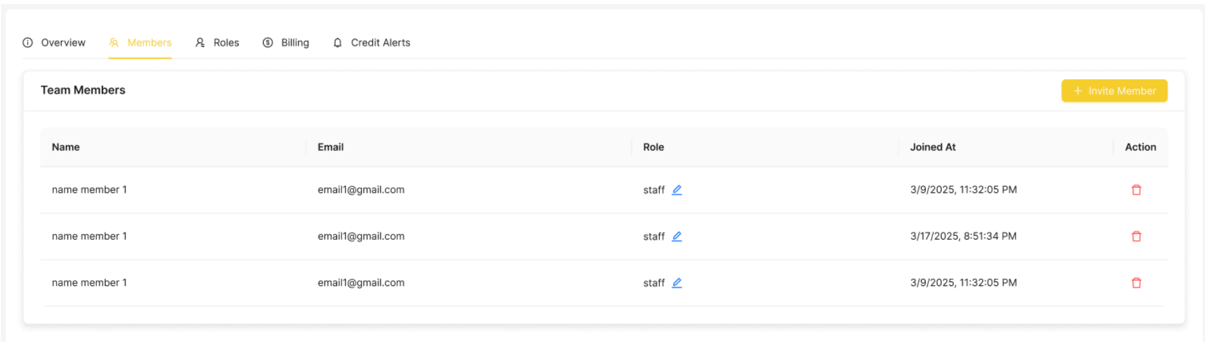
3.3.1.2 Member List

This screen allows the users to:

- View Member List: view member list of the organization.

On the screen, s/he can also

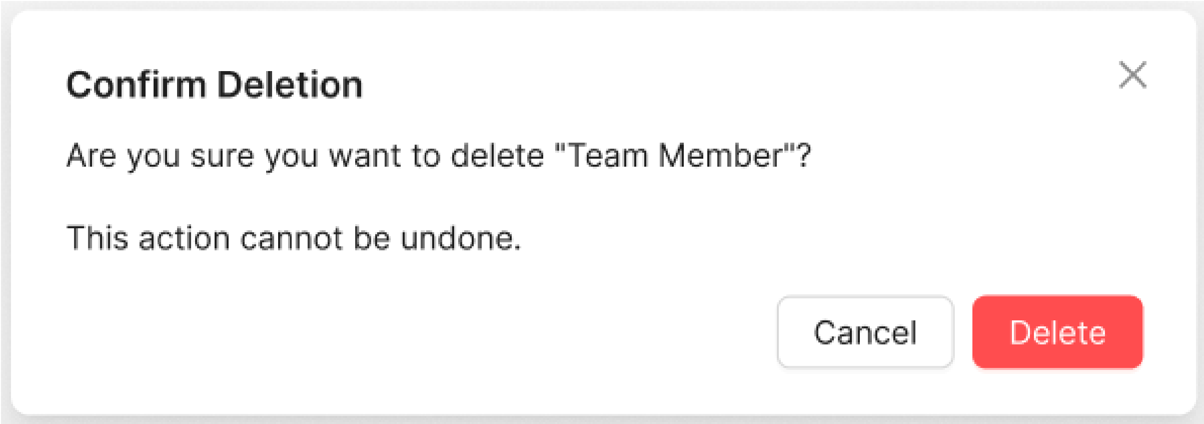
- Go to the Invite Member screens for invite members by clicking Invite Member.
- Delete Member: Delete member by clicking Delete



Field Description

Field Name	Field Type	Description
Invite Member	Button	Click to open Invite Member Form
Delete	Button	Click to remove a member from the organization.

This screen allows users to confirm before deleting a member from an organization.

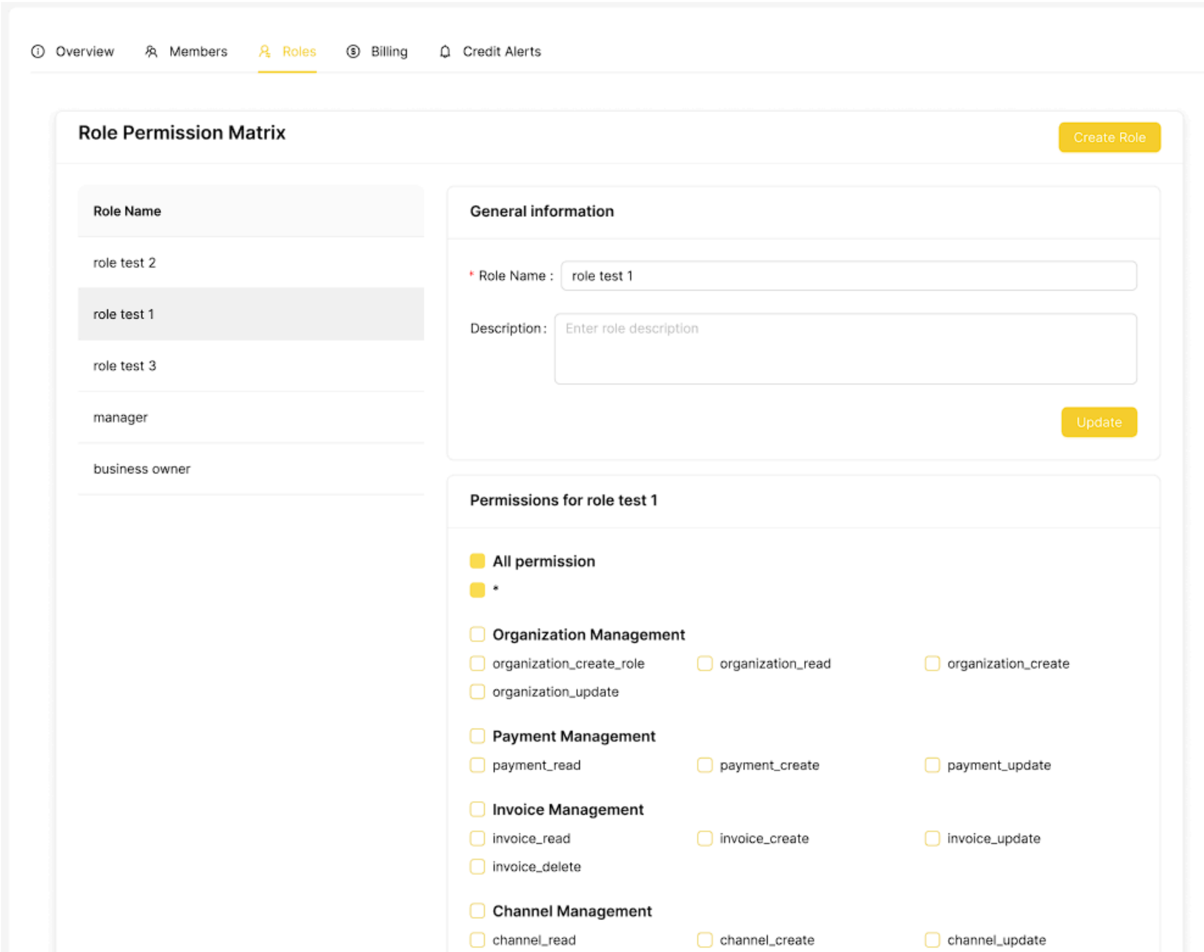


Field Description

Field Name	Field Type	Description
Cancel	Button	Click this button to cancel the current action and return to the previous screen without making any changes.
Delete	Button	Click this button to confirm and remove the selected member from the organization.

3.3.1.3 Role List

This screen allows users to view role list of an organization

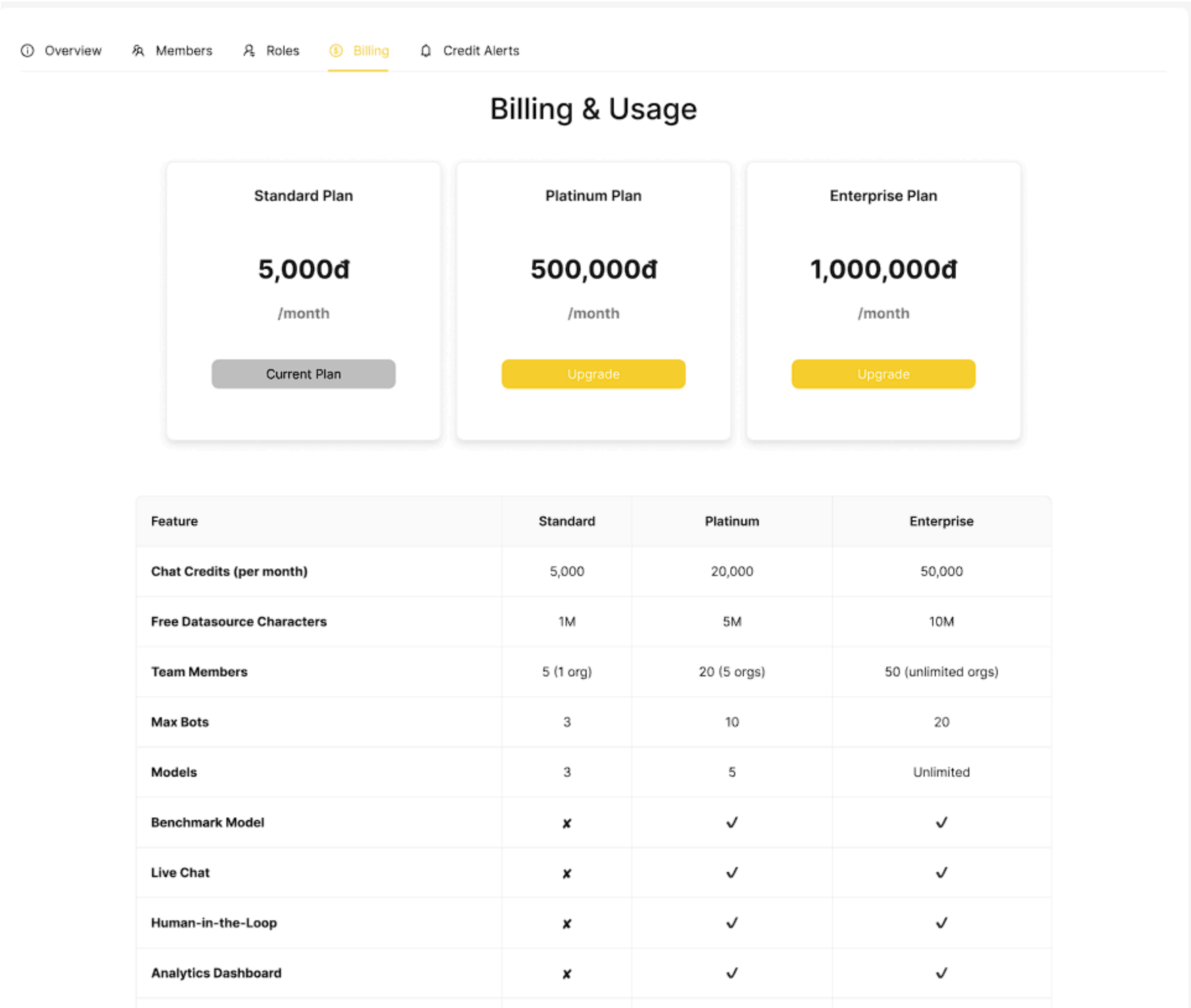


Field Description

Field Name	Field Type	Description
Create Role	Button	Click to open New Role Form

3.3.1.4 Plan

This screen allows users to view current plan and plan list of the organization

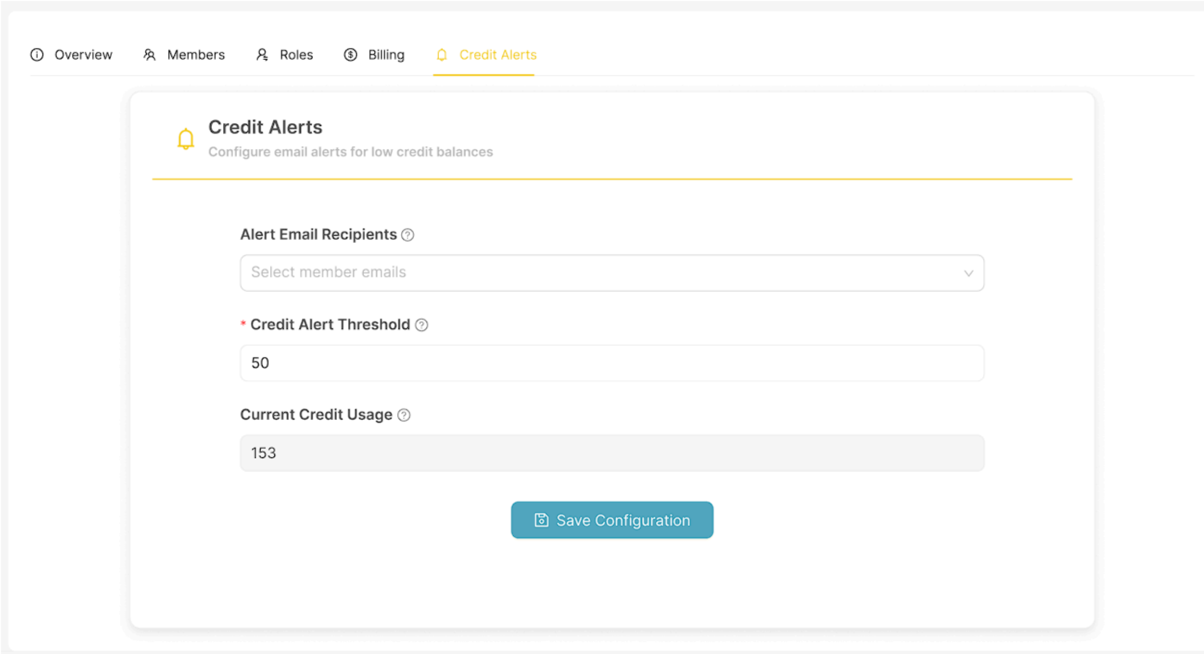


Field Description

Field Name	Field Type	Description
Upgrade	Button	Click to upgrade plan

3.3.1.5 Credit Alerts

This screen allows users to credit of the organization

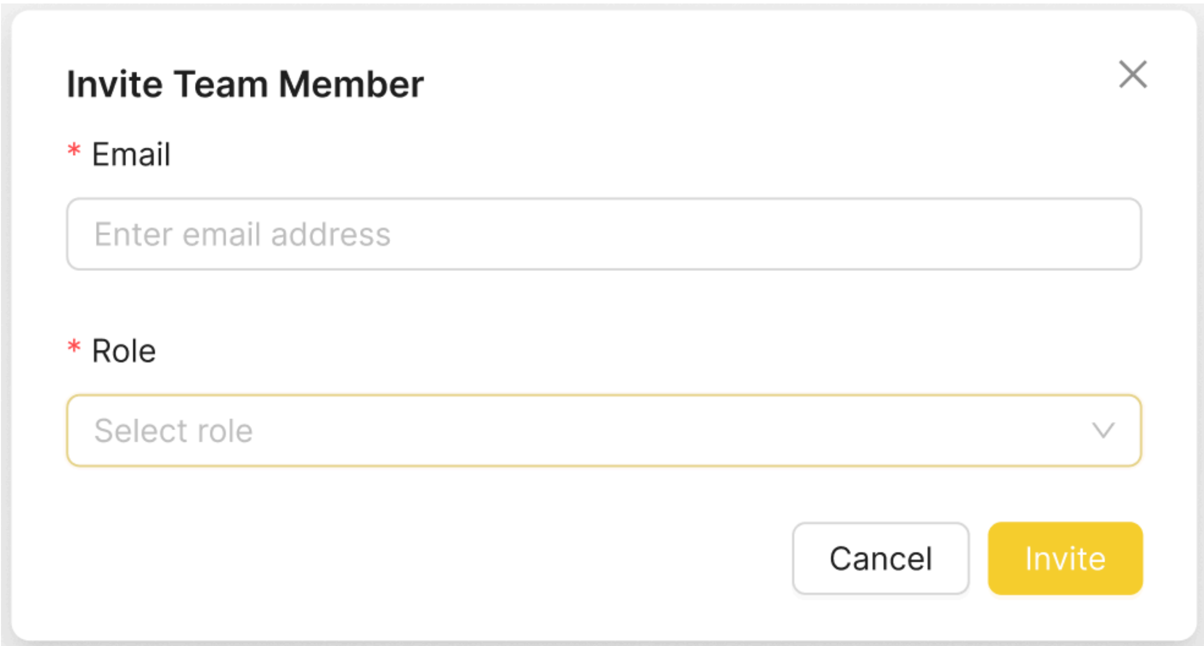


Field Description

Field Name	Field Type	Description
Alert Email Recipients	Text Box	Email addresses to receive credit alert notifications
Credit Alert Threshold	Text Box	Specify the credit balance amount that, when reached or fallen below, will trigger an alert.
Save Configuration	Button	Click to save the current alert settings and apply the configuration.

3.3.3 New Member

This screen allows users to add a new member

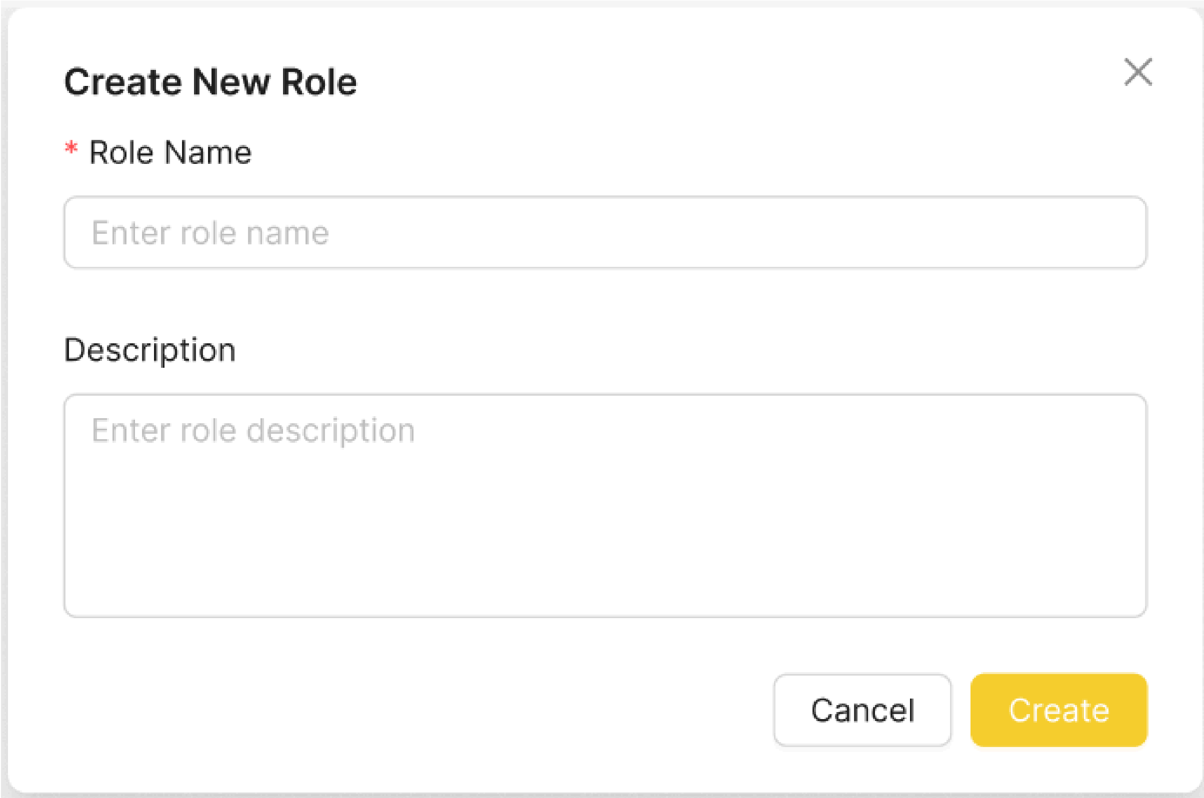


Field Description

Field Name	Field Type	Description
Email	Text Box	Data type: string. Required. Email address of the member you wish to invite to the organization.
Role	Selection Box	Required. Initial data values: all roles of organization. Select the role you want to assign to the new member within the organization.
Invite	Button	Click this button to send an invitation to the specified email address.
Cancel	Button	Click this button to cancel the invitation process and close the form without making changes.

3.3.4 New Role

This screen allows users to add a new role

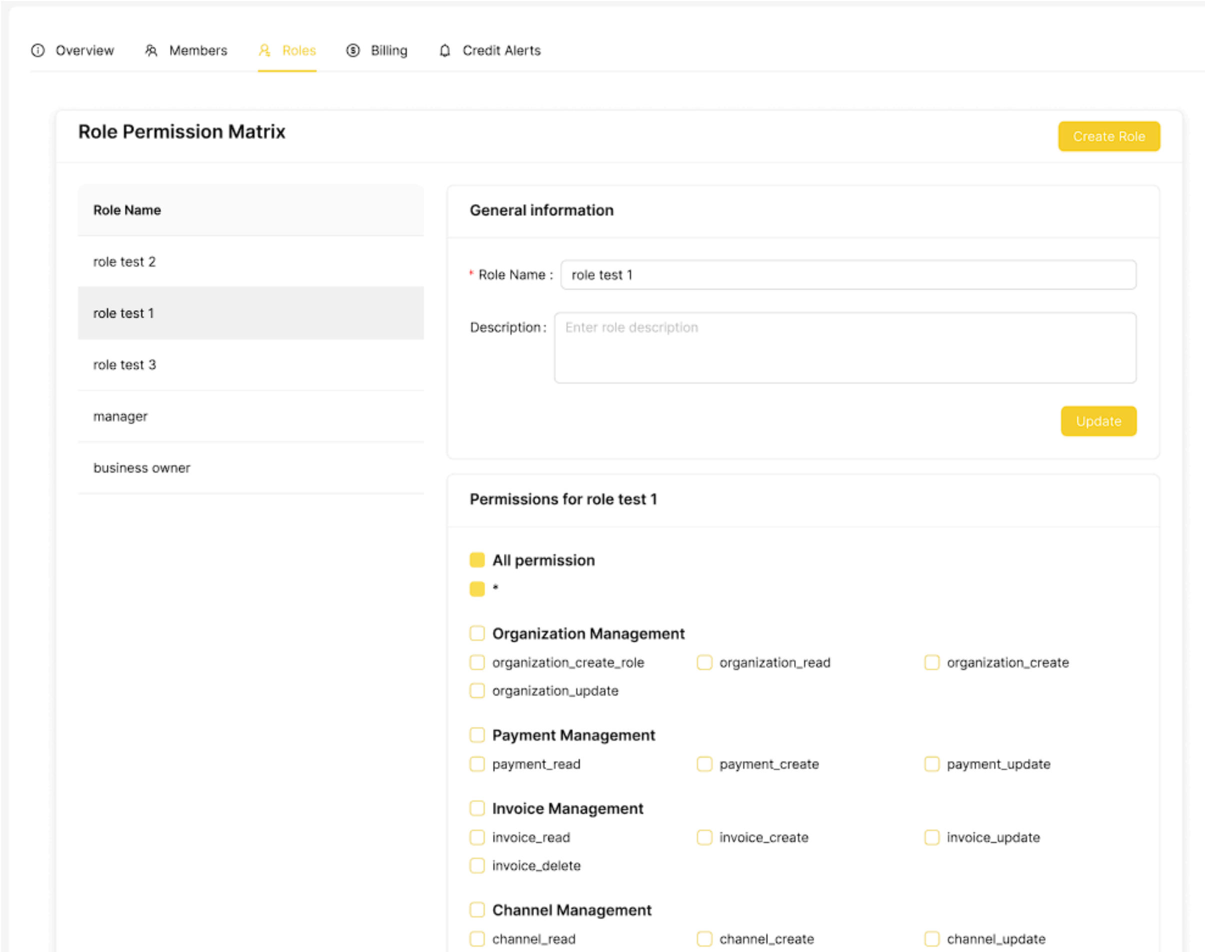


Field Description

Field Name	Field Type	Description
Role Name	Text Box	Data type: string. Required. Name of the role you want to create.
Description	Text Box	Data type: string. Provide a brief description of the role.
Create	Button	Click to submit New Role Form
Cancel	Button	Click this button to cancel the role creation process and close the form without saving

3.3.5 Role Details

This screen allows users to view role details



Field Description

Field Name	Field Type	Description
Role Name	Text Box	Data type: string. Required. Name of the role you want to create.
Description	Text Box	Data type: string. Provide a brief description of the role.
Permissions	Check Box	Choose the permissions that you want the role to have.
Update	Button	Click to save the role with the selected details and permissions.

3.4 Admin Staff Feature

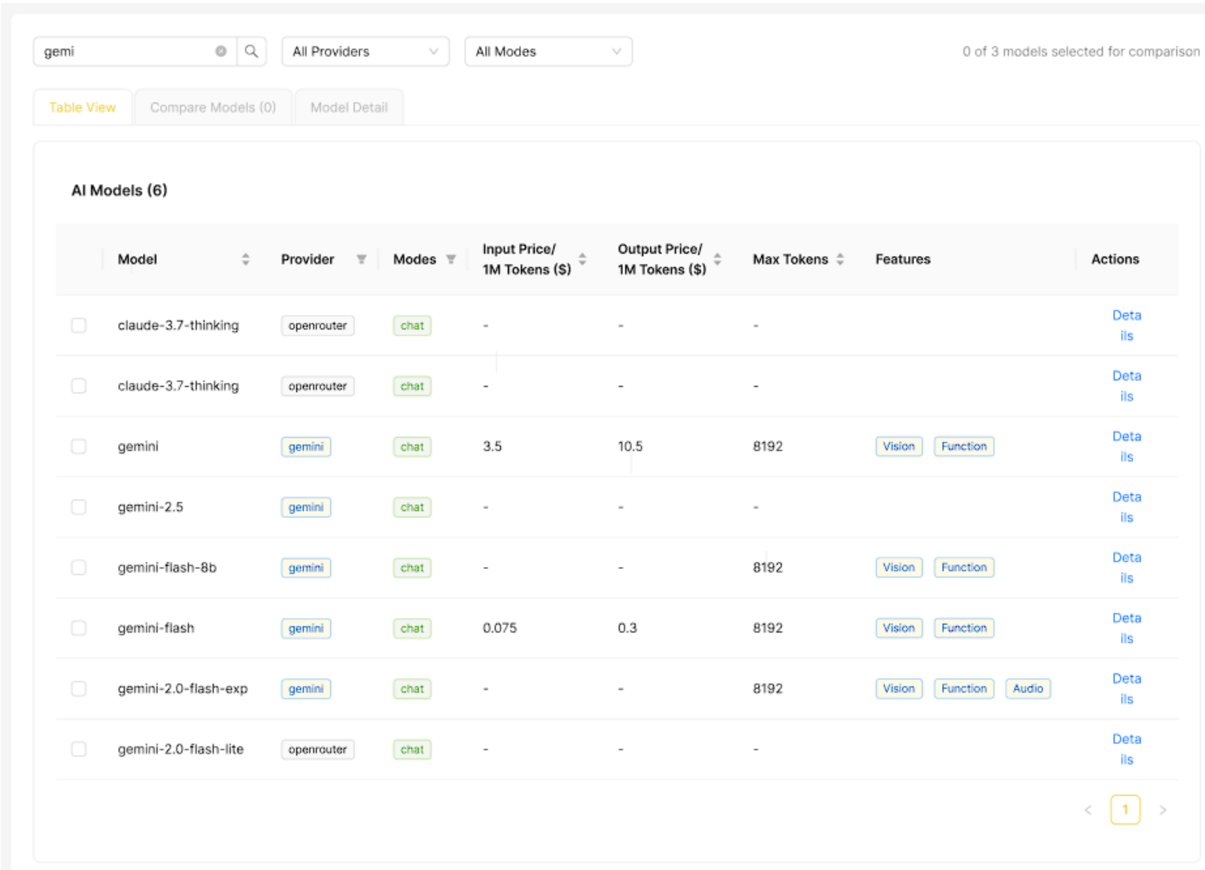
3.4.1 Model List

This screen allows the users to:

- View Model List: view list of model.
- Filter Model List: filter model by providers, modes
- Search Model: enter keyword(s) to search model by name

On the screen, s/he can also

- Go to the Model Details screens for view model details by clicking Details.

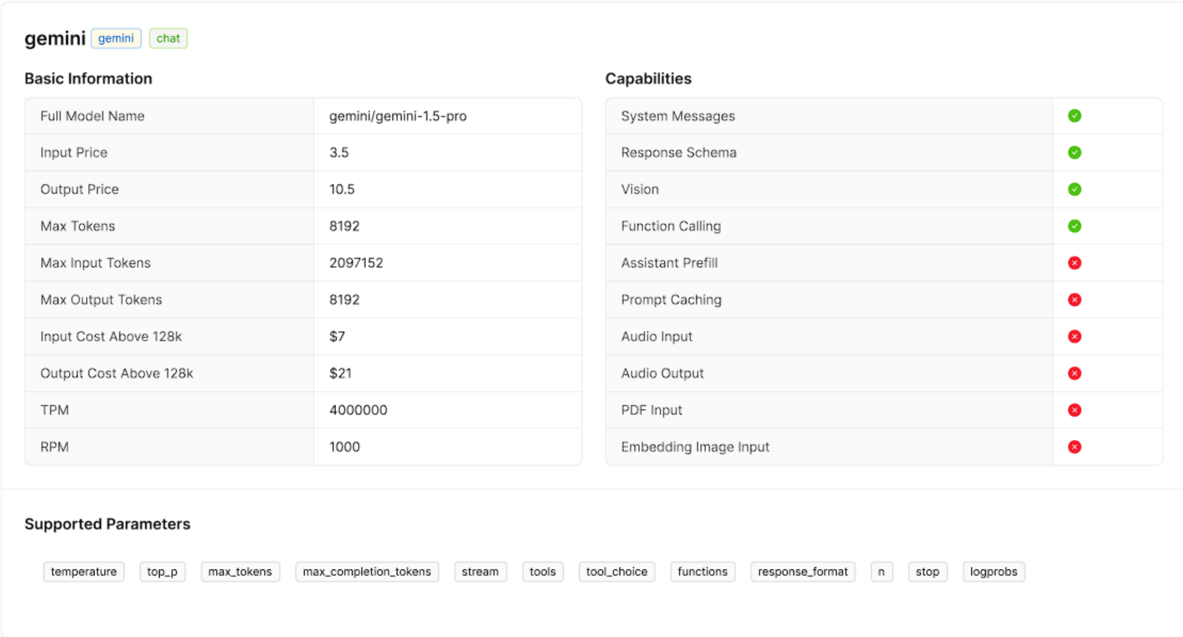


Field Description

Field Name	Field Type	Description
Search	Text Box	Data type: string. Allows users to search or filter models by name.
Provider	Selection Box	Choose provider to filter model
Mode	Selection Box	Choose mode to filter model
Details	Button	Click to open the Model Details Page.

3.4.2 Model Details

This screen allows users to view model details


gemini gemini chat

Basic Information

Full Model Name	gemini/gemini-1.5-pro
Input Price	3.5
Output Price	10.5
Max Tokens	8192
Max Input Tokens	2097152
Max Output Tokens	8192
Input Cost Above 128k	$7
Output Cost Above 128k	$21
TPM	4000000
RPM	1000

Capabilities

System Messages	✓
Response Schema	✓
Vision	✓
Function Calling	✓
Assistant Prefill	✗
Prompt Caching	✗
Audio Input	✗
Audio Output	✗
PDF Input	✗
Embedding Image Input	✗

Supported Parameters

temperature top_p max_tokens max_completion_tokens stream tools tool_choice functions response_format n stop logprobs

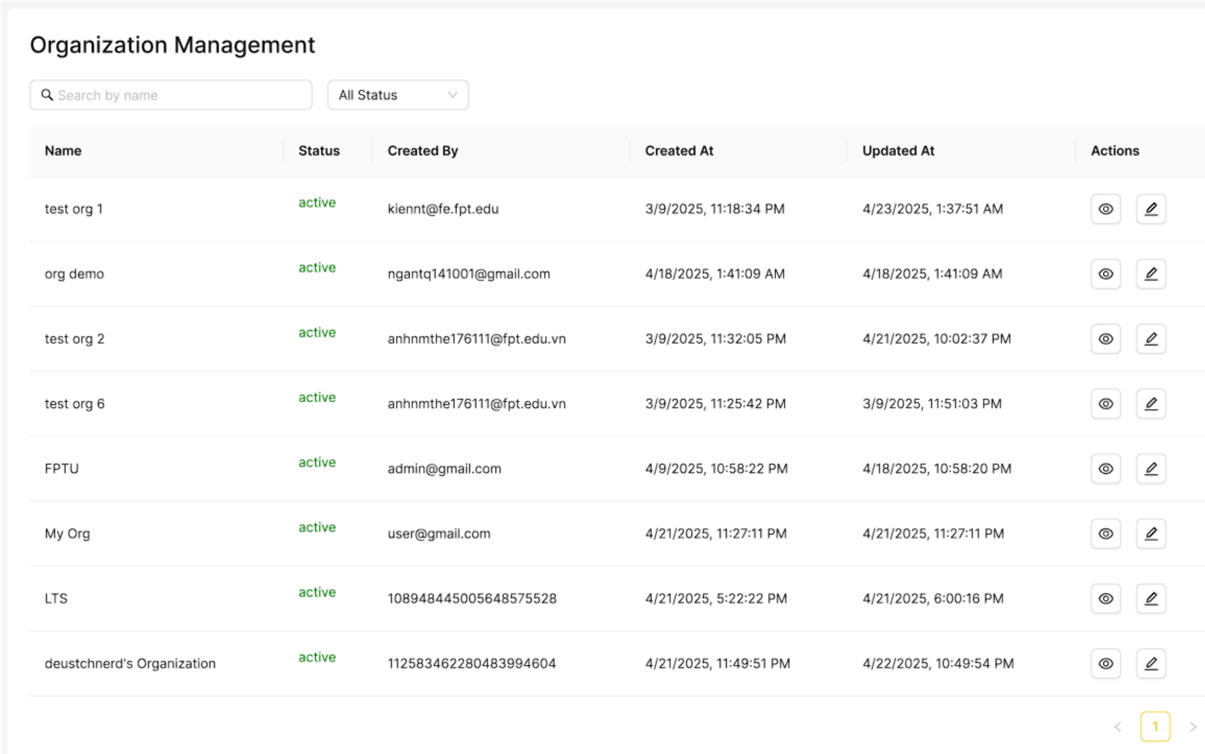
3.4.3 Organization List

This screen allows the users to:

- View Organization List: view list of organization .
- Filter Organization List: filter datasource by statuses
- Search Organization : enter keyword(s) to search organization by name

On the screen, s/he can also

- Go to the New Organization screens for adding new by clicking Add Organization.
- Go to the Organization Details screens for view datasource details by clicking Details.
- Go to the Organization Details screens for edit datasource details by clicking Edit.



Field Description

Field Name	Field Type	Description
Search	Text Box	Data type: string. Allows users to search organizations by name.
Status	Selection Box	Initial data values: all status of organization.Filter organizations based on their current status
Add	Button	Click to open the "New Organization" page
Details	Button	Click to view detailed information about the selected organization.
Edit	Button	Click to open Organization Details page

3.4.4 New Organization

This screen allows users to add new organization

Field Description

Field Name	Field Type	Description
Organization Name	Text Box	Data type: string. Required. Name of the organization you want to create.
Create	Button	Click to submit New Organization Form
Cancel	Button	Click this button to cancel the organization creation process and close the form without saving

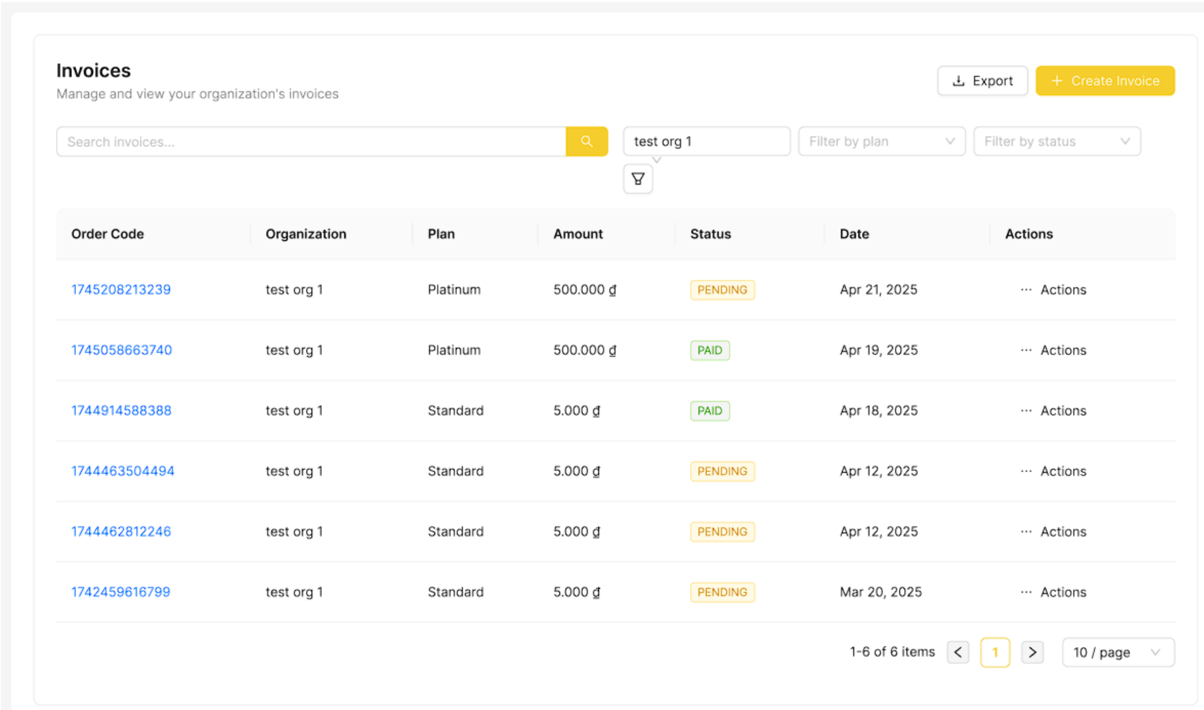
3.4.6 Invoice List

This screen allows the users to:

- View Invoice List: view list of invoice.
- Filter Invoice List: filter invoice by statuses, organization, plan
- Search Invoice: enter keyword(s) to search invoice by organization name, plan name

On the screen, s/he can also

- Go to the New Invoice screens for adding new by clicking Create Invoice.
- Go to the Invoice Details screens for view invoice details by clicking Details.
- Mark as paid: change invoice status to Paid



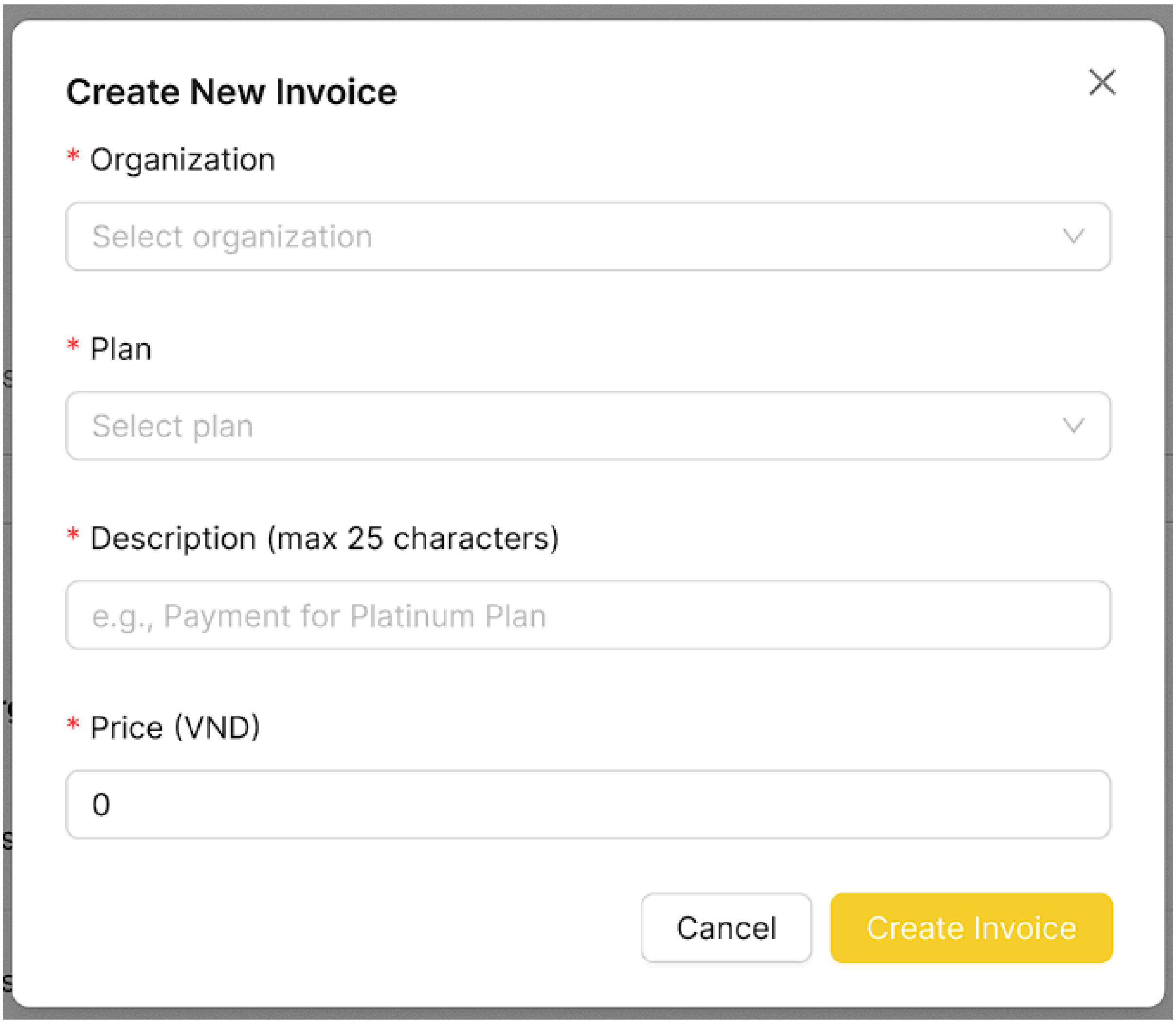
Field Description

Field Name	Field Type	Description
Search	Text Box	Data type: string. Allows users to search invoices by organization name, plan name
Organization	Selection Box	Initial data values: all organizations have invoice. Filter invoice based on their organization
Plan	Selection Box	Initial data values: all plans. Filter invoice based on their plan
Status	Selection Box	Initial data values: all status of invoice.Filter invoice based on their current status

Create Invoice	Button	Click to open the "New Invoice" page
Details	Button	Click to view detailed information of the invoice
Mark as paid	Button	Click to change status of invoice to Paid

3.4.7 New Invoice

This screen allows users to view user list



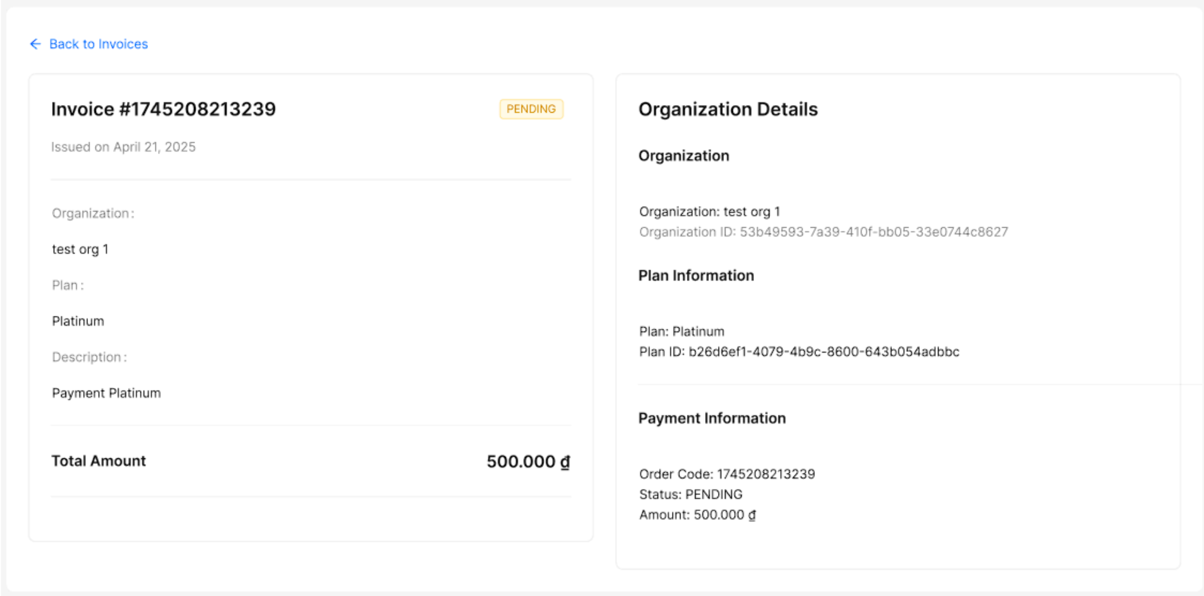
Field Description

Field Name	Field Type	Description
Organization	Selection Box	Select the organization for which the invoice is being created. Required
Plan	Selection Box	Choose the billing plan applicable to the invoice. Required
Description	Text Box	Data type: string. Details related to invoice
Price	Button	Data type: Number. Required. Total amount for the invoice.
Create Invoice	Button	Click to submit New Invoice Form

Cancel	Button	Click this button to cancel the invoice creation process and close the form without saving

3.4.8 Invoice Details

This screen allows users to view details of an invoice



3.5 Admin Feature

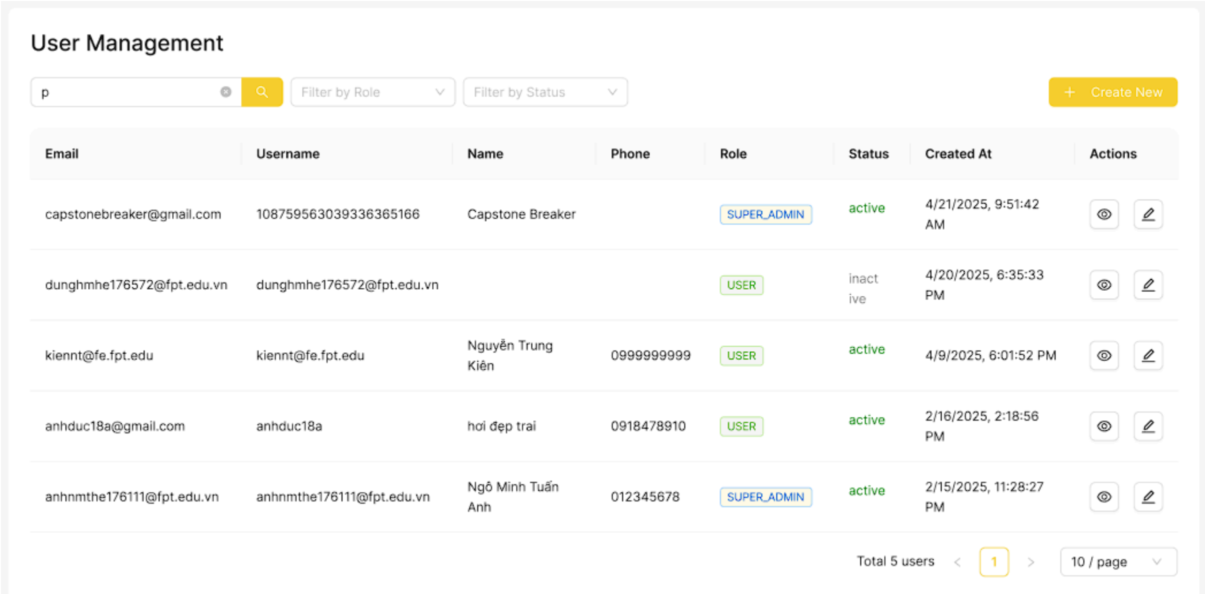
3.5.1 User List

This screen allows the users to:

- View User List: view list of users.
- Filter User List: filter users by statuses, roles
- Search User: enter keyword(s) to search user by email, username, name, phone

On the screen, s/he can also

- Go to the New User screens for adding new by clicking Create New.
- Go to the User Details screens to view user details by clicking Details.
- Go to the User Details screens to edit user details by clicking Edit.

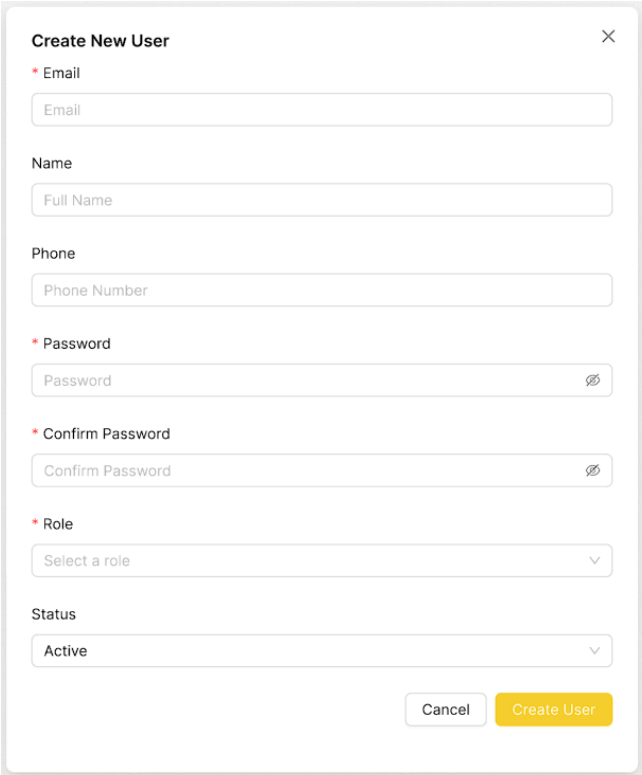


Field Description

Field Name	Field Type	Description
Search	Text Box	Data type: string. Allows users to search user by email, username, name
Role	Selection Box	Initial data values: all roles. Filter user based on their role
Status	Selection Box	Initial data values: all status of user. Filter users based on their current status
Create New	Button	Click to open the "New User" page
Edit	Button	Click to open the User Details Page
Details	Button	Click to view detailed information of the user

3.5.2 New User

This screen allows users to add new user

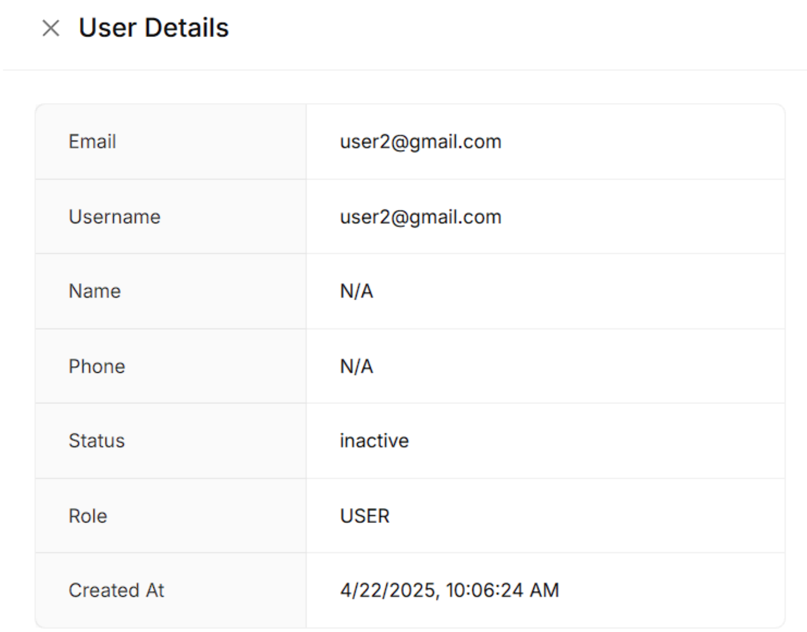


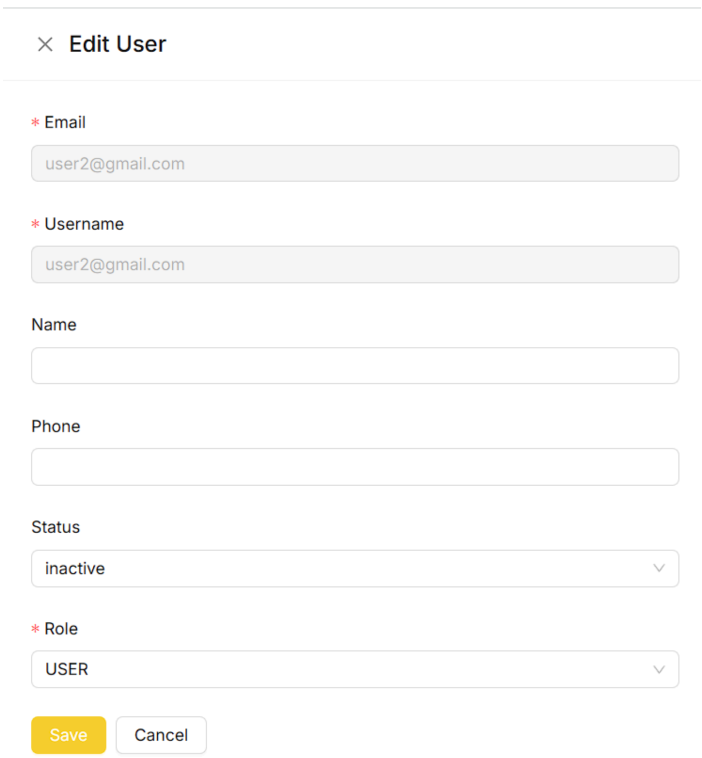
Field Description

Field Name	Field Type	Description
Email	Text Box	Data type: string. Required. User's email address
Name	Text Box	Data type: string. User's full name.
Phone	Text Box	Data type: number. User's phone number.
Password	Password Box	Data type: string. Required. Secure password for the user.
Confirm Password	Password Box	Data type: string. Required. Re-enter the password to confirm.
Role	Selection Box	Required. Select the user's role within the system.
Status	Selection Box	Set the user's account status (e.g., Active, Inactive).
Create User	Button	Click to submit the form and create the user with the provided details.
Cancel	Button	Click to abort the user creation process and discard any entered information.

3.5.3 User Details

This screen allows users to view user details





Field Description

Field Name	Field Type	Description
Name	Text Box	Data type: string. User's full name.
Phone	Text Box	Data type: number. User's phone number.
Role	Selection Box	Required. Select the user's role within the system.

Status	Selection Box	Set the user's account status (e.g., Active, Inactive).
Save	Button	Click to submit the form and update the user with the updated details.
Cancel	Button	Click to abort the user update process and discard any entered information.

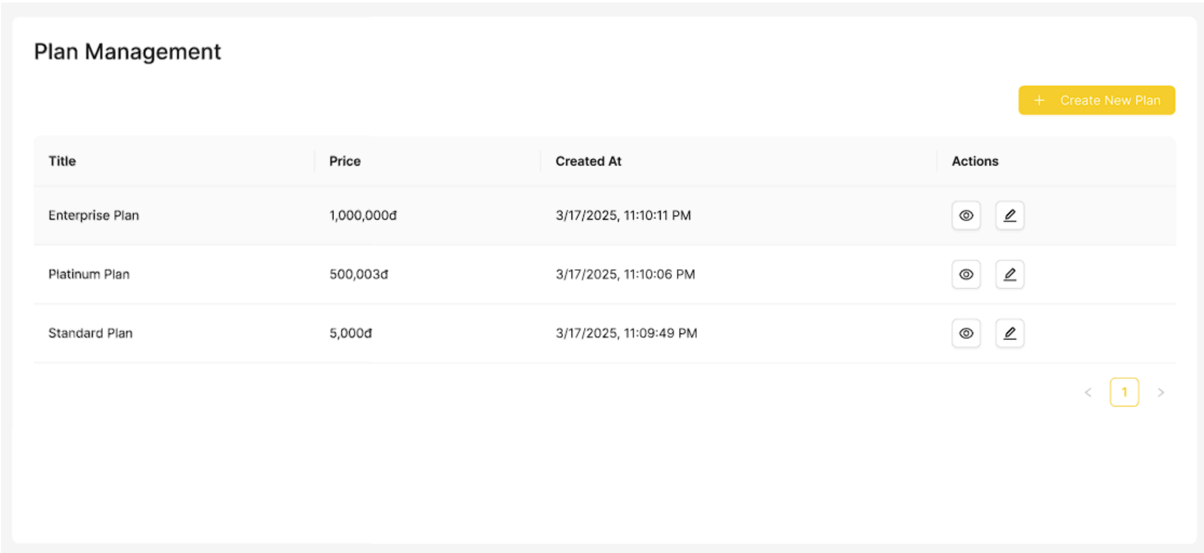
3.5.4 Plan List

This screen allows the users to:

- View Plan List: view list of plan.

On the screen, s/he can also

- Go to the New Plan screens for adding new by clicking Create New Plan.
- Go to the Plan Details screens to view plan details by clicking Details.
- Go to the Plan Details screens to edit plan details by clicking Edit.

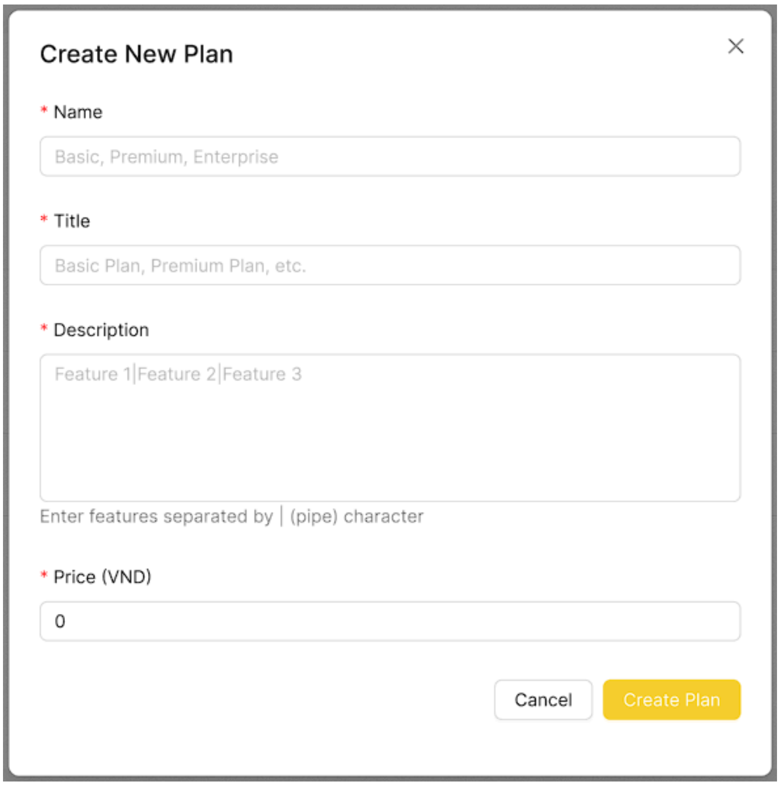


Field Description

Field Name	Field Type	Description
Create New	Button	Click to open the "New Plan" page
Edit	Button	Click to open the Plan Details Page
Details	Button	Click to view detailed information of the plan

3.5.5 New Plan

This screen allows users to add new plan

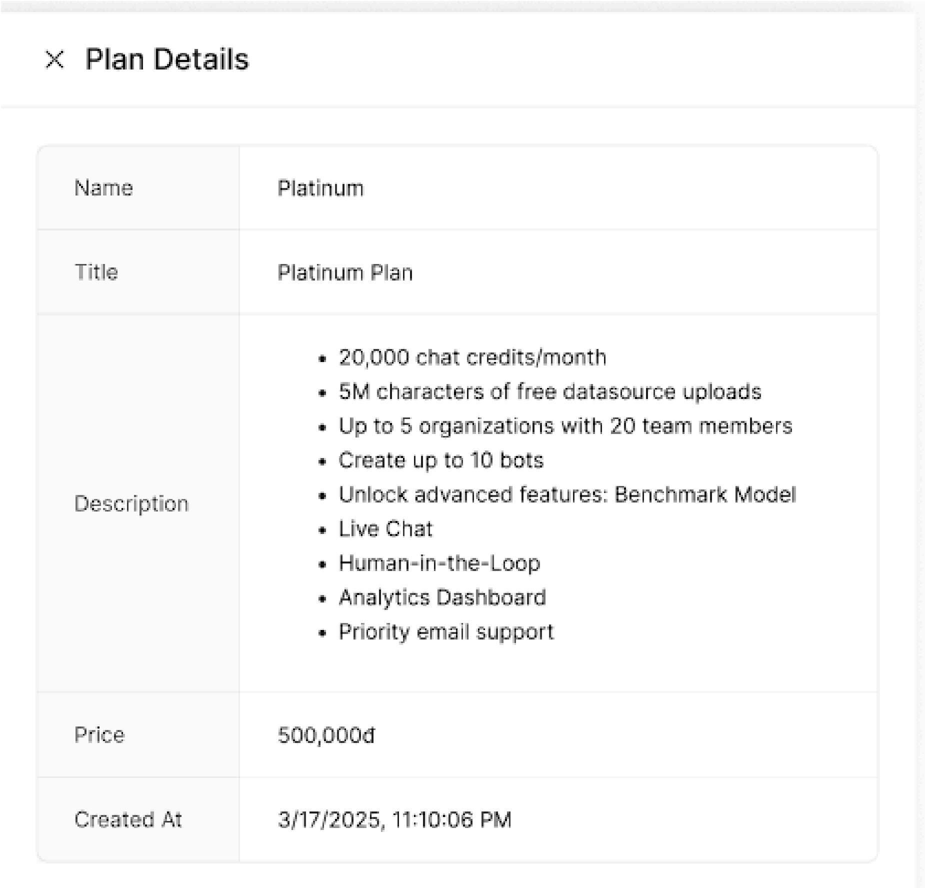


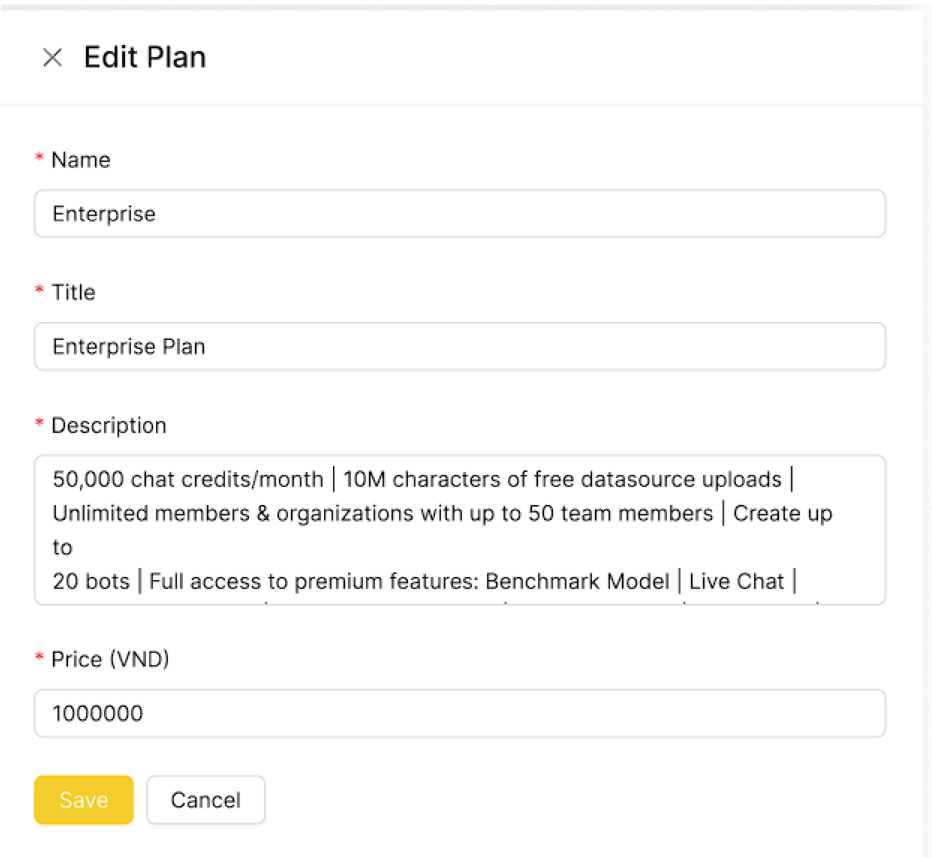
Field Description

Field Name	Field Type	Description
Name	Text Box	Data type: string. Required. Name of the plan
Title	Text Box	Data type: string. Required. Title of the plan
Description	Password Box	Data type: string. Required. Provide a brief description of the plan
Price	Password Box	Data type: number. Required. Price of the plan
Create Plan	Button	Click to submit the form and create the plan with the provided details.
Cancel	Button	Click to abort the plan creation process and discard any entered information.

3.5.6 Plan Details

This screen allows users to view plan details


Plan Details
Name
Platinum
Title
Platinum Plan
Description
20,000 chat credits/month
5M characters of free datasource uploads
Up to 5 organizations with 20 team members
Create up to 10 bots
Unlock advanced features: Benchmark Model
Live Chat
Human-in-the-Loop
Analytics Dashboard
Priority email support
Price
500,000đ
Created At
3/17/2025, 11:10:06 PM


Edit Plan
* Name
Enterprise
* Title
Enterprise Plan
* Description
50,000 chat credits/month | 10M characters of free datasource uploads |
Unlimited members & organizations with up to 50 team members | Create up
to
20 bots | Full access to premium features: Benchmark Model | Live Chat |
* Price (VND)
1000000
Save
Cancel

Field Description

Field Name	Field Type	Description
Name	Text Box	Data type: string. Required. Name of the plan
Title	Text Box	Data type: string. Required. Title of the plan
Description	Password Box	Data type: string. Required. Provide a brief description of the plan
Price	Password Box	Data type: number. Required. Price of the plan
Create Plan	Button	Click to submit the form and update the plan with the updated details.
Cancel	Button	Click to abort the plan updation process and discard any entered information.

4. Non-Functional Requirements

4.1 External Interfaces

ID	Requirements
UI-01	The font size is easy to distinguish where is the title, where is the content.
UI-02	The color of the background and text are eye-catching and easy to read.
UI-03	A confirmation dialog is required for delete or status change actions.
UI-04	Error messages must be shown in case user input in forms does not pass validation.
UI-05	Responsive design ensures dashboard accessibility on various devices.
UI-06	The CAIP Page Structure is easy to add, read, edit, and delete when adding product information fields on their website.

4.2 Quality Attributes

4.2.1 Usability

ID	Role	Requirements
QAT-01	Business owner	Business owners must be able to update knowledge base entries or FAQ data in a few click
QAT-02	Customer	Users can effortlessly communicate with chatbot provided by business owners.
QAT-03	Customer	The system must display valid information by user request given the knowledge base provided by the business owner

4.2.2 Performance

ID	Role	Requirements
QAT-04	Business owner, Customer, Admin	System should handle high concurrent socket and API traffic without significant performance degradation.
QAT-05	Business owner, Customer, Admin	Average API response time should be below 300ms under standard load conditions.
QAT-06	Business owner	Webhook data uploads must be processed and acknowledged within 1 minute.
QAT-07	Admin	System must handle concurrent user connections without significant latency increases.

4.2.3 Scalability

ID	Role	Requirements
QAT-08	Admin	System must support horizontal scaling to handle increased user traffic and data volume.
QAT-09	Admin	Major services (message handling, file processing, webhook handling) must operate independently to enable flexible scaling.
QAT-10	Admin	Queue mechanisms must be implemented to decouple producer and consumer services for efficient workload processing.

4.2.4 Fault Tolerance

ID	Role	Requirements
QAT-11	Admin	The system must remain operational even if individual service instances fail.
QAT-12	Admin	Failed LLM must automatically retry up to a configured maximum number of attempts.

4.2.5 Availability & Reliability

ID	Role	Requirements
QAT-13	Admin	Health checks must automatically restart unhealthy instances without manual intervention.

QAT-14	Admin	Load balancers must evenly distribute incoming traffic to available service instances.

4.2.6 Security

ID	Role	Requirements
QAT-15	Business owner, Customer, Admin	All API access must be protected using JWT authentication.
QAT-16	Business owner, Customer,	Implement rate limiting and input validation to prevent prompt injection or model abuse.

4.2.7 Observability & Monitoring

ID	Role	Requirements
QAT-17	Admin	Metrics like CPU, Memory, TPS, and Latency must be collected and visualized via Grafana.
QAT-18	Admin	Centralized logging must be implemented with correlation IDs (traceId, userId) for effective troubleshooting.

5. Requirement Appendix

5.1 Business Rules

ID	Rule Definition
BR-01	User’s password must be encoded
BR-02	Role permissions must be defined by the Business Owner before assigning to members.
BR-03	Users must verify their email before accessing organization creation features.
BR-04	Only Business Owners can create and manage organizations.
BR-05	A default organization is automatically created for a user upon email verification.
BR-06	Users must select and pay for a subscription plan to activate organization features.
BR-07	Payments must be processed through a third-party payment provider.
BR-08	Business Owners can invite members to the organization only after purchasing a plan.
BR-09	Invited members must accept the email invitation to join the organization.
BR-10	Only Business Staff can create, configure, and manage bots.
BR-11	Bots must be linked to a verified data source before activation.
BR-12	Bot responses must adhere to the configured persona and tone settings.

BR-14	Chat history must be logged and accessible to Business Staff for review.
BR-15	Only Admins can manage user accounts, plans, and invoices across the system.
BR-16	Subscription plans must be active to grant access to corresponding features.
BR-17	Credit alerts must be sent to the configured email when usage thresholds are reached.
BR-18	Bots can only be integrated with supported communication platforms (e.g., Zalo, Telegram, Slack).
BR-19	Only Business Owners can view and upgrade organization plans.
BR-20	The chatbot must not ask for or process sensitive information such as passwords or credit card numbers.
BR-21	The system must support sorting and filtering by bot name, status

5.2 System Messages

#	Message Code	Message Type	Context	Content
1	MSG01	In line	No search results returned	No search results.
2	MSG02	In red, under the text box	Required input fields are empty	The * field is required.
3	MSG03	Toast message	Successful update of bot information	Bot updated successfully.
4	MSG04	Toast message	Successful creation of a new bot	Bot created successfully.
5	MSG05	Toast message	Email confirmation sent for account activation	A confirmation email has been sent to {email_address}.
6	MSG06	Toast message	Successful deletion of a bot	Bot deleted successfully.
7	MSG07	Toast message	Successful creation of a new organization	Organization created successfully.
8	MSG08	In red, under the text box	Input exceeds maximum length	Exceed max length of {max_length}.

9	MSG09	In line	Incorrect username or password during login	Incorrect username or password. Please check again.
10	MSG10	Toast message	Successful user registration	Register Success.
11	MSG11	Toast message	No bot found for the provided ID	Bot not found.
12	MSG12	Toast message	No model found for the provided ID	Model not found.
13	MSG13	Toast message	Bot name is required	Name is required.
14	MSG14	Toast message	Bot with the specified name already exists	Bot with this name already exist.
15	MSG15	Toast message	Successful creation of a new channel	Channel created successfully.
16	MSG16	Toast message	Successful update of a channel	Channel updated successfully.
17	MSG17	Toast message	No chat found for the provided ID	Chat not found.
18	MSG18	Toast message	Successful creation of a new chat	Chat created successfully.
19	MSG19	Toast message	Successful deletion of a chat	Chat deleted successfully.
20	MSG20	Toast message	Successful retrieval of data content	Get list data content success.
21	MSG21	Toast message	Successful creation of data content	Create Success!
22	MSG22	Toast message	Successful update of data content	Update Success!

23	MSG23	Toast message	Successful deletion of data content	Delete Success!
24	MSG24	Toast message	Unauthorized access attempt	Unauthenticated.
25	MSG25	Toast message	Successful update of an invoice	Invoice updated successfully.
26	MSG26	Toast message	Successful deletion of an invoice	Invoice deleted successfully.
27	MSG27	Toast message	Successful retrieval of live chats	Live chats retrieved successfully.
28	MSG28	Toast message	Successful login	Login success.
29	MSG29	Toast message	Organization ID is required	Organization ID is required.
30	MSG30	Toast message	Successful sending of an invitation	Invitation sent successfully.
31	MSG31	Toast message	Successful update of member role	Member role updated successfully.
32	MSG32	Toast message	Successful removal of a member	Member removed successfully.
33	MSG33	Toast message	Successful retrieval of organizations	Organizations and users retrieved successfully.
34	MSG34	Toast message	Successful update of an organization	Organization updated successfully.
35	MSG35	Toast message	Successful deletion of an organization	Organization deleted successfully.
36	MSG36	Toast message	Successful creation of a new role	New role created successfully!

37	MSG37	Toast message	Successful update of role permissions	Role permissions updated successfully.
38	MSG38	Toast message	Successful payment confirmation	Payment confirmed.
39	MSG39	Toast message	Successful cancellation of a payment link	Payment link cancelled.
40	MSG40	Toast message	Successful update of user profile	Profile updated successfully.
41	MSG41	Toast message	Successful password update	Password updated successfully.
42	MSG42	Toast message	Successful email update	Email updated successfully.
43	MSG43	Toast message	Successful account activation	Account activated successfully. You can now log in.
44	MSG44	Toast message	Password reset link sent	Password reset link sent successfully. Please check your email.
45	MSG45	Toast message	Successful password reset	Password reset successfully. You can now log in.
46	MSG46	Toast message	Successful Slack integration	Integration Slack successfully!
47	MSG47	Toast message	Successful deletion of Slack integration	Delete Integration Slack successfully!
48	MSG48	Toast message	Successful Telegram integration	Integration Telegram successfully!
49	MSG49	Toast message	Successful deletion of Telegram integration	Delete Integration Telegram successfully!

50	MSG50	Toast message	Successful message sent to Slack	Message sent to Slack successfully.
51	MSG51	Toast message	Successful message sent to Telegram	Message sent to Telegram successfully.
52	MSG52	Toast message	No user found for the provided ID	User not found.
53	MSG53	Toast message	Successful creation of a new user	User created successfully.
54	MSG54	Toast message	Successful update of a user	User updated successfully.
55	MSG55	Toast message	Successful deletion of a user	User deleted successfully.
56	MSG56	Toast message	No file uploaded for benchmark creation	No file uploaded!
57	MSG57	Toast message	Title is required for benchmark creation	Title is required!
58	MSG58	Toast message	At least one benchmark configuration is required	At least one benchmark configuration is required!
59	MSG59	Toast message	Invalid benchmark configuration at specified index	Invalid benchmark configuration at index {index}!
60	MSG60	Toast message	Invalid organization ID provided	Invalid organization ID.
61	MSG61	Toast message	No channel found for the specified bot	No channel found for this bot.
62	MSG62	Toast message	Hostname must be an array of strings	Hostname must be an array of strings.

63	MSG63	Toast message	Invalid bot model name provided	Invalid bot model name.
64	MSG64	Toast message	Failed to retrieve AI models	Failed to retrieve AI models.
65	MSG65	Toast message	AI model not found for the specified name	AI model {botModelName} not found.
66	MSG66	Toast message	Invalid credit per chat value for the model	Invalid creditPerChat value for model {botModelName}.
67	MSG67	Toast message	Insufficient credits for chat interaction	Insufficient credits for this chat.
68	MSG68	Toast message	Failed to save organization credits	Failed to save organization credits.
69	MSG69	Toast message	Failed to send credit alert email	Failed to send credit alert email.
70	MSG70	Toast message	File is required for data content creation	File is required.
71	MSG71	Toast message	URL is required for data content creation	URL is required.
72	MSG72	Toast message	Invalid URL provided	URL is invalid.
73	MSG73	Toast message	Data source not found	Data Source not found.
74	MSG74	Toast message	Data content not found	Data Content not found.
75	MSG75	Toast message	Account not found during login	Account not found.
76	MSG76	Toast message	Incorrect password provided during login	Wrong password!

77	MSG77	Toast message	Invalid permission IDs provided	Invalid permission IDs provided.
78	MSG78	Toast message	Role does not exist	Role does not exist.
79	MSG79	Toast message	No matching role permissions found to update	No matching role permissions found to update.
80	MSG80	Toast message	Role does not exist in the organization	Role does not exist in Organization.
81	MSG81	Toast message	Invalid member data provided	Invalid member.
82	MSG82	Toast message	Account already exists during registration	Account is already existed.
83	MSG83	Toast message	Account not activated during registration	Account is not already activated.
84	MSG84	Toast message	Invalid or expired activation code	Invalid or expired activation code.
85	MSG85	Toast message	Account does not exist or is inactive for password reset	Account with this email does not exist or is inactive.
86	MSG86	Toast message	Cannot reset password for non-system provider account	Cannot reset password for this account. Please log in using your original sign-in method.
87	MSG87	Toast message	Invalid or expired reset token	Invalid or expired reset token.
88	MSG88	Toast message	Slack Bot Token is required for integration	Slack Bot Token is required!
89	MSG89	Toast message	Slack Signing Secret is required for integration	Slack Signing Secret is required!

90	MSG90	Toast message	Slack App ID is required for integration	Slack App ID is required!
91	MSG91	Toast message	Missing required fields for Slack message	Missing required fields: botId, message, channelSlackId, or replyToMessageId.
92	MSG92	Toast message	Telegram API Token is required for integration	Telegram API Token is required!
93	MSG93	Toast message	Invalid Telegram API Token or failed to connect	Invalid Telegram API Token or failed to connect to Telegram API.
94	MSG94	Toast message	Missing required fields for Telegram message	Missing required fields: botId, message, channelTelegramId, or replyToMessageId.
95	MSG95	Toast message	Unable to identify the source of the request	Unable to identify the source of your request.
96	MSG96	Toast message	Domain mismatch for the bot	Domain mismatch: {requestHostname} is not allowed for this bot.

IV. Software Design Document

1. System Design

1.1 Overview of RAG Architecture

1.1.1 Basic RAG Architecture

The Retrieval-Augmented Generation (RAG) pipeline implemented in this research comprises three primary stages: indexing, retrieval, and generation. During the indexing stage, documents are segmented and encoded into vectors using embedding models, then stored in a vector database. The retrieval stage employs user queries to identify relevant document chunks by comparing vector similarities. Finally, the generation stage integrates retrieved document chunks into a large lan guage model (LLM) to produce contextually grounded responses, ensuring accuracy and relevance.

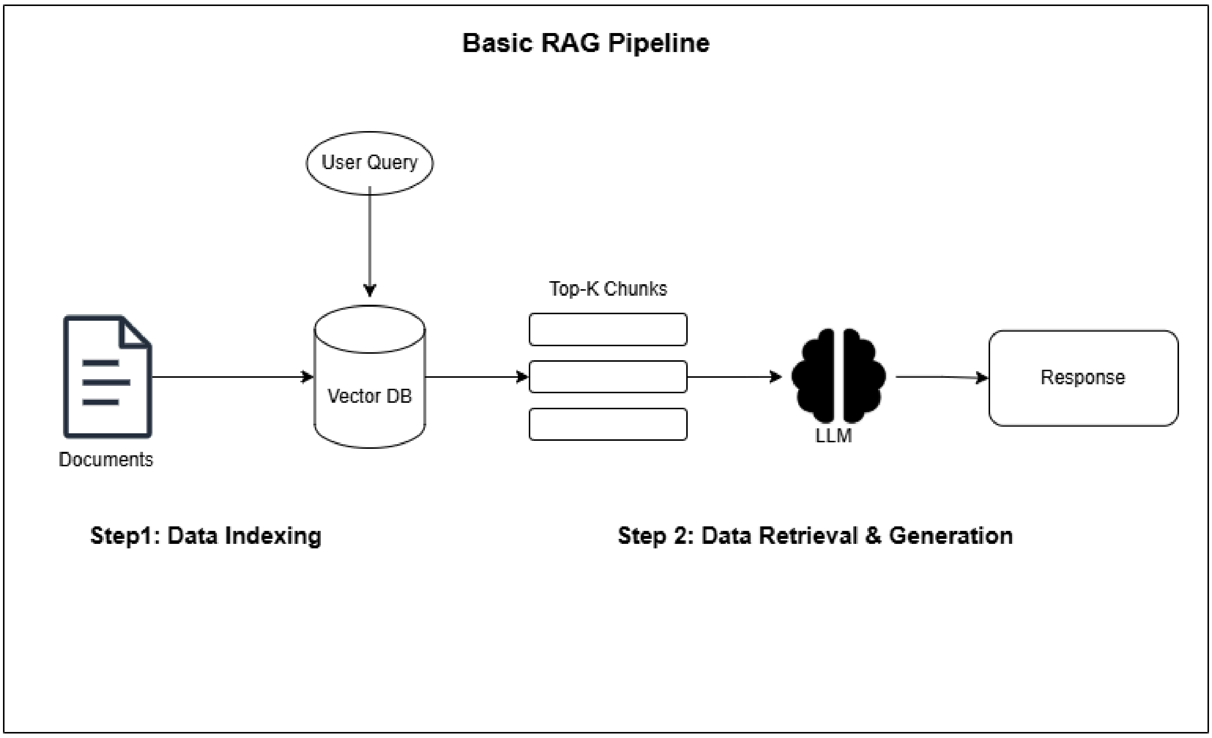


Figure 1.1.1.1 Basic RAG Architecture

1.1.2 Advanced RAG Architecture:

1.1.2.1 Document Indexing

Indexing is not merely about storing data but also involves organizing and optimizing data so that the necessary information can be easily retrieved without losing the important context of the data. Some techniques in the indexing process:

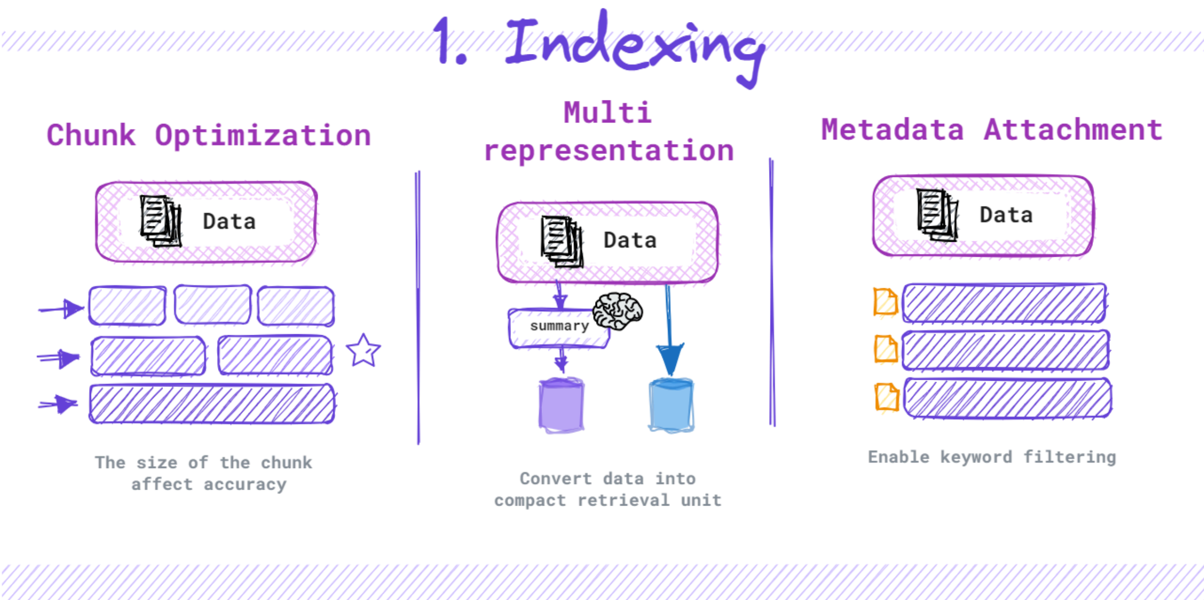


Figure 1.1.2.1.1 Document Indexing

- Chunk Optimization: Optimizes the size and structure of chunks to ensure that they are not too large or too small, helping to maintain the necessary context without exceeding the length limit of LLM.
- Metadata Attachment: Adding metadata to each chunk or piece of data increases the ability to analyze and classify information, allowing data to be retrieved in a more systematic way, suitable for a variety of specific situations.
- Multi-Representation: Multi-representation techniques are employed, generating multiple vector representations for each document chunk, including summary-based and content-based embeddings. Additionally, the Small-to-Big Retrieval approach is integrated, where smaller chunks are used for initial retrieval, subsequently referencing larger parent documents to provide comprehensive context.

1.1.2.2 Query Transformation and Routing

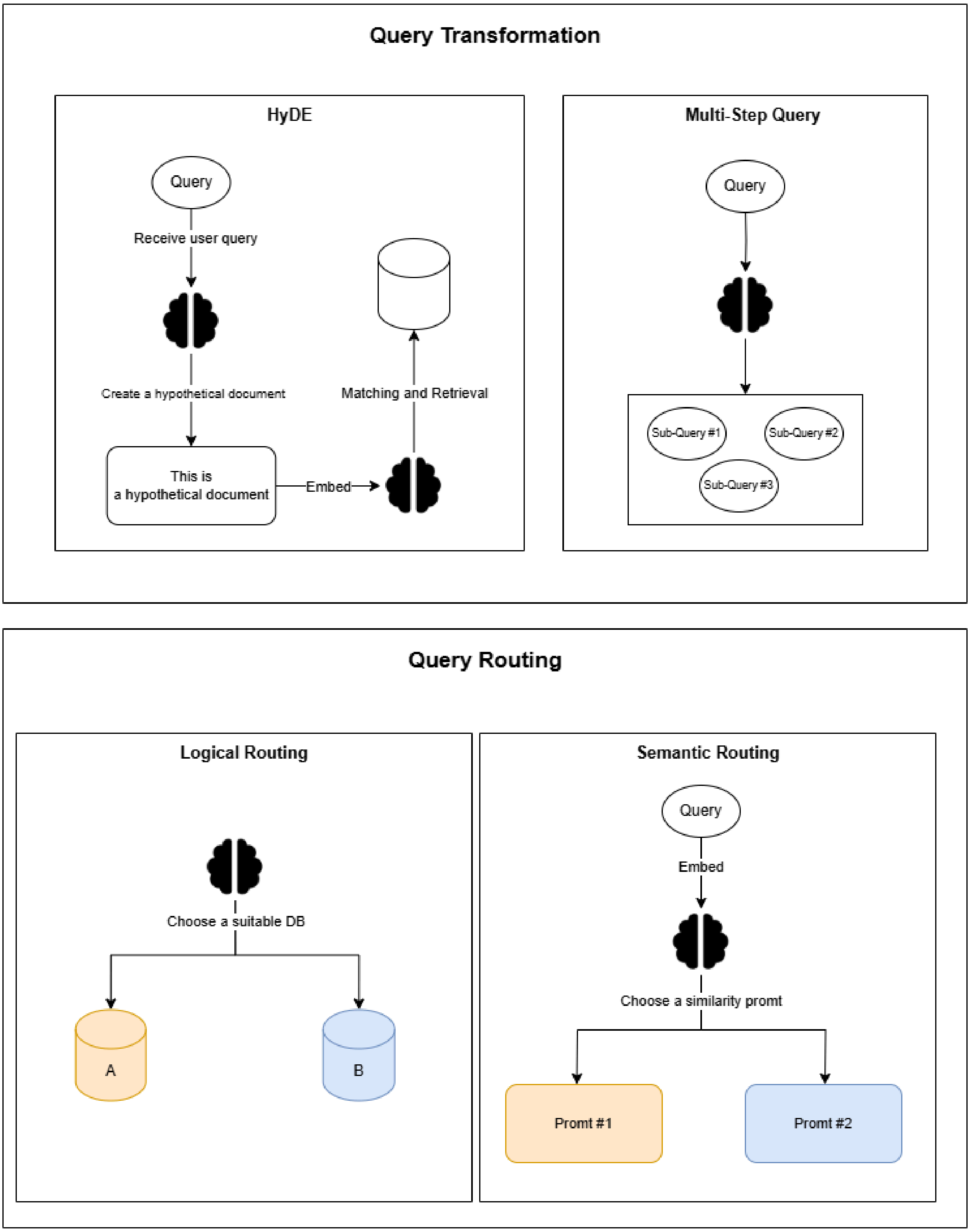


Figure 1.1.2.2.1 Query Transformation and Routing

a, Query Transformation is a technique that uses the model LLM as a reasoning tool to edit user input to improve the quality of information retrieval. LLM can convert original questions into clearer, more understandable questions, thereby increasing the ability to search and retrieve information effectively. Some techniques in the Query Transformation process:

- HyDE: is a reversal technique in which LLM is required to generate hypothetical data based on the question, and then use the vector of this data along with the vector of the question to improve the quality of reference retrieval. This technique enhances the semantic similarity between the question and the retrieved reference, thereby enhancing the accuracy and efficiency of the query process.
- Multi-Step Query: This is a method of decomposing a complex question into more simple sub-questions, performing parallel retrieval for each of these sub-questions, and combining the retrieval results together. This allows LLM for more accurate synthesis and responses. This method is especially useful in cases where there is no direct information about the content in question.

b, Query Routing is the step taken by LLM the model to determine the next action based on the user's query. Options can be to search for content in a specific dataset or to experiment with different search directions and then aggregate the results into a unified response. This process helps direct queries to the most relevant data source, from classic vector databases to graph or relational databases, or even various hierarchical indexes. Some techniques in the Query Routing process:

- Logical Routing: It is a technique that uses logic to direct queries to the right data source. By analyzing the structure and purpose of the question, the router selects the most appropriate index or data source to perform the query. This helps to optimize the process of retrieving information by ensuring that the query is processed by the data source that is capable of the most accurate response.
- Semantic Routing: This method leverages the semantics of the question for direction. The router analyzes the semantic meaning of the question and directs it to the appropriate index or data source, in order to increase the ability to search and retrieve relevant information accurately. This method is especially effective when dealing with complex queries, which require an understanding of the context and meaning of each word and phrase in the query.

1.1.2.3 Retrieval

The retrieval process is at the core of the RAG system, which focuses on retrieving reference data from various sources to provide the context and information needed for LLM the model to perform response sentence generation. Some techniques in the Retrieval process:

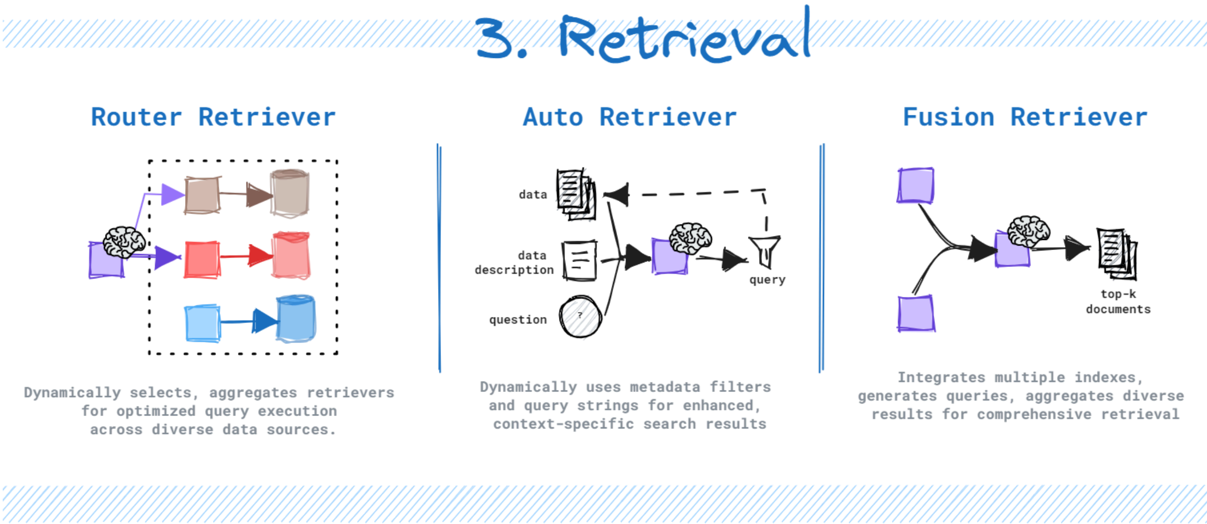


Figure 1.1.2.3.1 Retrieval

- Router Retriever: A technique that uses an LLM to dynamically decide the appropriate data source or query tool for each specific query.
- Auto Retriever: A method that automatically queries the database by leveraging an LLM to determine metadata for filtering or generating suitable queries for retrieval.
- Fusion Retriever: Combines results from multiple queries and indexes, optimizing information retrieval and ensuring comprehensive results without duplication, providing a multidimensional view of the retrieval process.

Post-Retrieval:

The Post-Retrieval process is where retrieved results are refined through filtering, reranking, or transformation. The goal of this process is to prepare and enhance the context before feeding it into the LLM to generate the final response, ensuring that the information provided to the LLM is as accurate and effective as possible.

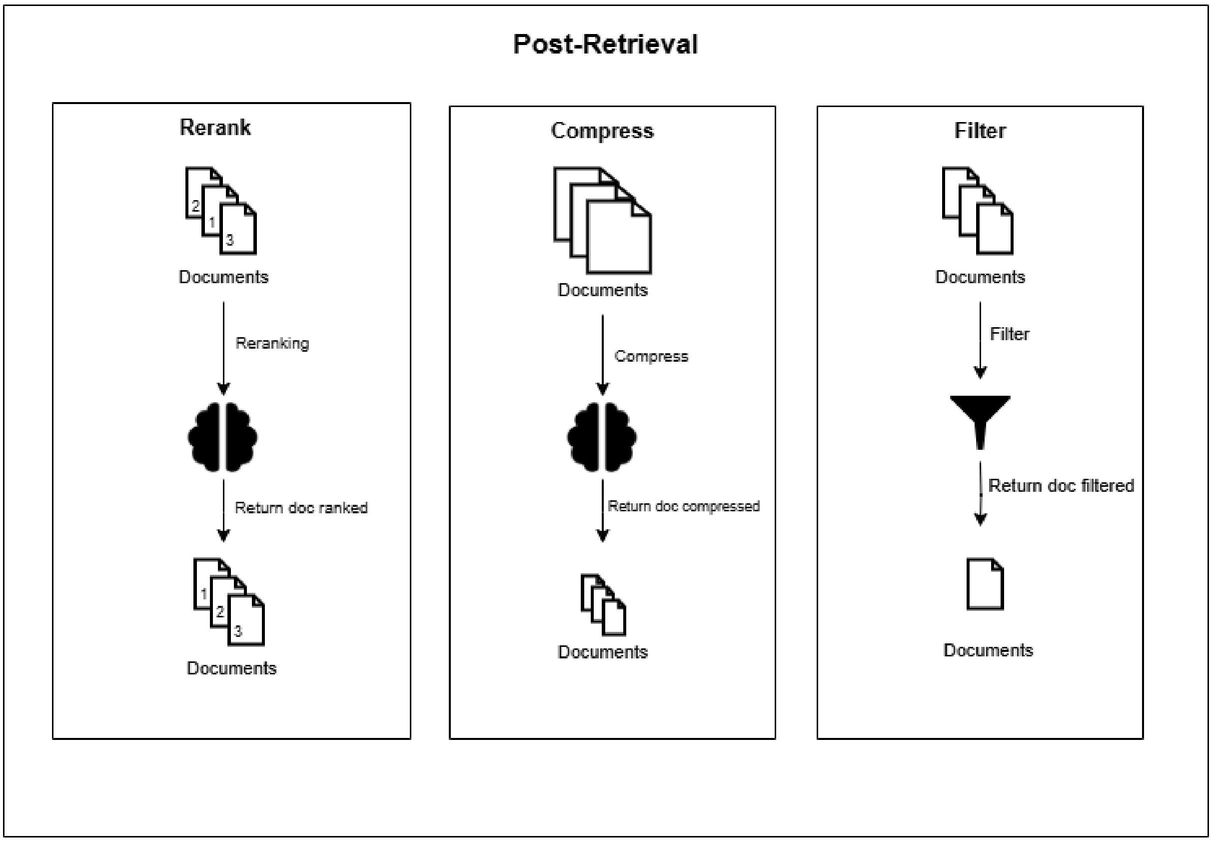


Figure 1.1.2.3.2 Post-Retrieval

a, Rerank: Reorders retrieved text passages so that the most relevant results appear first, improving information accuracy. Reranking not only reduces the number of documents that need to be fed into the LLM but also acts as a filter to process language more precisely.

b, Compress: Removes redundant and unnecessary context, eliminating noise to enhance the LLM's understanding of key information. Compression optimizes the context length that the LLM can handle, improving response quality by focusing on the most important details.

c, Filter: Selects relevant content before passing it to the LLM, discarding documents or information that are irrelevant or of low accuracy. This technique ensures that only high-quality and relevant information is used, enhancing the accuracy and reliability of responses.

1.2 System Architecture

a, Overview

This system is designed to provide a comprehensive platform for users to manage services and engage in intelligent interactions through multiple communication channels, primarily chat. It integrates robust internal services for business logic, data processing, and user interaction while leveraging external services for advanced natural language processing and specialized data storage. The system follows a Retrieval-Augmented Generation (RAG) and Agentic architecture to deliver accurate, context-aware responses.

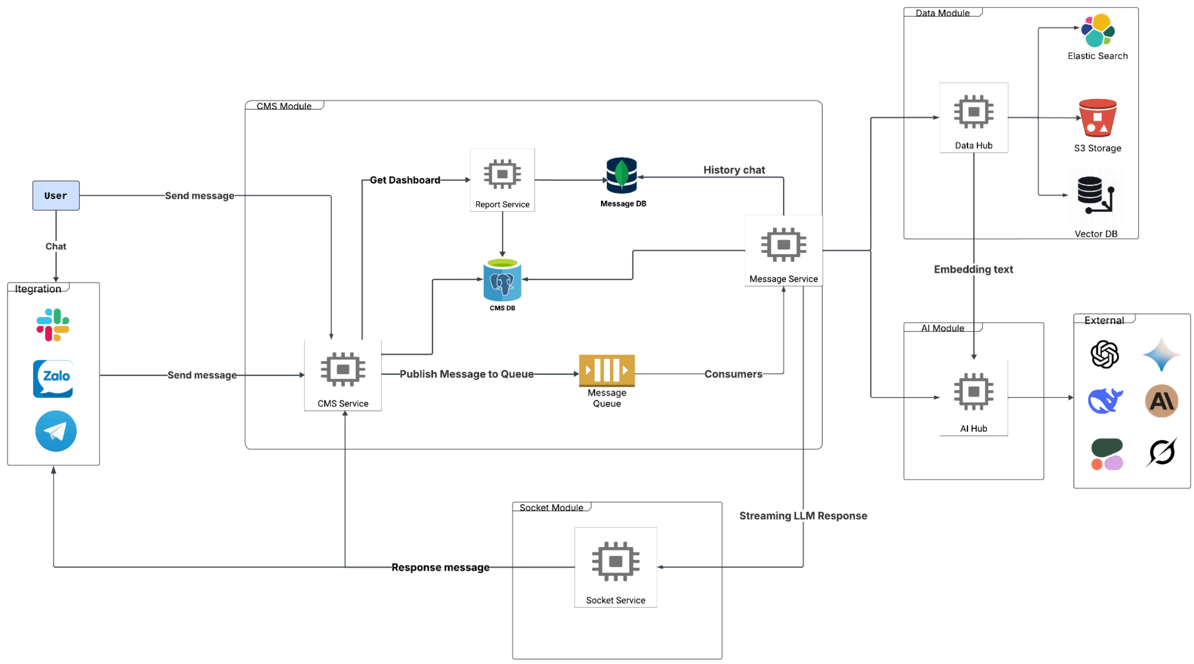


Figure 1.2.1 System Architecture Overview

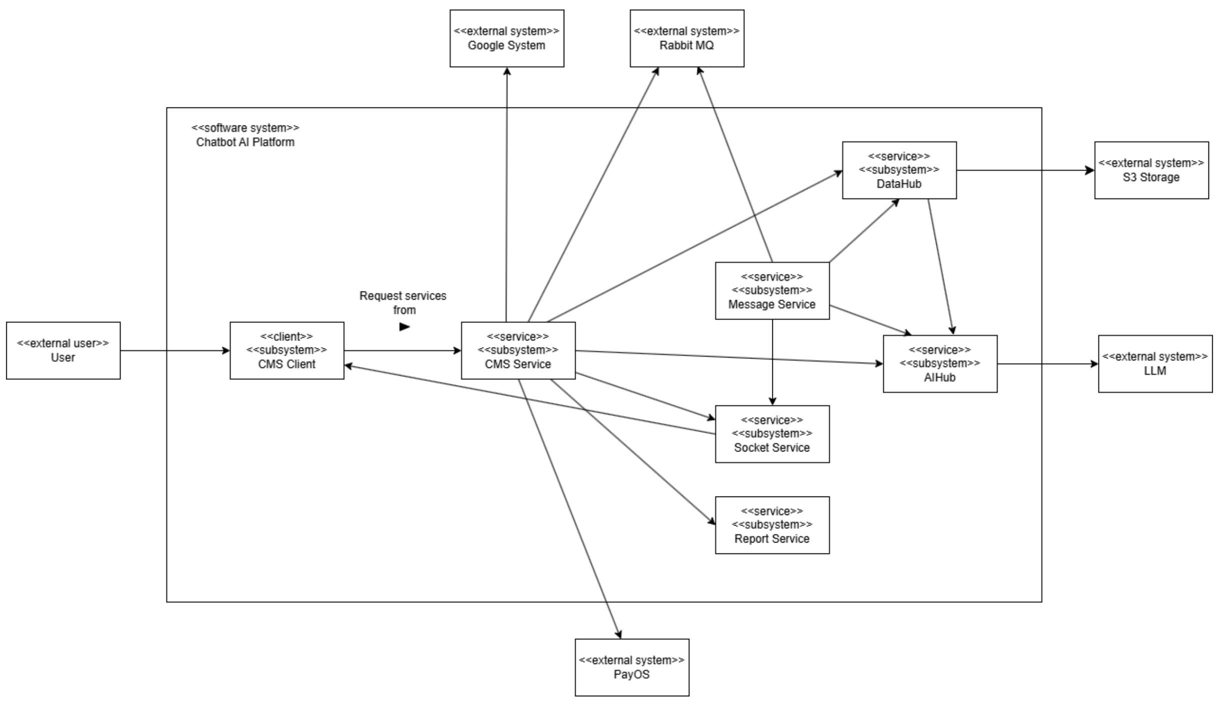


Figure 1.2.2 Structural View

b, Service Detail

No	Service	Description	Key Functions
1	**Internal Service**		

1.1	CMS Service (System Gateway and Management Core)	Content Management System Service serves as the primary entry point for user requests. It orchestrates and manages core system services, including chat management, bot configurations, organization details, payments, employee data, and business logic settings. All user interactions with the system begin at the CMS Service.	- Proxy receives user requests. - Manages system configurations (bots, organizations, users, permissions, etc.). - Provides APIs for other modules to access management data. - Forwards message-related requests to the Message Service.
1.2	Message Service (Message Flow Processing)	Message Service (Message Flow Processing) Description: The Message Service handles message flows from users after receiving them from the CMS Service. It acts as an intermediary, coordinating with DataHub for data retrieval and AIHub for natural language processing to analyze messages and generate appropriate responses. It also stores conversation history.	- Processes incoming messages from CMS Service. - Executes business logic for message handling. - Interacts with DataHub to retrieve relevant data. - Communicates with AIHub for NLP and response generation. - Stores conversation history in Message DB.
1.3	AIHub	AIHub centralizes the management of external large language models (LLMs), embedding models, and reranking models. It standardizes their input/output formats and provides a unified API for other services (primarily Message Service). AIHub also evaluates question complexity (via routeLLM) to select the most suitable model for processing.	- Manages and provides APIs for LLMs, embedding, and reranking models. - Standardizes input/output formats for AI models. - Routes questions to appropriate LLMs based on complexity (routeLLM).
1.4	DataHub	DataHub provides APIs to ingest user data (e.g., documents) and performs critical preprocessing tasks such as text chunking and indexing. Its primary goal is to optimize data retrieval, enabling the system to quickly and efficiently find relevant information for user queries.	- Ingests user data via APIs. - Performs text chunking. - Indexes processed data for efficient retrieval. - Provides APIs for Message Service to query processed data.
1.5	Socket Service	The Socket Service uses WebSocket to establish a real-time, bidirectional	- Manages WebSocket connections with user applications.

		communication channel between the server and user applications. It enables streaming responses from LLMs, enhancing the user experience with continuous, seamless interactions.	- Streams responses from AIHub (via Message Service) to users in real time.
1.6	Report Service	The Report Service collects message data from Message DB, preprocesses it (e.g., filtering, aggregation), and generates reports as needed. These reports provide insights into user interactions, system performance, and other business metrics.	- Retrieves message data from Message DB. - Preprocesses and aggregates data. - Generates reports in various formats.
1.7	CMS DB	CMS DB stores all configuration and business-related data, ensuring smooth operation of internal services.	- Stored data bot configurations, organization and member details, user permissions, and other system settings.
1.8	Message DB	Message DB stores the complete history of user-system conversations, providing critical data for tracking, analysis, and reporting.	- Message content, sender/receiver details, timestamps, and related metadata.
1.9	Message Queue	The Message Queue (RabbitMQ) manages asynchronous task processing for time-intensive operations, such as LLM response generation or data preprocessing. It ensures system stability by preventing service overload.	- Queues and distributes tasks reliably between services. - Supports asynchronous processing for improved performance.
2	**External Services**		
2.1	AI Service	AI Service includes LLMs, embedding models, and reranking models provided by third-party vendors. AIHub interacts with these services via APIs to leverage the latest advancements in natural language processing.	- APIs from OpenAI, Cohere, Hugging Face Inference API, etc.
2.2	Storage Service	Storage Service comprises optimized storage solutions for various data types, enhancing system efficiency and flexibility.	- S3 Storage: Object storage for user-uploaded files (e.g., documents). - Milvus Vector DB: Vector database for storing document embeddings, enabling fast similarity searches.

			- Elasticsearch: Search and analytics engine for text data, supporting high-speed, accurate full-text searches.

Table 1.2.1 Service Detail

III, Main Processing Flow

1. **User Request**: Users send messages via chatbox embedded into website or communication channels (e.g., Chat, Zalo, Telegram).
2. **CMS Service Handling:** CMS Service receives requests and initial chat session.
3. **Queueing:** CMS Service forwards messages to the Message Queue for asynchronous processing.
4. **Message Service Processing:** Message Service retrieves messages from the queue and initiates processing.
5. **AI and Data Interaction**: Message Service collaborates with AIHub for NLP tasks (e.g., embeddings, reranking, LLM responses) and DataHub for retrieving relevant data. AIHub and DataHub may interact with external AI and Storage Services.
6. **Response Delivery:** Responses are sent to users via Socket Service (for streaming) or CMS Service (for standard responses).
7. **Conversation Storage:** Message Service stores conversation history in Message DB.
8. **Reporting:** Report Service periodically retrieves and processes data from Message DB to generate analytical reports.

1.3 Agentic flow

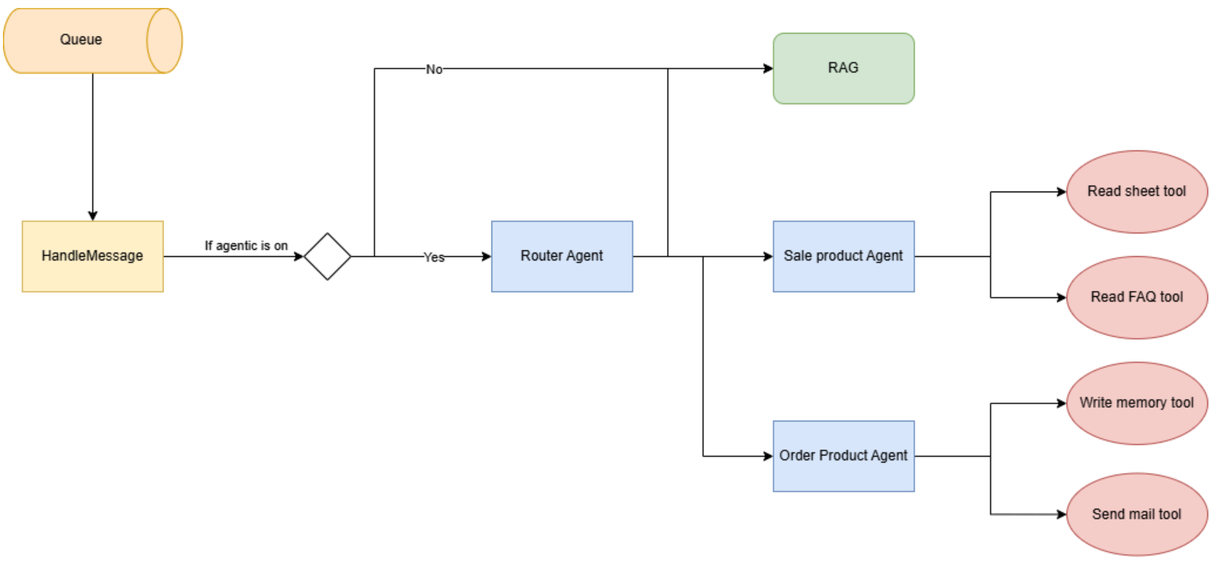


Figure 1.3.1 Agentic Flow

1.3.1 Agentic Flow Description

1. Queue: Incoming user messages are placed in a queue for processing.

2. HandleMessage: The HandleMessage service retrieves messages from the queue and prepares them for further handling.

Check: Is Agentic Mode Enabled

- If No then The system uses a RAG (Retrieval-Augmented Generation) model to handle the user's query directly.
- If Yes then The system proceeds with the agentic architecture.

3. Router Agent: This agent is responsible for routing messages to the appropriate specialized agents based on the message content. It forwards the task to:

- Sale Product Agent
- Order Product Agent

1.3.2 Functional Agents

1. Sale Product Agent

This agent handles requests related to product details, promotions, or sales information.
It uses the following tools:

- **Read Sheet Tool** – to fetch data from spreadsheets (e.g., Google Sheets)
- **Read FAQ Tool** – to answer frequently asked questions.

2. Order Product Agent

This agent manages order-related actions, such as placing or tracking orders.
It uses:

- **Write Memory Tool** – to store relevant user or order information.
- **Send Mail Tool** – to send confirmation or notification emails.

1.4 Analytic Flow

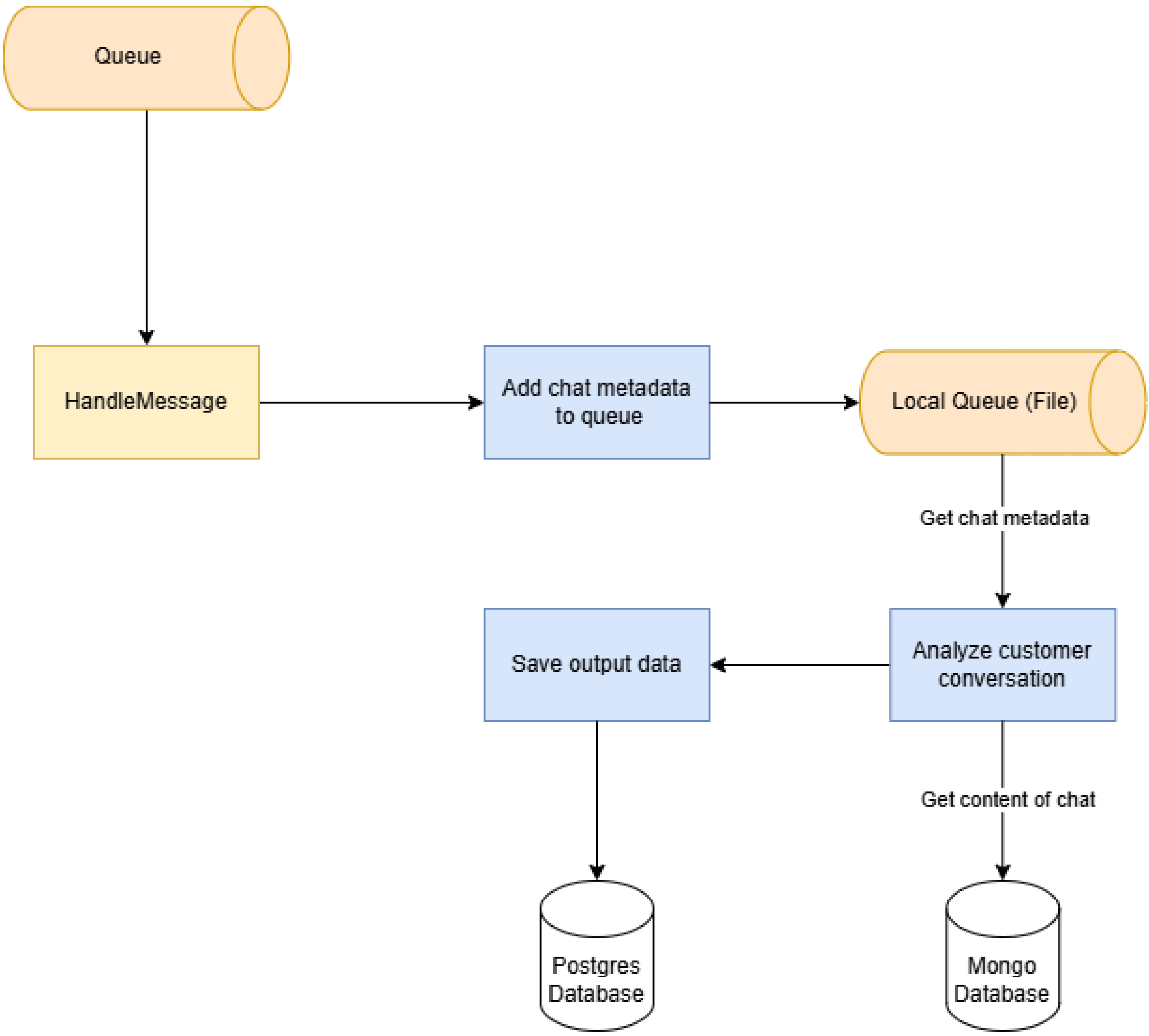


Figure 1.4.1 Analytic Flow

After handling a message then add chat metadata to the queue labeled "Local Queue" and store in block storage. Metadata contains: Chat Id, User Id, timestamp

When consumers get chat metadata for analyzing customer conversations, they retrieve the chat metadata from the queue and chat content from the mongo database.

In analyzing process conversation, it could involve tasks like sentiment analysis, topic detection, or identifying key customer issues. After that save the results of the analysis are then saved to a Postgres Database.

1.5 Package Diagram

1.5.1 CMS Service

a. CMS-BE

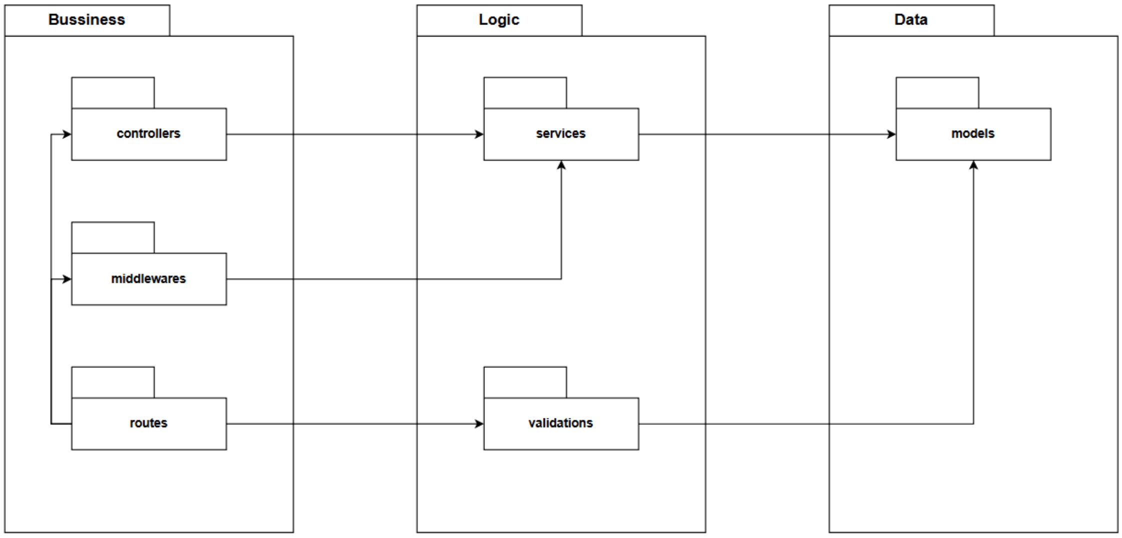


Figure 1.5.1.1 CMS-BE Package Diagram

Package descriptions

No	Package	Description
1	Routes	Defines API endpoints and their corresponding controllers.
2	Middleware	Handles request pre-processing, authentication, and logging.
3	Controllers	Manages incoming requests and interacts with services to process data
4	Services	Implements business logic and interacts with the data layer.
5	Validations	Ensures data integrity by validating input requests.
6	Models	Defines the structure of data entities and database schema.

Table 1.5.1.1 CMS-BE Package Description

b. CMS-FE

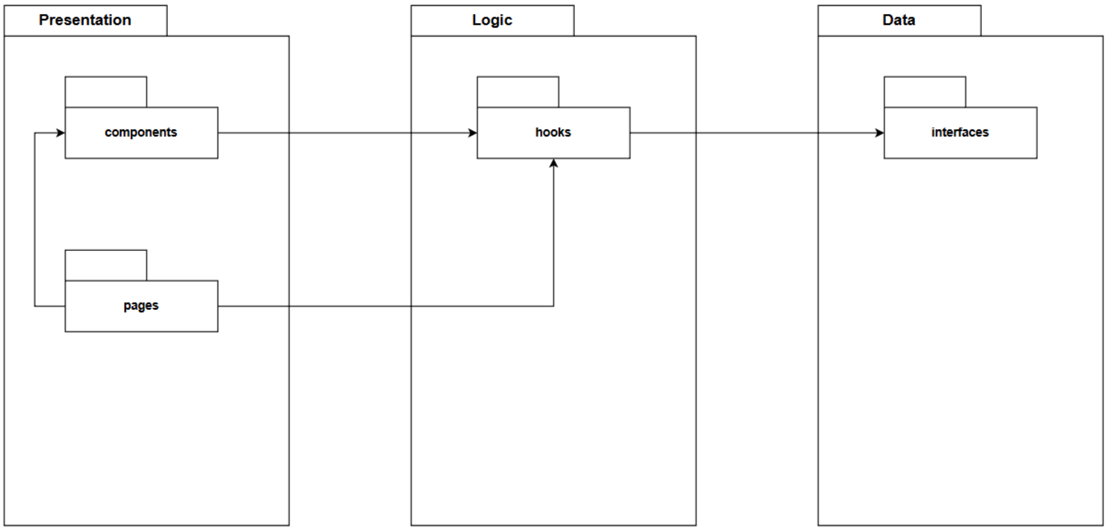


Figure 1.5.1.2 CMS-FE Package Diagram

Package descriptions

No	Package	Description
1	Components	Contains reusable UI elements used across different pages.
2	Pages	Represents different views or screens of the application.
3	Hooks	Custom React hooks for managing state and side effects.
4	Interfaces	Defines TypeScript interfaces for data models and API responses.

Table 1.5.1.2 CMS-FE Package Description

1.5.2 AI Hub

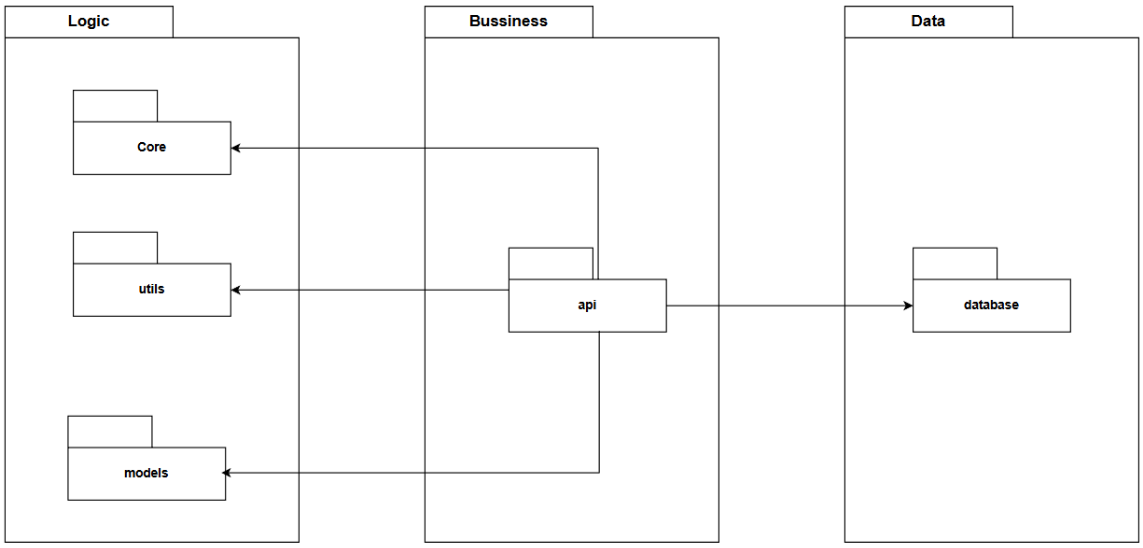


Figure 1.5.2.1 AI Hub Package Diagram

Package descriptions

No	Package	Description
1	Core	Implements the main business logic and operations.
2	Utils	Includes helper functions and utility methods.
3	Models	Defines data structures used throughout the application.
4	API	Handles communication between the application and external services or databases.
5	Database	Represents the data storage system.

Table 1.5.2.1 AI Hub Package Description

1.5.3 DataHub

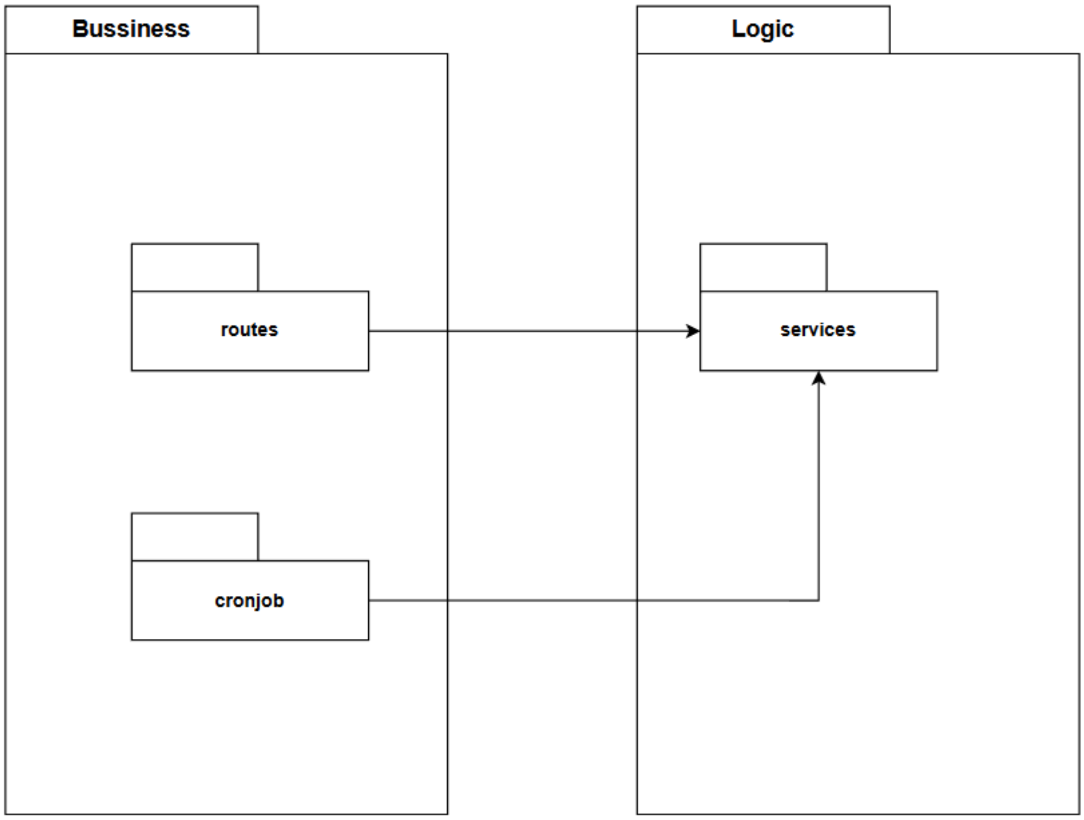


Figure 1.5.3.1 DataHub Package Diagram

Package descriptions

No	Package	Description
1	Routes	Defines API routes and handles incoming requests.
2	Croniob	Manages scheduled background tasks and automation.
3	Services	Implements business logic and interacts with other layers.

Table 1.5.3.1 DataHub Package Description

1.5.4 Message Service

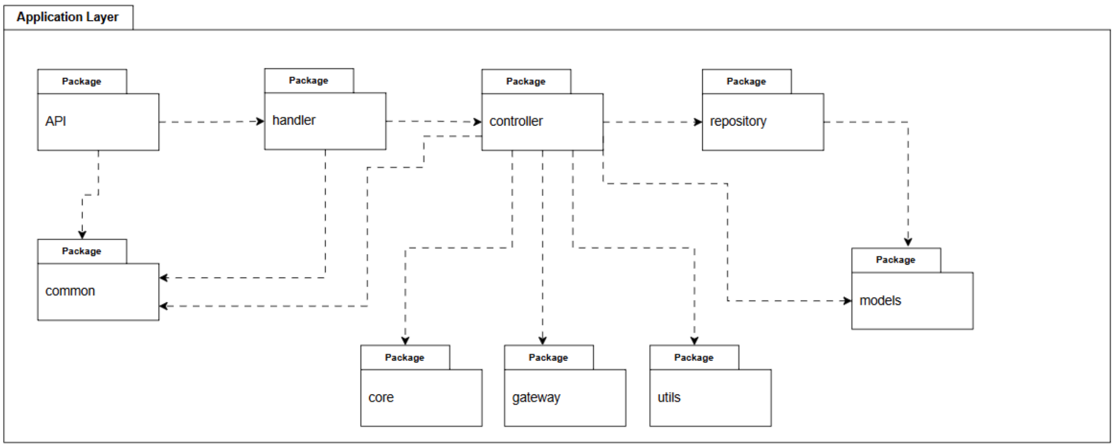


Figure 1.5.4.1 Message Service Package Diagram

Package descriptions

No	Package	Description
1	API	Manages external requests and API interactions.
2	Handler	Processes API requests before passing them to controllers.
3	Controller	Handles business logic and orchestrates requests.
4	Repository	Manages data access and persistence.
5	Common	Contains shared utilities and helpers.
6	Core	Handles core functionalities and business rules.
7	Gateway	Manages external service integrations.
8	Utils	Provides utility functions and reusable components.
9	Models	Defines data structures and database models.

Table 1.5.4.1 Message Service Package Description

1.5.5 Socket Service

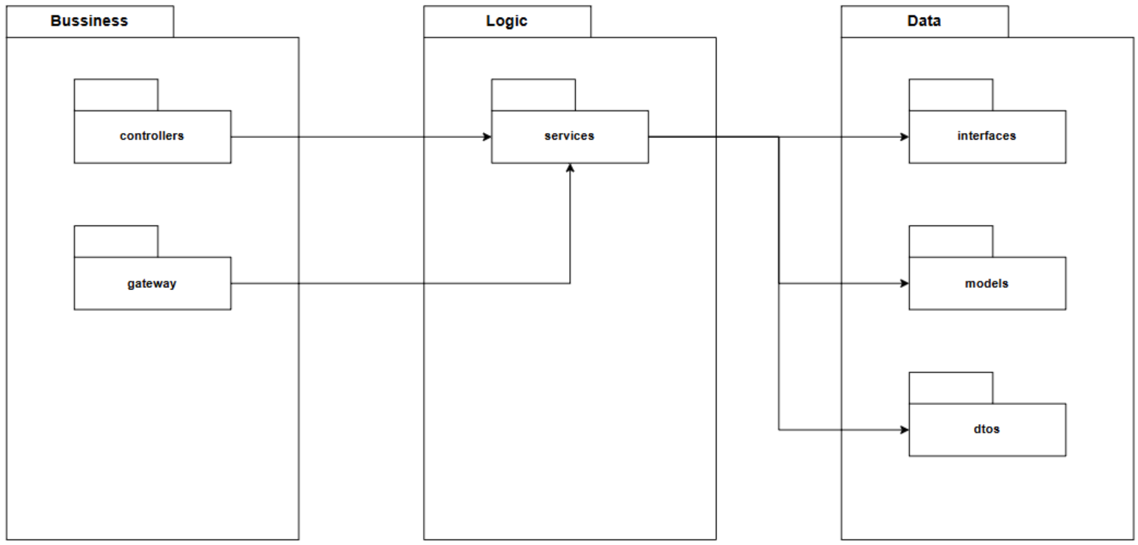


Figure 1.5.5.1 Socket Service Package Diagram

Package descriptions

No	Package	Description
1	Controllers	Handles HTTP requests and responses, orchestrates API logic.
2	Gateway	Manages external service calls and integrations.
3	Services	Contains business logic and processes data.
4	Interfaces	Defines contracts for data access and service implementations.
5	Models	Represents database entities and data structures.
6	DTOs	Data Transfer Objects used for communication between layers.

Table 1.5.5.1 Socket Service Package Description

1.5.6 Report Service

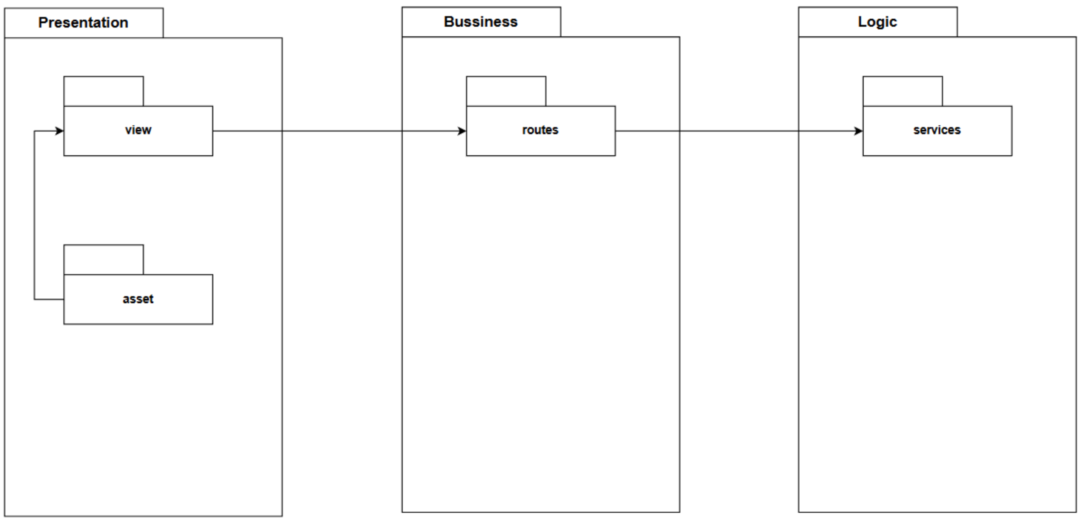


Figure 1.5.6.1 Report Service Package Diagram

Package descriptions

No	Package	Description
1	View	Responsible for rendering the interface presented to the user.
2	Asset	Provides static resources necessary for styling, interactivity, and media display.
4	Routes	Defines API routes and handles incoming requests.
3	Services	Implements business logic and interacts with other layers.

Figure 1.5.6.1 Report Service Package Description

2. Database Design

2.1 SQL Database (CMS DB)

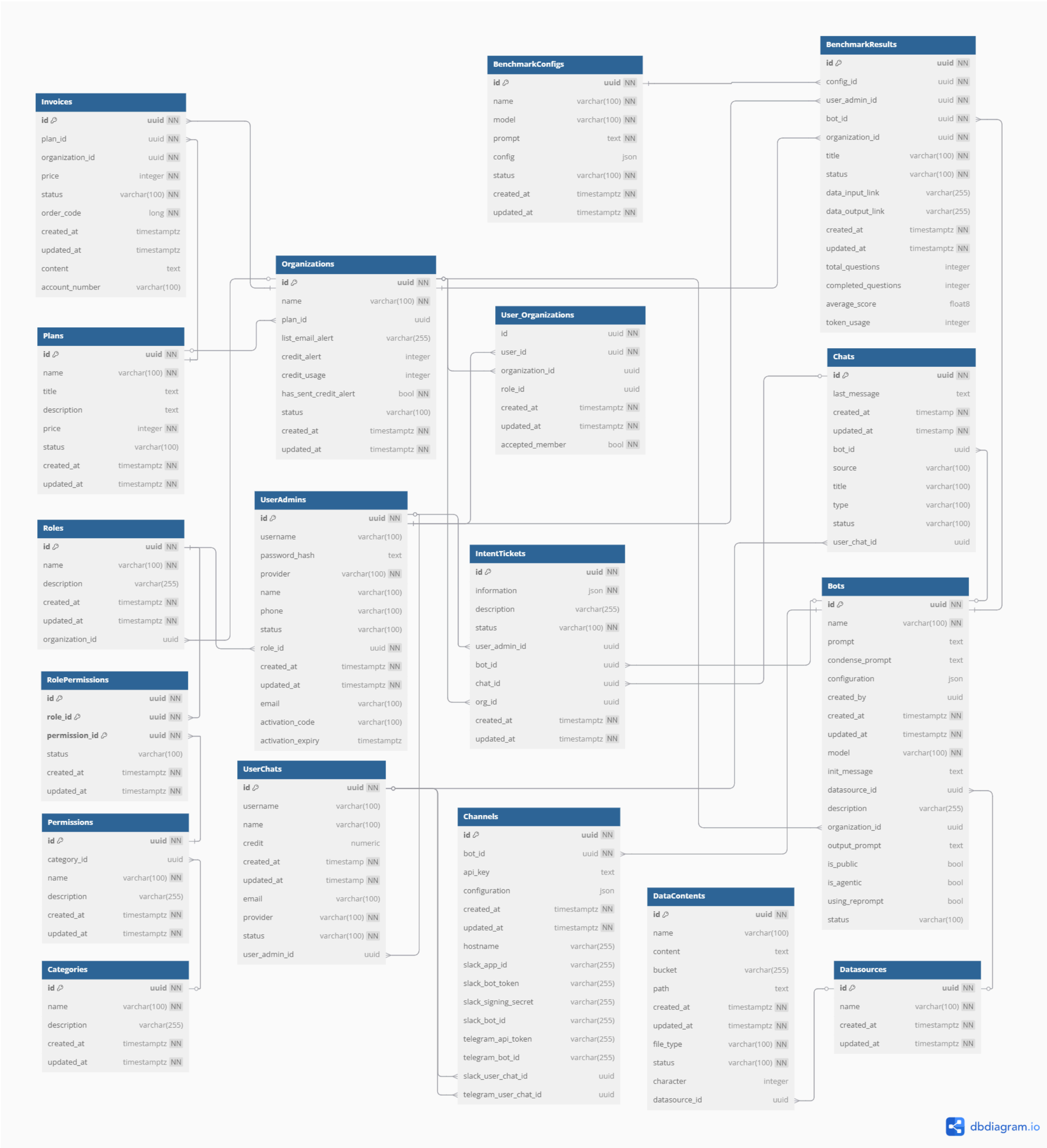


Figure 2.1.1 SQL Database Design

2.1.1 UserAdmins

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	username	varchar(100)	- Unique - Not null	
3	password_hash	text	- Not null	

4	provider	varchar(100)	- Not null	
5	name	varchar(100)		
6	phone	varchar(100)		
7	role_id	uuid	- Foreign key - Not null	
8	status	varchar(100)	- Not null	
9	created_at	timestamp	- Not null	
10	updated_at	timestamp	- Not null	
11	email	varchar(100)	- Not null	
12	activation_code	varchar(100)		
13	activation_expiry	timestamp		

Table 2.1.1.1 UserAdmins Description

2.1.2 Organizations

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)		
3	plan_id	uuid	- Foreign key - Not null	
4	status	varchar(100)	- Not null	
5	list_email_alert	varchar(255)		
6	credit_alert	integer		
7	credit_usage	integer		
8	has_sent_credit_alert	character varying(255)[]		
9	created_at	timestamp	- Not null	
10	updated_at	timestamp	- Not null	

Table 2.1.2.1 Organizations Description

2.1.3 User_Organizations

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	user_id	uuid	- Foreign key - Not null	
3	organization_id	uuid	- Foreign key - Not null	
4	role_id	uuid	- Foreign key	
5	accepted_member	boolean		
6	created_at	timestamp	- Not null	
7	updated_at	timestamp	- Not null	

Table 2.1.3.1 User_Organizations Description

2.1.4 Bots

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)		
3	model_id	uuid	- Foreign key - Not null	
4	prompt	text		
5	condense_prompt	text		
6	configuration	text		
7	init_message	text		
8	datasource_id	uuid	- Foreign key	
9	description	varchar(255)		
10	organization_id	uuid	- Foreign key	
11	output_prompt	text		
12	is_public	boolean		
13	is_agentic	boolean		
14	using_reprompt	boolean		
15	status	varchar(100)	- Not null	
16	created_at	timestamp	- Not null	
17	created_by	timestamp	- Not null	
18	updated_at	timestamp		

Table 2.1.4.1 Bots Description

2.1.5 Channels

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	api_key	text	- Not null	
3	host_name	varchar(255)		
4	configuration	json		
5	bot_id	uuid	- Foreign key - Not null	
6	slack_app_id	varchar(255)		
7	slack_bot_token	varchar(255)		
8	slack_signing_secret	varchar(255)		
9	slack_bot_id	varchar(255)		
10	telegram_api_token	varchar(255)		
11	telegram_bot_id	varchar(255)		
12	slack_user_chat_id	uuid	- Foreign key	
13	telegram_user_chat_id	uuid	- Foreign key	
14	created_at	timestamp	- Not null	
15	updated_at	timestamp		

Table 2.1.5.1 Channels Description

2.1.6 Chats

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	user_chat_id	uuid	- Foreign key	
3	bot_id	uuid	- Foreign key	
4	source	varchar(100)		
5	title	varchar(100)		
6	status	varchar(100)	- Not null	
7	last_message	text		
8	type	varchar(100)		
9	created_at	timestamp	- Not null	
10	updated_at	timestamp	- Not null	

Table 2.1.6.1 Chats Description

2.1.7 Datasources

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)	- Not null	
4	created_at	timestamp	- Not null	
6	updated_at	timestamp	- Not null	

Table 2.1.7.1 Datasources Description

2.1.8 DataContents

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)	- Not null	
3	content	text	- Not null	
4	bucket	varchar(255)		
5	path	varchar(255)		
6	file_type	varchar(100)	- Not null	
7	character	integer		
8	datasource_id	uuid	- Foreign key - Not null	
9	status	varchar(100)		
10	created_at	timestamp	- Not null	
11	updated_at	timestamp		

Table 2.1.8.1 DataContents Description

2.1.9 Plans

#	Field Name	Type	Constraint	Note

1	id	uuid	- Primary key - Not null	
2	name	varchar(100)	- Not null	
3	title	text		
4	description	text		
5	price	integer	- Not null	
6	status	varchar(100)	- Not null	
7	created_at	timestamp	- Not null	
8	updated_at	timestamp	- Not null	

Table 2.1.9.1 Plans Description

2.1.10 Invoices

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	organization_id	uuid	- Foreign key - Not null	
3	plan_id	uuid	- Foreign key - Not null	
4	price	integer	- Not null	
5	content	text		
6	account_number	varchar(100)		
7	status	varchar(100)	- Not null	
8	order_code	long	- Not null	
9	created_at	timestamp	- Not null	
10	updated_at	timestamp	- Not null	

Table 2.1.10.1 Invoices Description

2.1.11 UserChats

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)		
3	username	varchar(100)		
4	credit	decimal(10,2)	- Not null	
5	email	varchar(100)		
6	provider	varchar(100)		
7	user_admin_id	uuid	- Foreign key	
8	status	varchar(100)	- Not null	
9	created_at	timestamp	- Not null	
10	updated_at	timestamp	- Not null	

Table 2.1.11.1 UserChats Description

2.1.12 Categories

#	Field Name	Type	Constraint	Note

1	id	uuid	- Primary key - Not null	
2	name	varchar(100)	- Not null	
3	description	varchar(255)		
4	created_at	timestamp	- Not null	
5	updated_at	timestamp	- Not null	

Table 2.1.12.1 Categories Description

2.1.13 Roles

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)	- Not null	
3	description	varchar(255)		
4	organization_id	uuid	- Foreign key	
5	created_at	timestamp	- Not null	
6	updated_at	timestamp	- Not null	

Table 2.1.13.1 Roles Description

2.1.14 Permissions

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	name	varchar(100)	- Not null	
3	description	varchar(255)		
4	category_id	uuid	- Foreign key	
5	created_at	timestamp	- Not null	
6	updated_at	timestamp	- Not null	

Table 2.1.14.1 Permissions Description

2.1.15 RolePermissions

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	role_id	uuid	- Foreign key - Not null	
3	permission_id	uuid	- Foreign key - Not null	
4	status	varchar(100)		
5	created_at	timestamp	- Not null	
7	updated_at	timestamp	- Not null	

Table 2.1.15.1 RolePermissions Description

2.1.16 IntentTickets

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key	

			- Not null	
2	information	json		
3	description	varchar(255)		
4	status	varchar(100)		
5	user_admin_id	uuid	- Foreign key	
6	bot_id	uuid	- Foreign key	
7	chat_id	uuid	- Foreign key	
8	org_id	uuid	- Foreign key	
9	created_at	timestamp	- Not null	
10	updated_at	timestamp	- Not null	

Table 2.1.16.1 IntentTickets Description

2.1.17 BenchmarkConfigs

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	org_id	uuid	- Foreign key - Not null	
3	name	varchar(100)	- Not null	
4	model	varchar(100)	- Not null	
5	prompt	text		
6	config	json		
7	status	varchar(100)	- Not null	
8	created_at	timestamp	- Not null	
9	updated_at	timestamp	- Not null	

Table 2.1.17.1 BenchmarkConfigs Description

2.1.18 BenchmarkResults

#	Field Name	Type	Constraint	Note
1	id	uuid	- Primary key - Not null	
2	user_admin_id	uuid	- Foreign key - Not null	
3	organization_id	uuid	- Foreign key - Not null	
4	bot_id	uuid	- Foreign key - Not null	
5	config_id	uuid	- Foreign key - Not null	
6	title	varchar(100)	- Not null	
7	status	varchar(100)	- Not null	
8	data_input_link	varchar(255)		
9	data_output_link	varchar(255)		
10	total_questions	integer		

11	completed_questions	integer		
12	average_score	float		
13	token_usage	integer		
14	created_at	timestamp	- Not null	
15	updated_at	timestamp	- Not null	

Table 2.1.18.1 BenchmarkResults Description

2.2 No-SQL Database (Message DB)

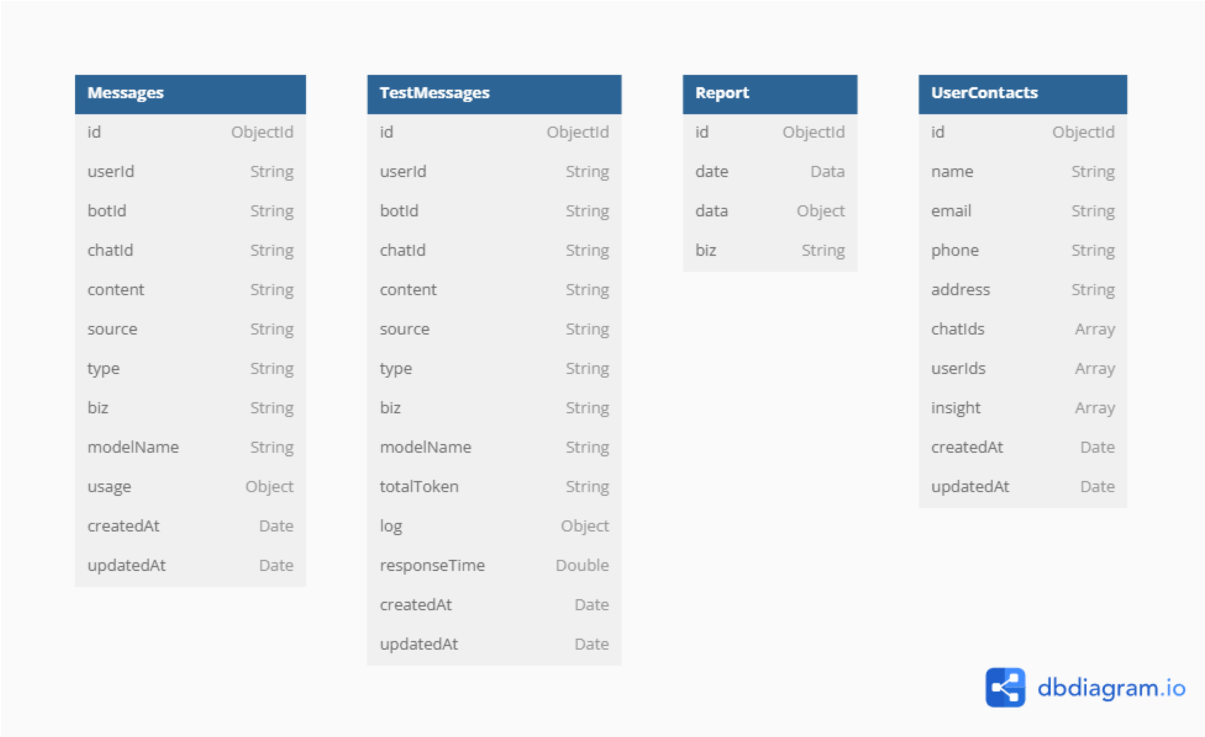


Figure 2.2.1 No-SQL Database Design

2.2.1 Messages

#	Field Name	Type	Constraint	Note
1	id	ObjectId	- Primary key - Not null	
2	userId	String	- Not null	
3	botId	String	- Not null	
4	chatId	String	- Not null	
5	content	String		
6	source	String		
7	type	String		
8	biz	String		
9	modelName	String		
10	usage	Object		
11	createdAt	Date	- Not null	

12	updatedAt	Date	- Not null	

Table 2.2.1.1 Messages Description

2.2.2 TestMessages

#	Field Name	Type	Constraint	Note
1	id	ObjectId	- Primary key - Not null	
2	userId	String	- Not null	
3	botId	String	- Not null	
4	chatId	String	- Not null	
5	content	String	- Not null	
6	source	String		
7	type	String		
8	biz	String		
9	modelName	String		
10	totalToken	Integer		
11	log	Object		
12	responseTime	Double		
13	createdAt	Date	- Not null	
14	updatedAt	Date	- Not null	

Table 2.2.2.1 TestMessages Description

2.2.3 Report

#	Field Name	Type	Constraint	Note
1	id	ObjectId	- Primary key - Not null	
2	date	Date		
3	data	Object		
4	biz	String		

Table 2.2.3.1 Report Description

2.2.4 UserContacts

#	Field Name	Type	Constraint	Note
1	id	ObjectId	- Primary key - Not null	
2	name	Date		
3	email	Object		
4	phone	String		
5	address	String		
6	chatIds	Array		
7	userIds	Array		
8	insight	Array		
9	createdAt	Date	- Not null	
10	updatedAt	Date	- Not null	

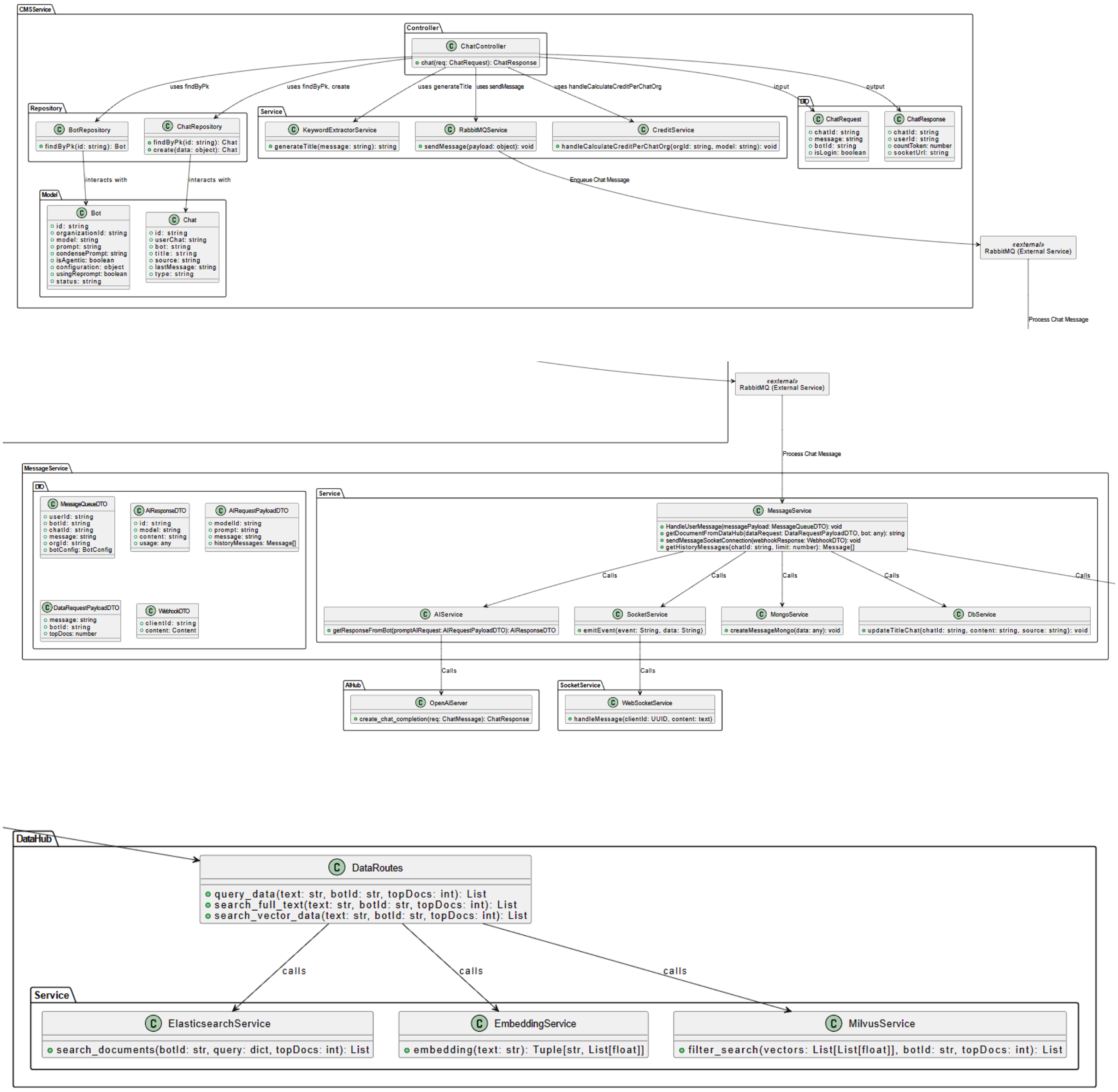
Table 2.2.4.1 UserContacts Description

3. Detailed Design

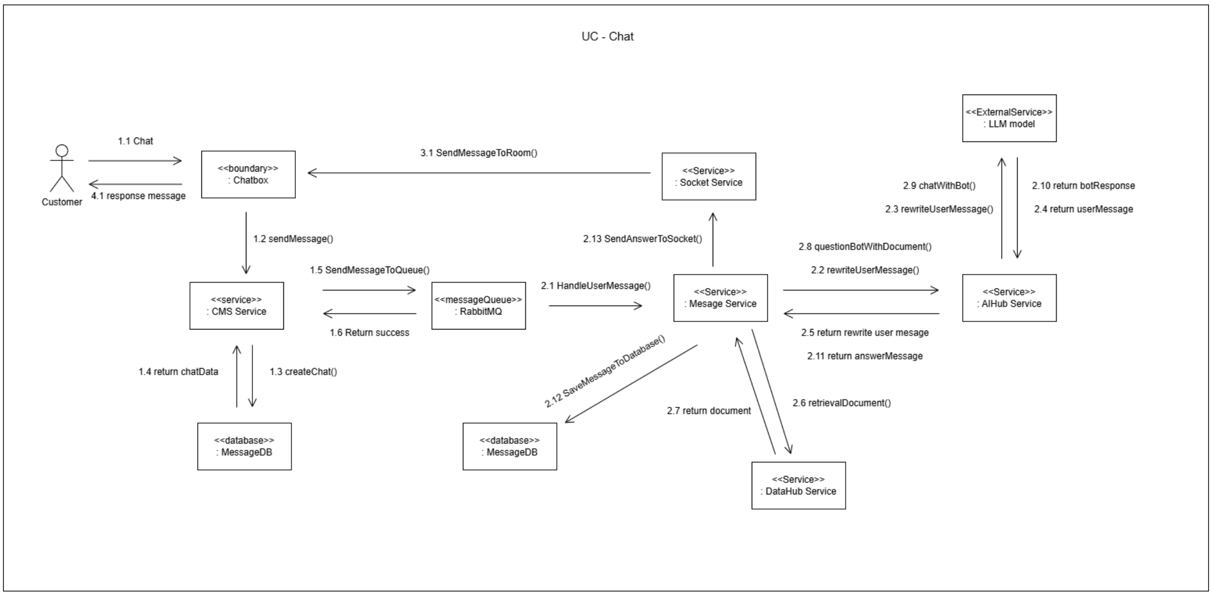
3.1 AI Interaction

3.1.1 Chat

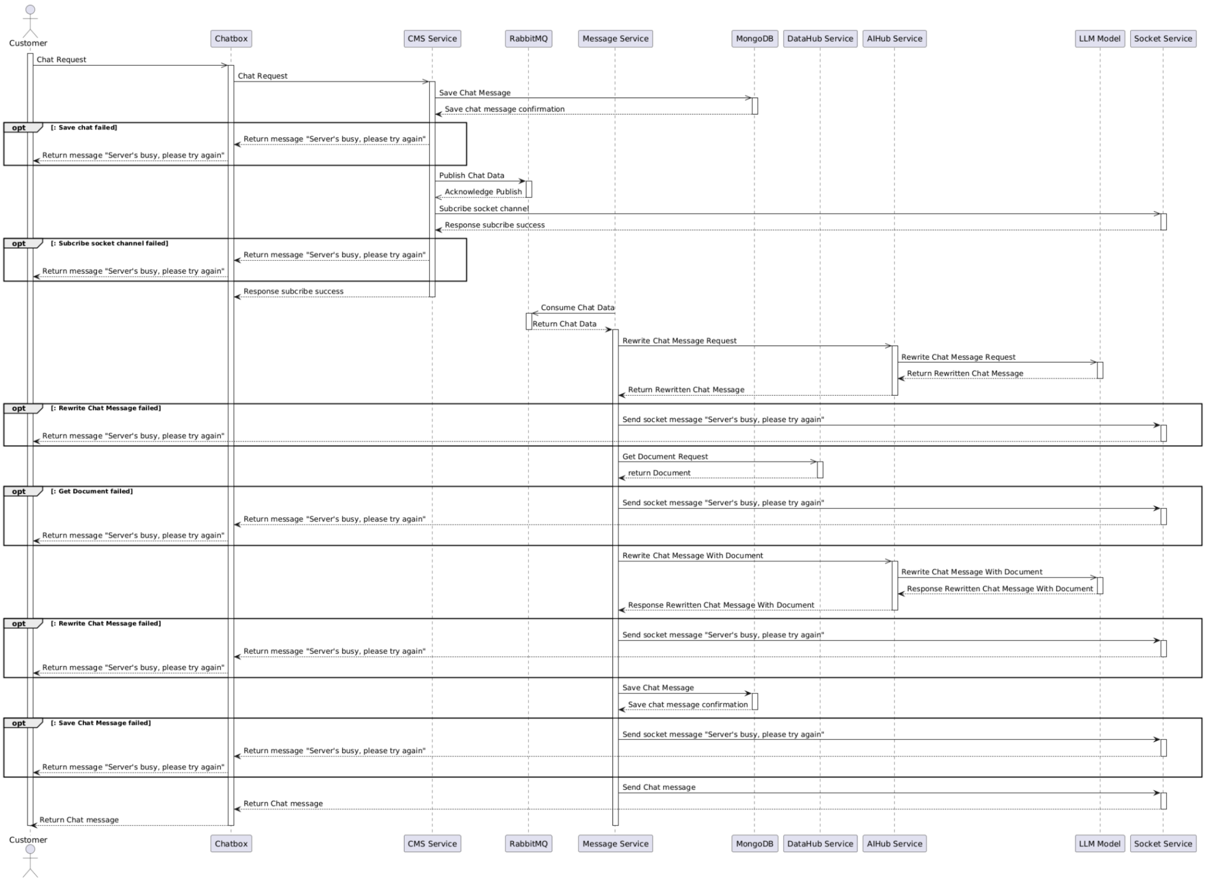
a, Class Diagram



b, Communication Diagram



c, Sequence Diagram



Components

1. **Chatbox**– Receives messages from the customer and forwards them to the backend.
2. **CMS Service** – Handles backend processing and communicates with other services.

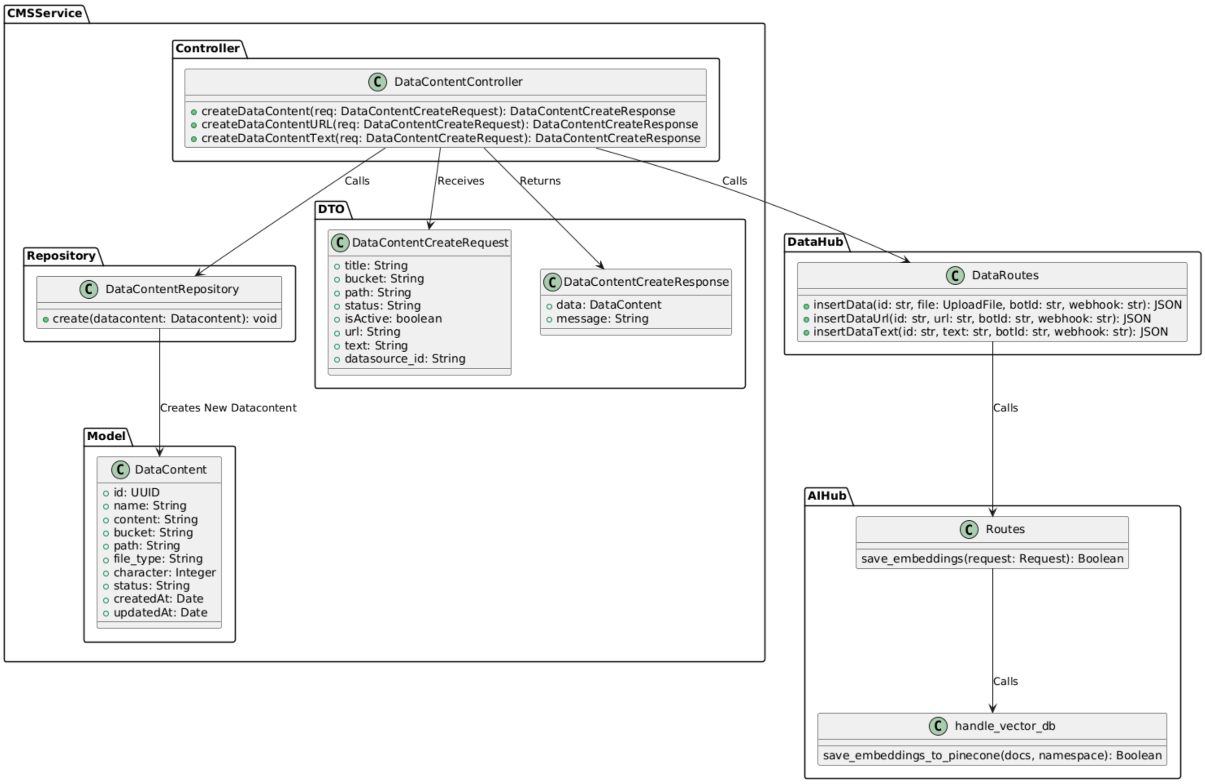
3. **RabbitMQ** – Manages asynchronous tasks by enqueueing messages for processing.
4. **Message Service** – Processes user messages, retrieves documents, and handles bot responses.
5. **Socket Service** – Enables real-time message delivery between components.
6. **AIHub**– Processes user queries and generates bot responses using AI.
7. **External Service (LLM Model)** – Provides language model capabilities for generating bot responses.
8. **DataHub** – Manages document retrieval to support bot responses.

Process Flow

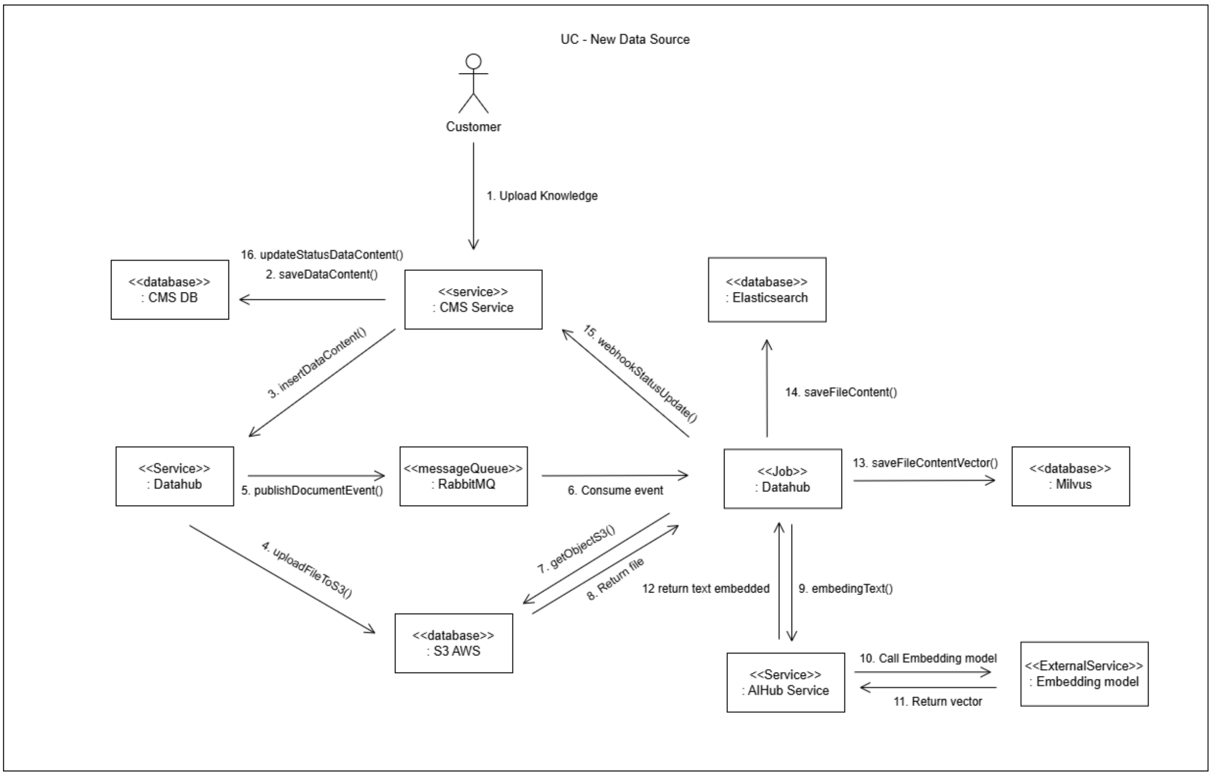
1. Customer Starts Chatting:
    - The customer sends a message through the Chatbox.
2. Chatbox Forwards the Message:
    - The Chatbox sends the message to the CMS Service. The CMS Service saves the chat info in the MessageDB and pushes the message to RabbitMQ.
3. RabbitMQ Handles the Message:
    - The Message Service picks up the message from RabbitMQ, processes it, and sends the rewritten message through the Socket Service to display it back on the Chatbox.
4. Fetch Documents and Ask the Bot:
    - The Message Service grabs the message from the MessageDB, asks the DataHub Service for related documents, and sends the question to the AIHub Service. The AIHub Service then asks the External Service (LLM) to generate a bot response.
5. Send the Bot's Response Back:
    - The AIHub Service gives the bot's response to the Message Service. The Message Service saves it in the MessageDB and sends it through the Socket Service to show on the Chatbox for the customer to see.

3.1.2 Add New Data Source

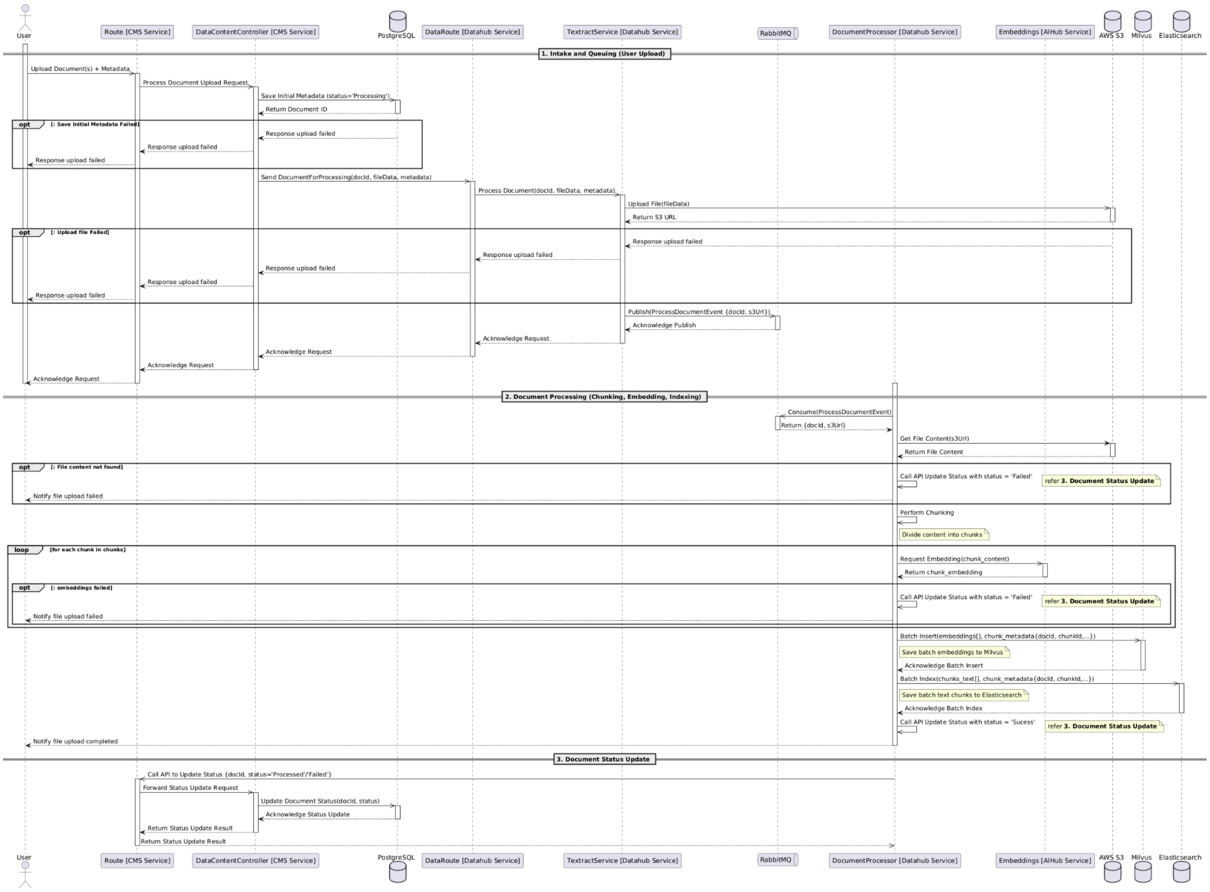
a, Class Diagram



b, Communication Diagram



c, Sequence Diagram



Components

1. Client – Initiates the file insertion process.
2. ChatBox App – Receives the file from the client and forwards it to the backend.
3. CMS Service – Handles the backend processing and communicates with other services.
4. CMS DB – Stores metadata related to the data source.
5. Data Hub – Manages data storage and retrieval, integrating with other components.
6. Queue – Manages asynchronous tasks by enqueueing file metadata.
7. AI Hub – Processes text extraction and embedding.
8. Vector Store – Stores vectorized representations of the extracted text.
9. Elastic Search – Enables full-text search capabilities.
10. Object Storage – Stores the actual file content.

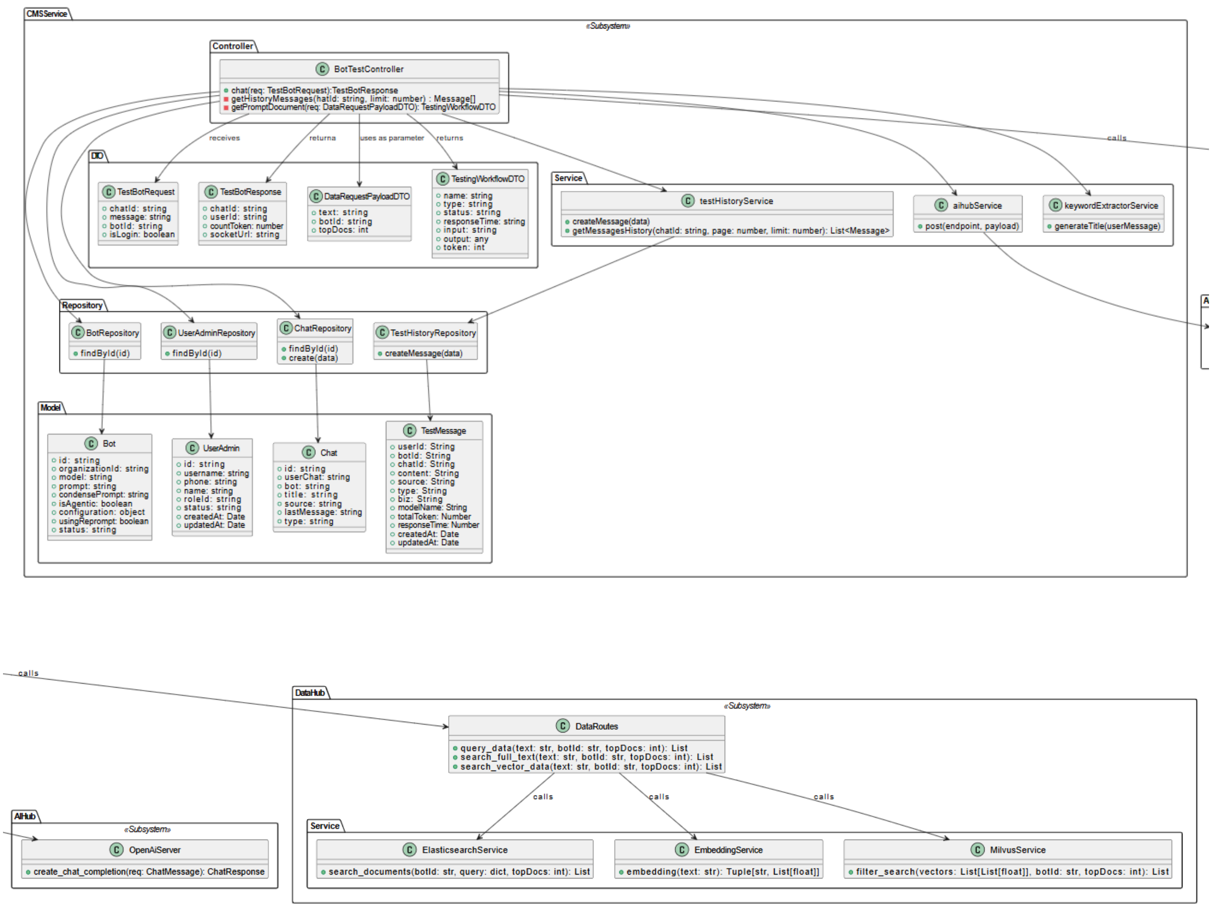
Process Flow

1. Client Uploads File
   - The client submits a file to the CMS Service.
2. File Forwarding & Metadata Storage
   - The CMS Service creates a new data source entry in the DataSource DB.
   - The file is forwarded to the Data Hub.
3. Queue Processing
   - The Data Hub enqueues the file metadata into a message queue.
   - The system polls the queue for new file metadata.

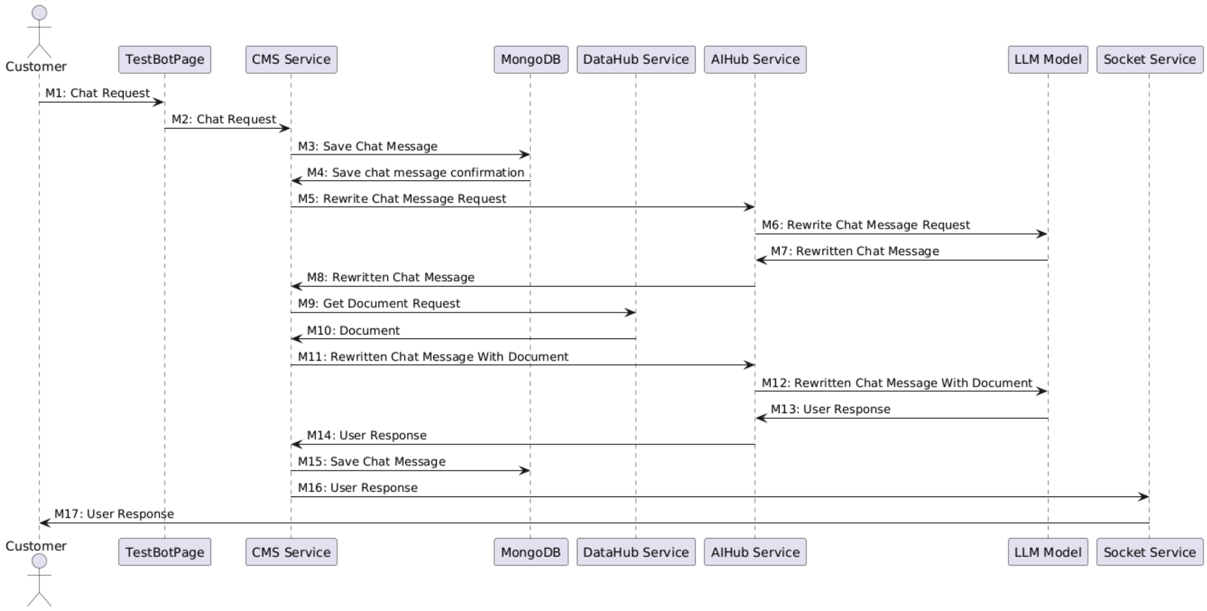
4. File Retrieval & Text Extraction
    - The system requests the file from Object Storage.
    - Upon retrieval, text is extracted from the file.
5. Embedding & Storage
    - The AI Hub processes the extracted text and generates vector embeddings.
    - The vector embeddings and raw text are stored in the Vector Store.
    - The raw text is stored in Elastic Search for full-text search.
6. Completion & Status Update
    - The system updates the status in the Data Hub and DataSource DB.
    - The CMS Service receives the updated status and informs the ChatBox App.
    - The ChatBox App notifies the client of the file insertion status.

3.1.3 Test Bot

a, Class Diagram



b, Sequence Diagram

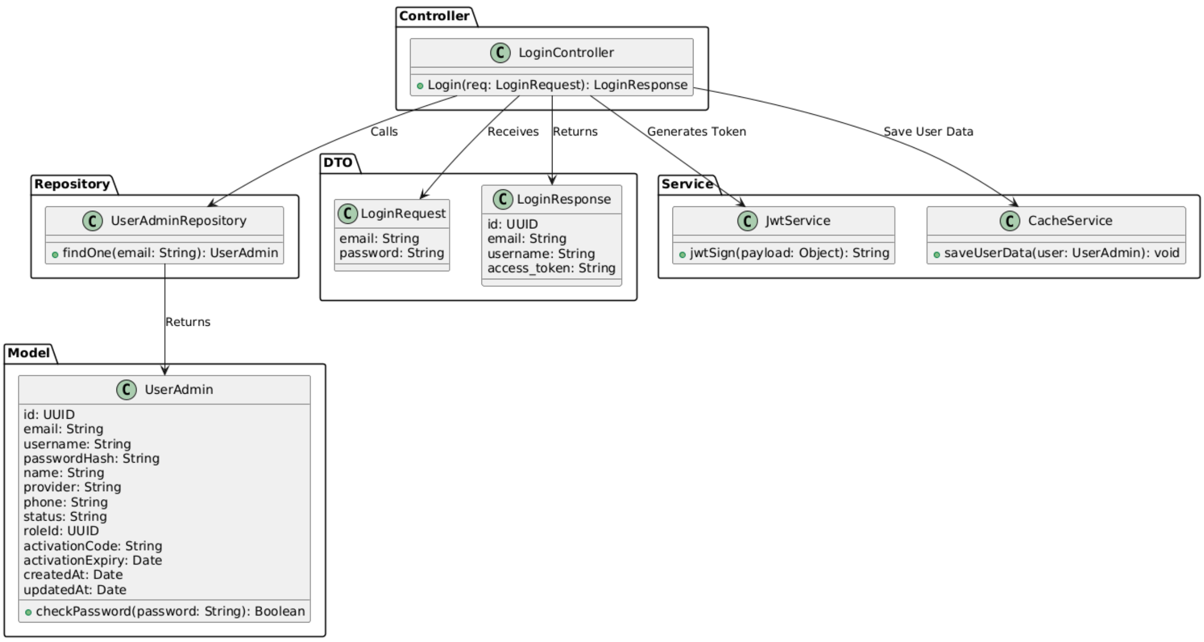


3.2 Common Feature

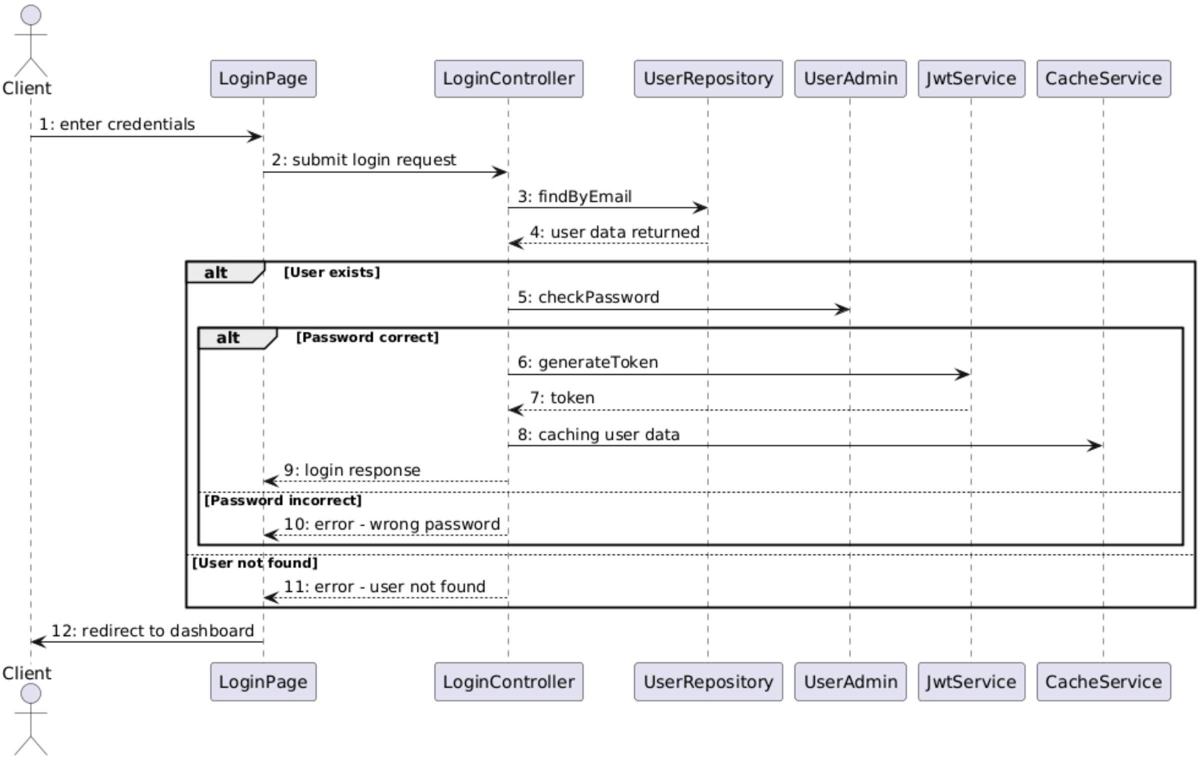
3.2.1 Login

3.2.1.1 Login

a, Class Diagram

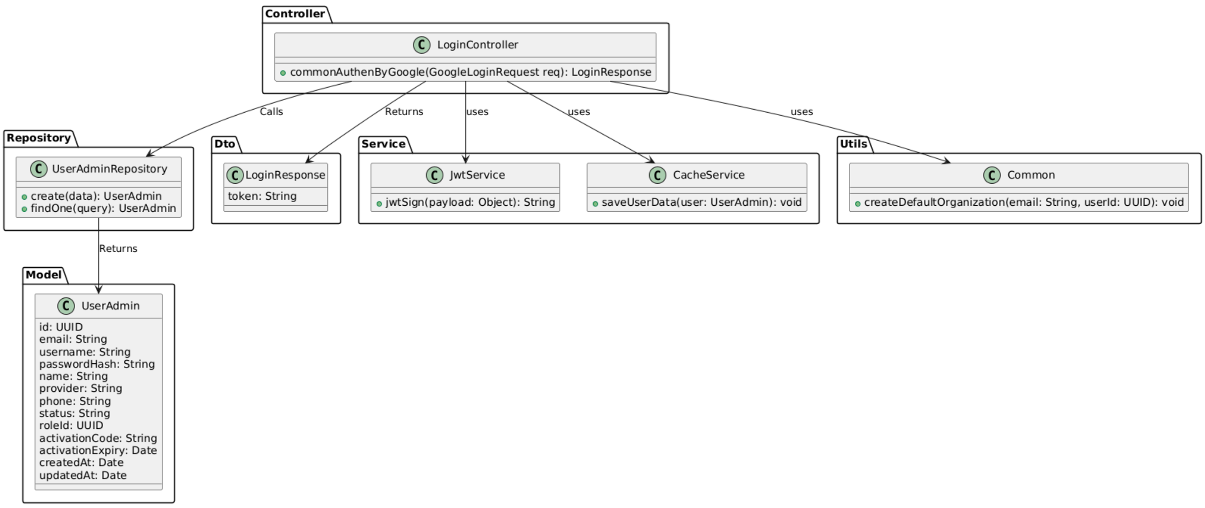


b, Sequence Diagram

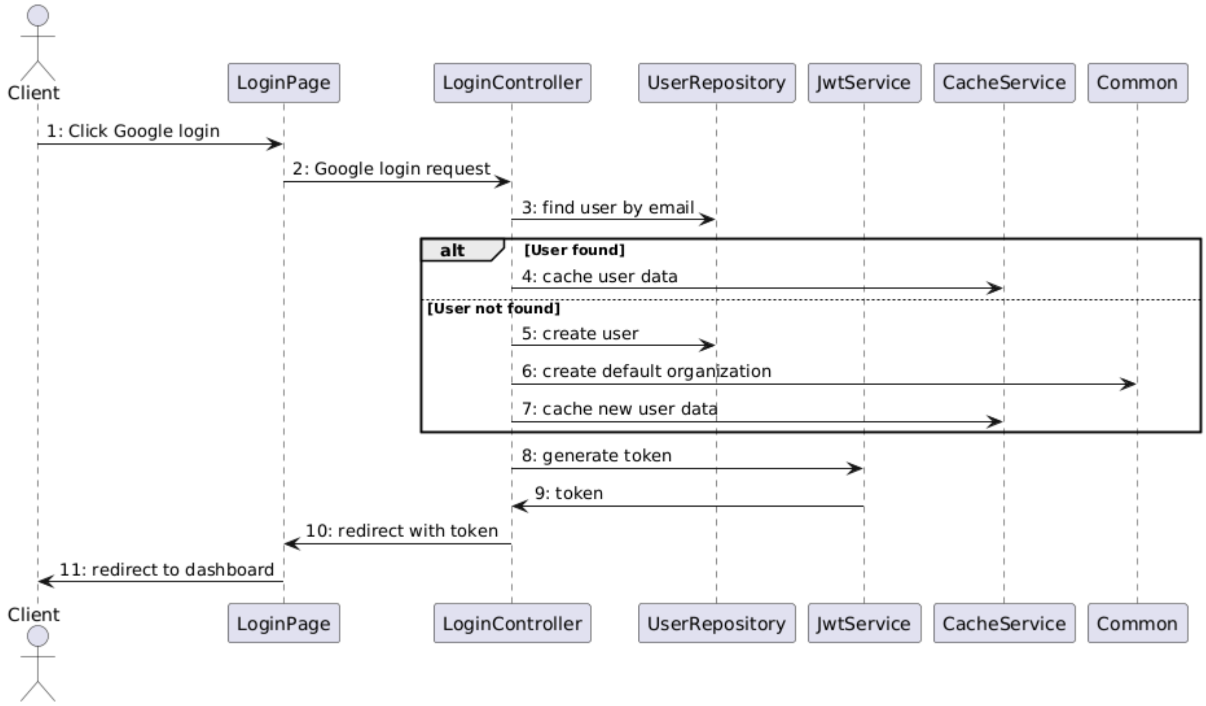


3.2.1.2 Login With Google

a, Class Diagram

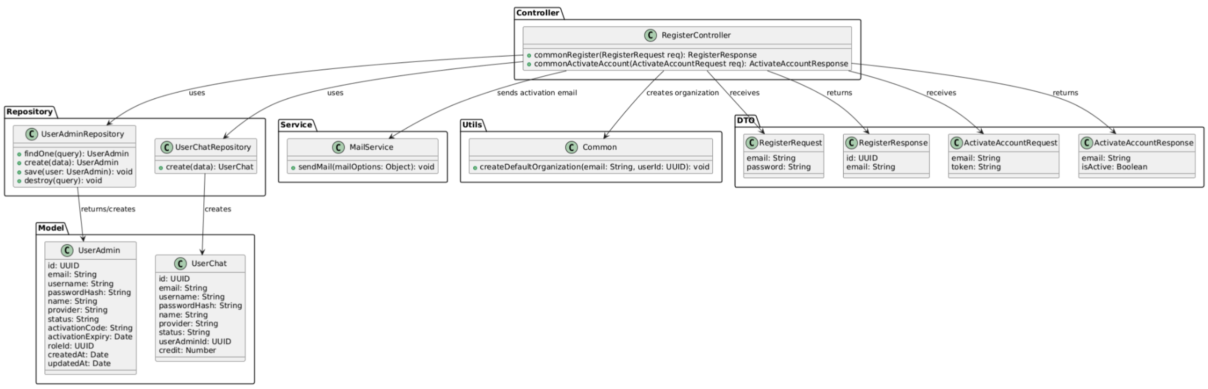


b, Sequence Diagram

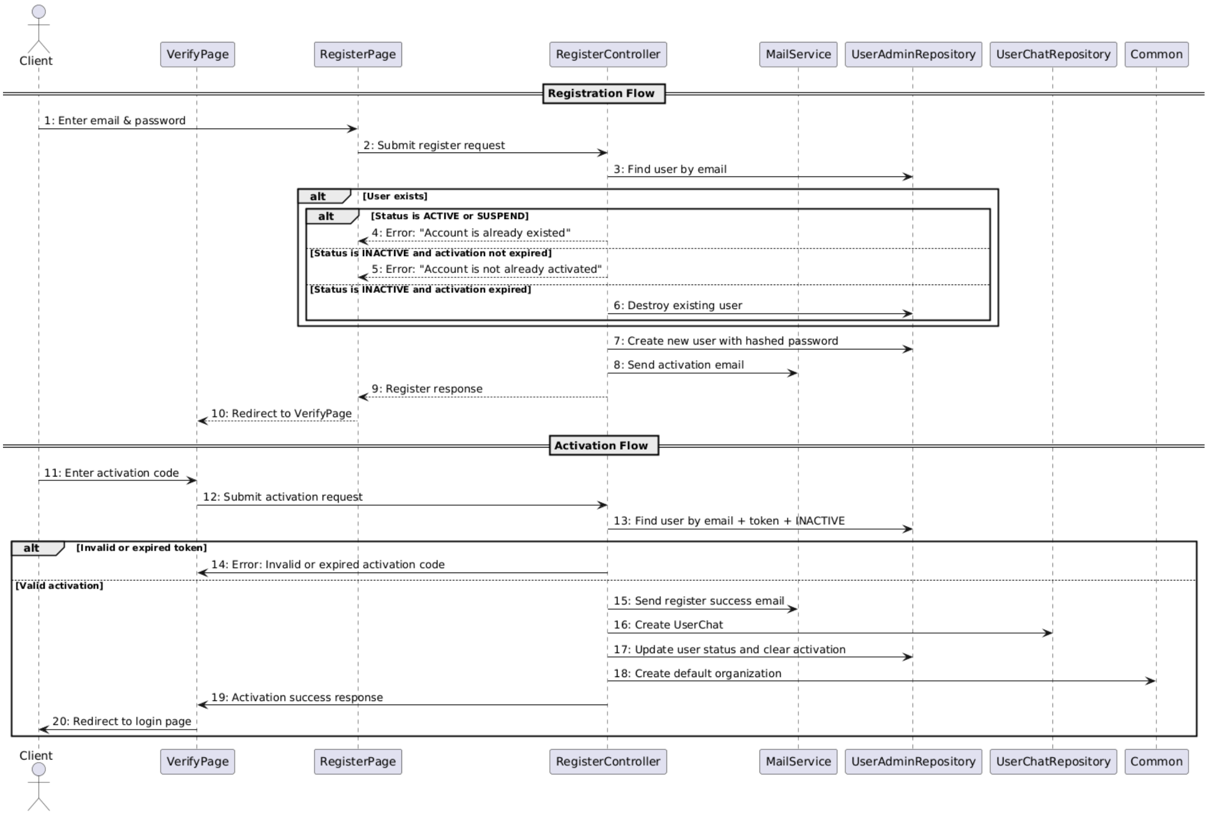


3.2.2 Register

a, Class Diagram

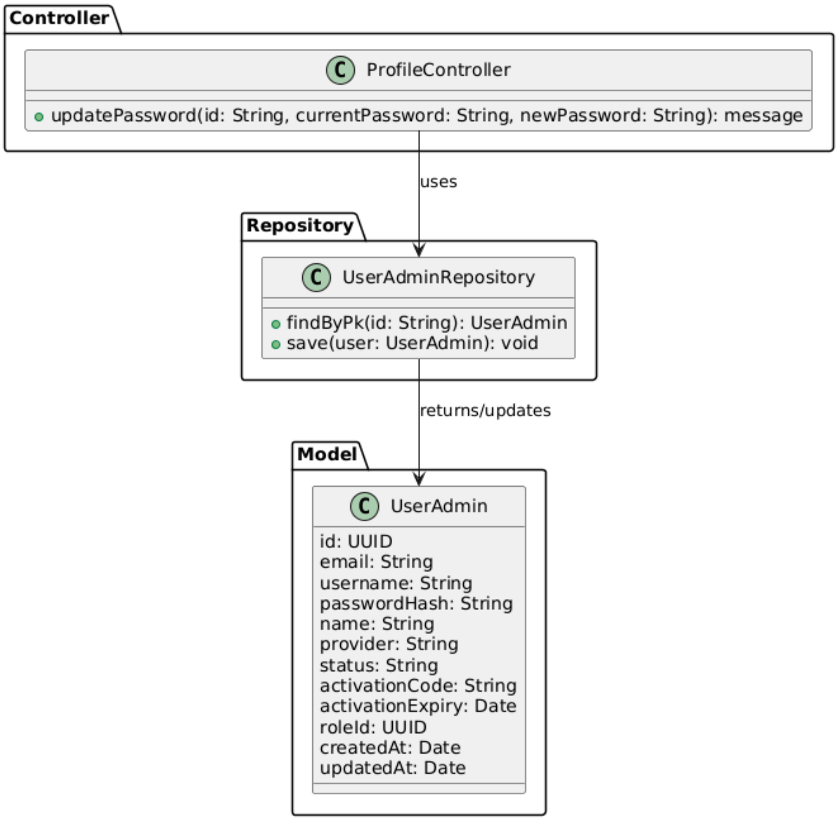


b, Sequence Diagram

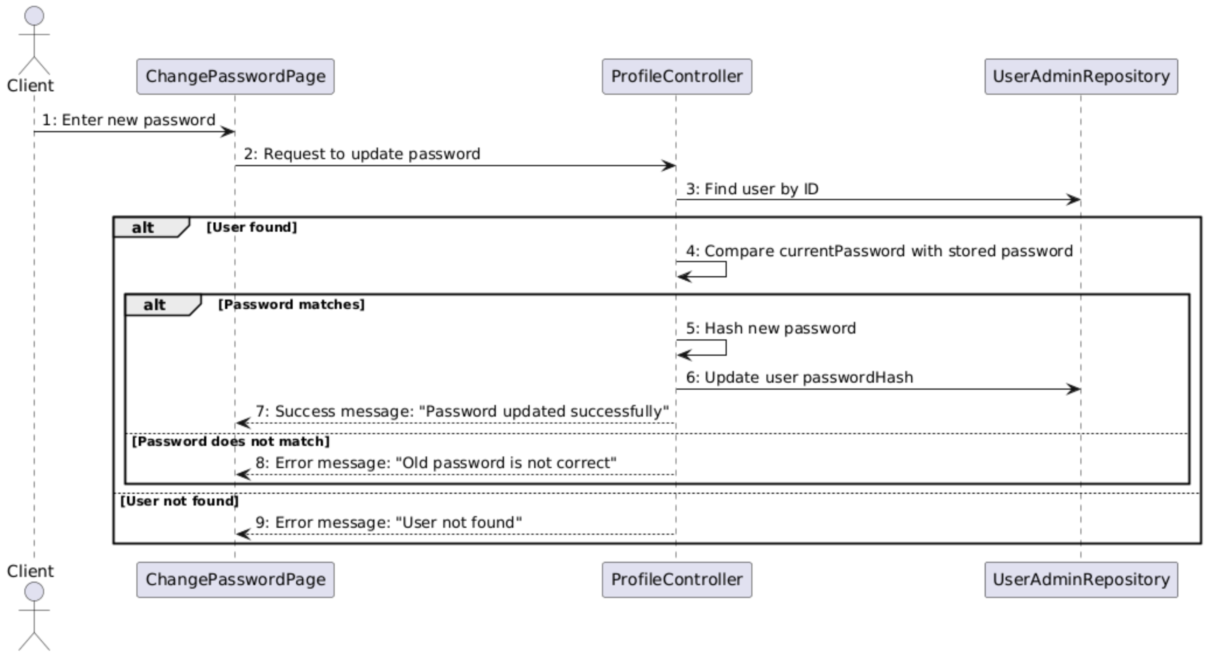


3.2.2 Change Password

a, Class Diagram

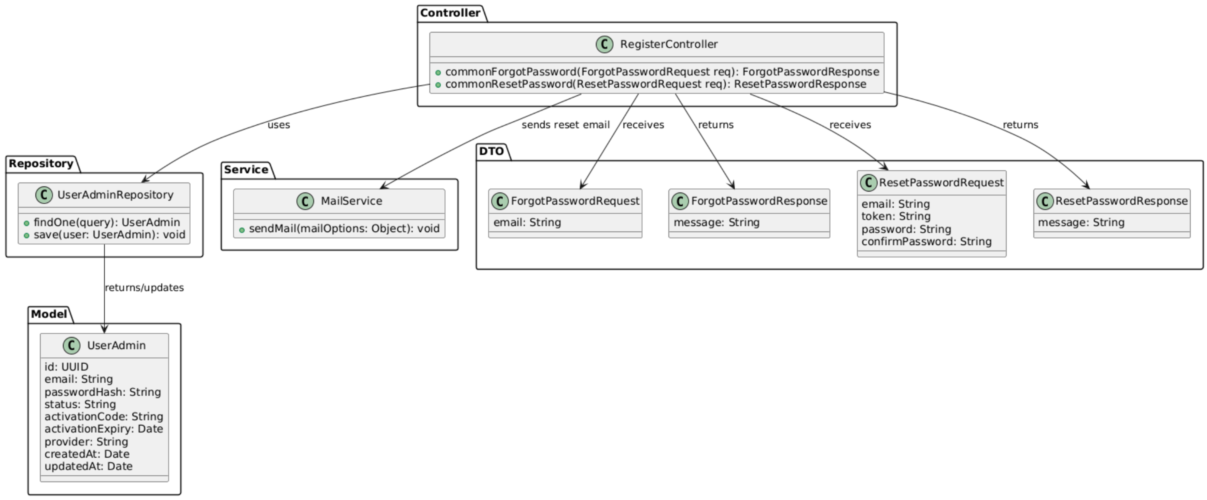


b, Sequence Diagram

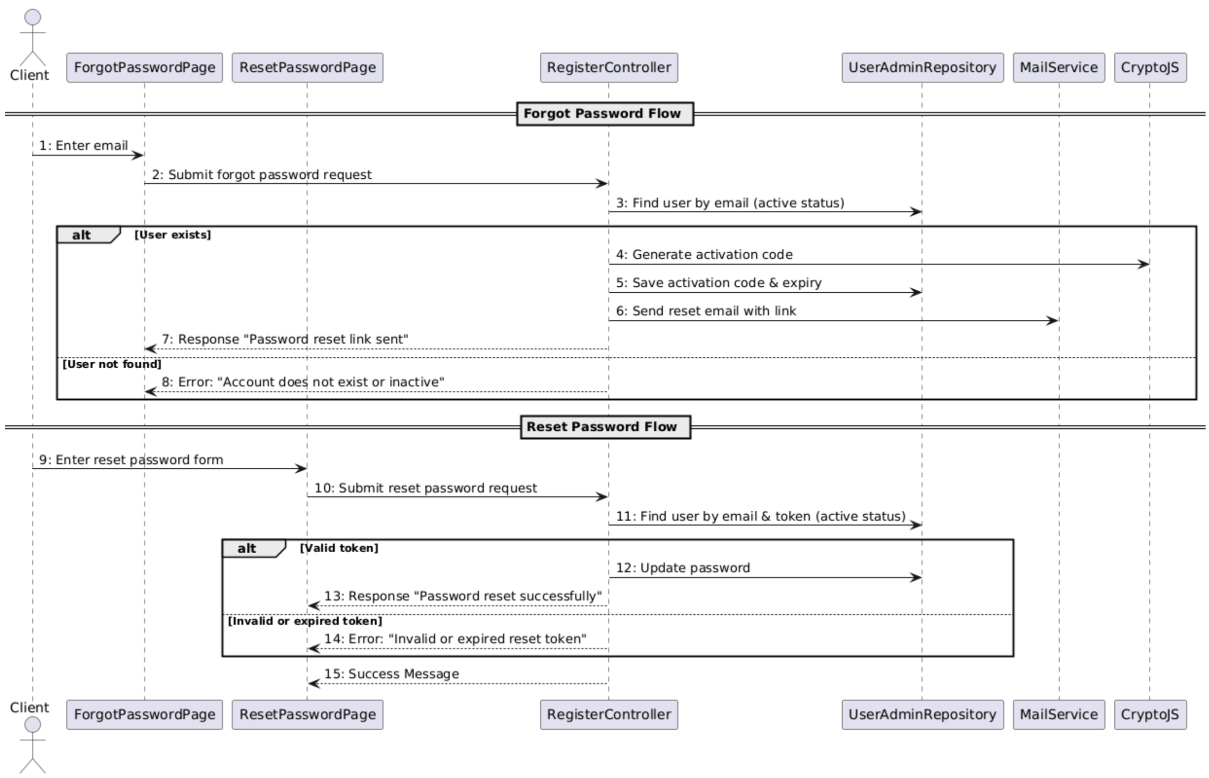


3.2.3 Reset Password

a, Class Diagram

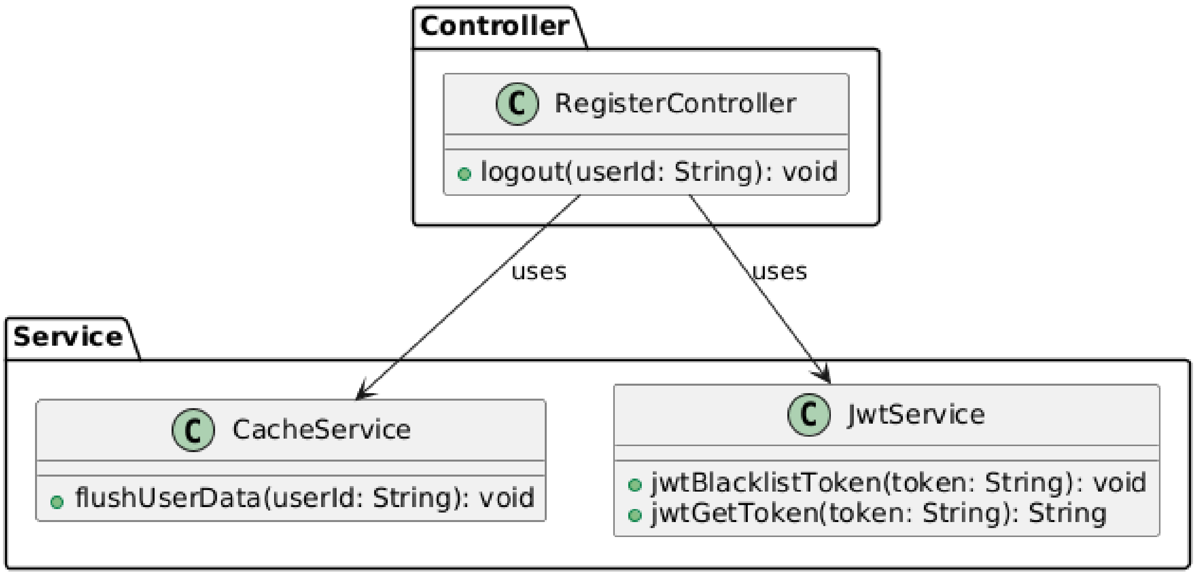


b, Sequence Diagram

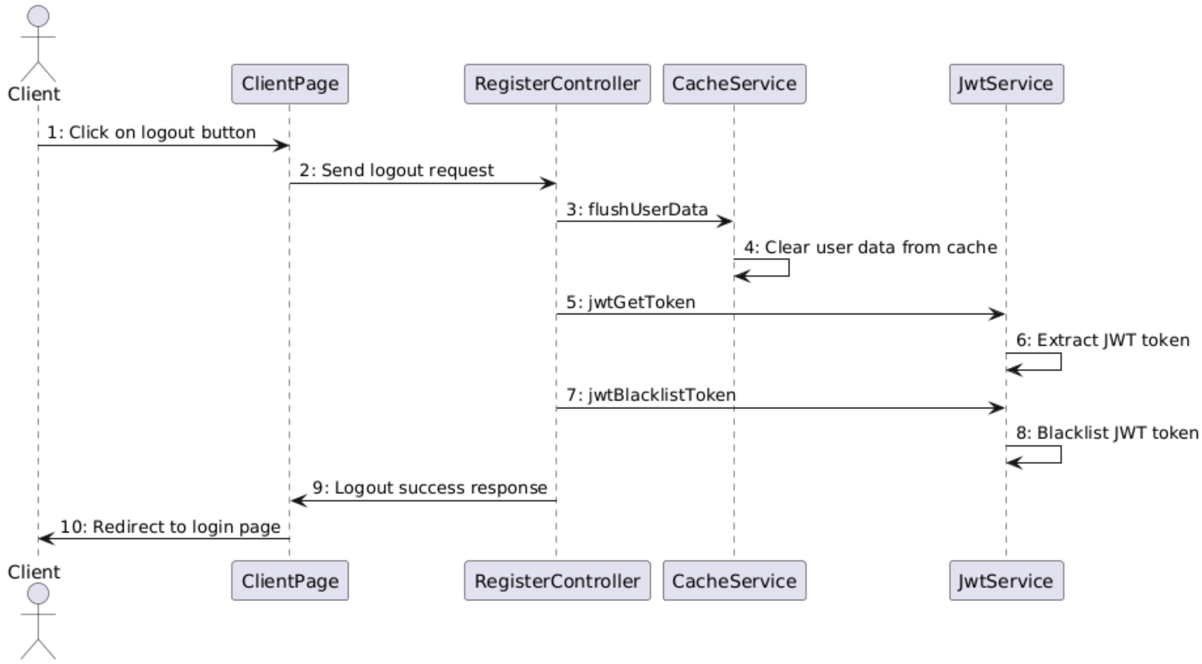


3.2.4 Logout

a, Class Diagram

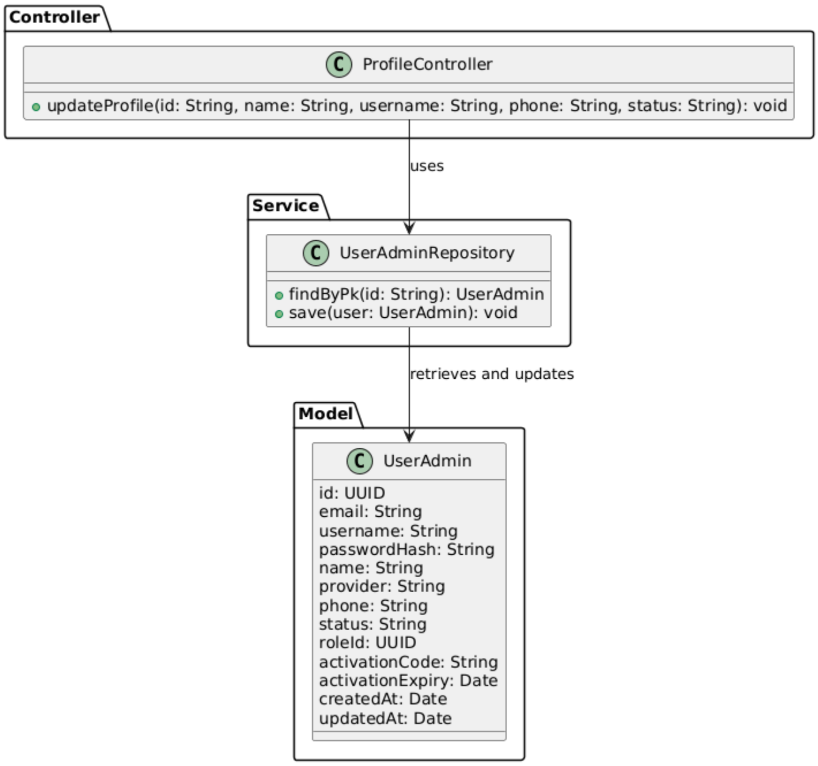


b, Sequence Diagram

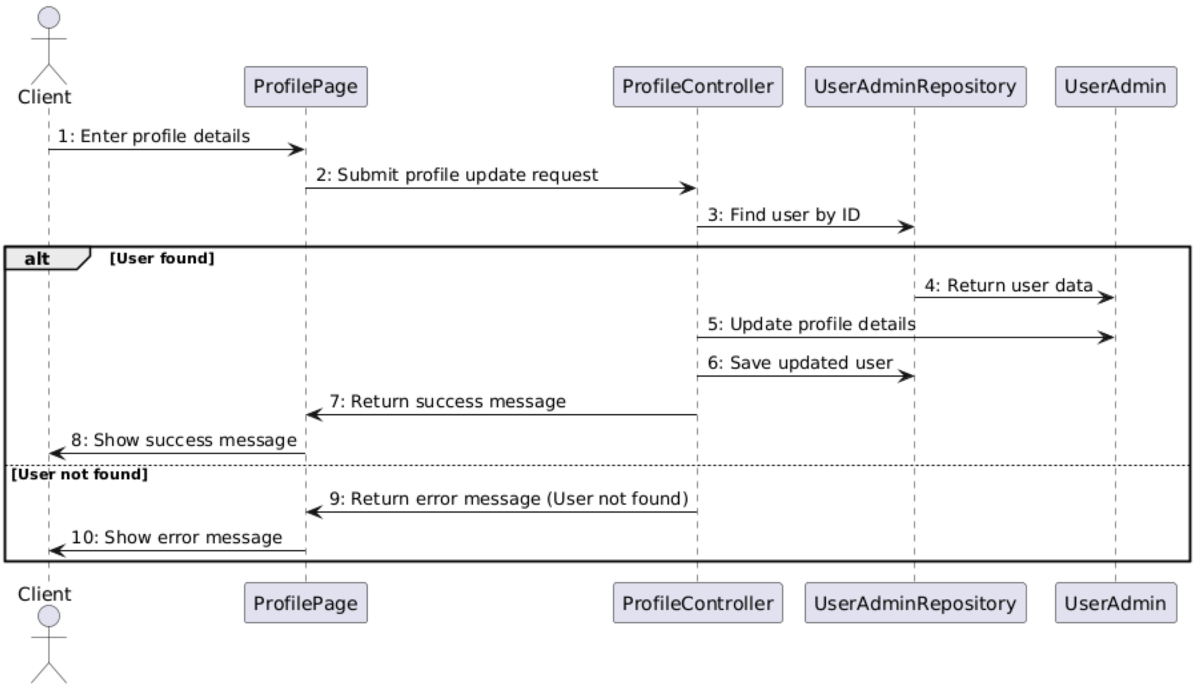


3.2.5 Edit Profile

a, Class Diagram



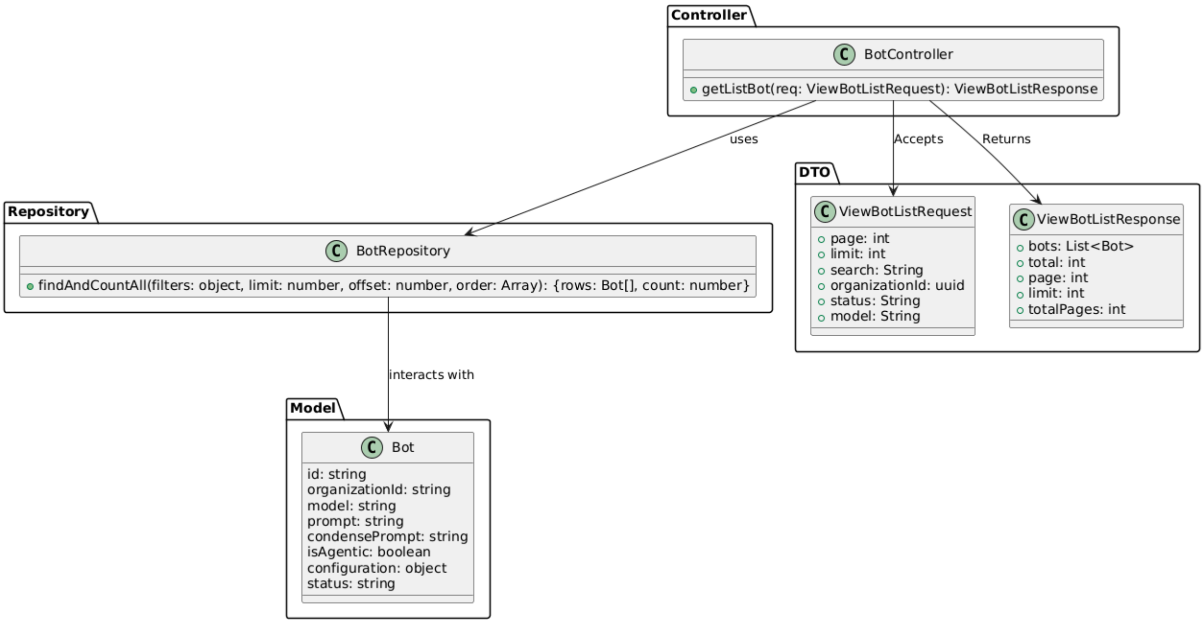
b, Sequence Diagram



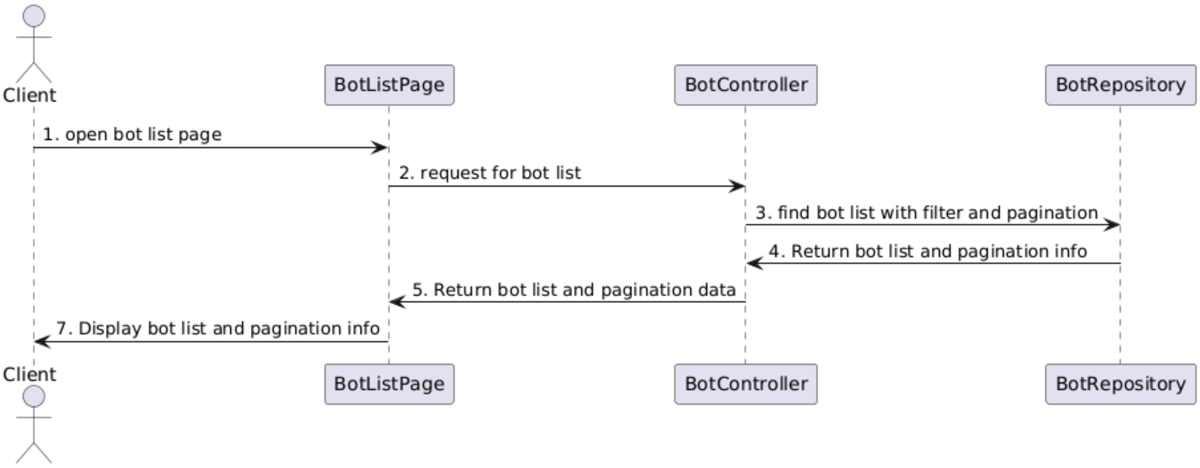
3.3 Business Staff Feature

3.3.1 View bot list

a, Class Diagram

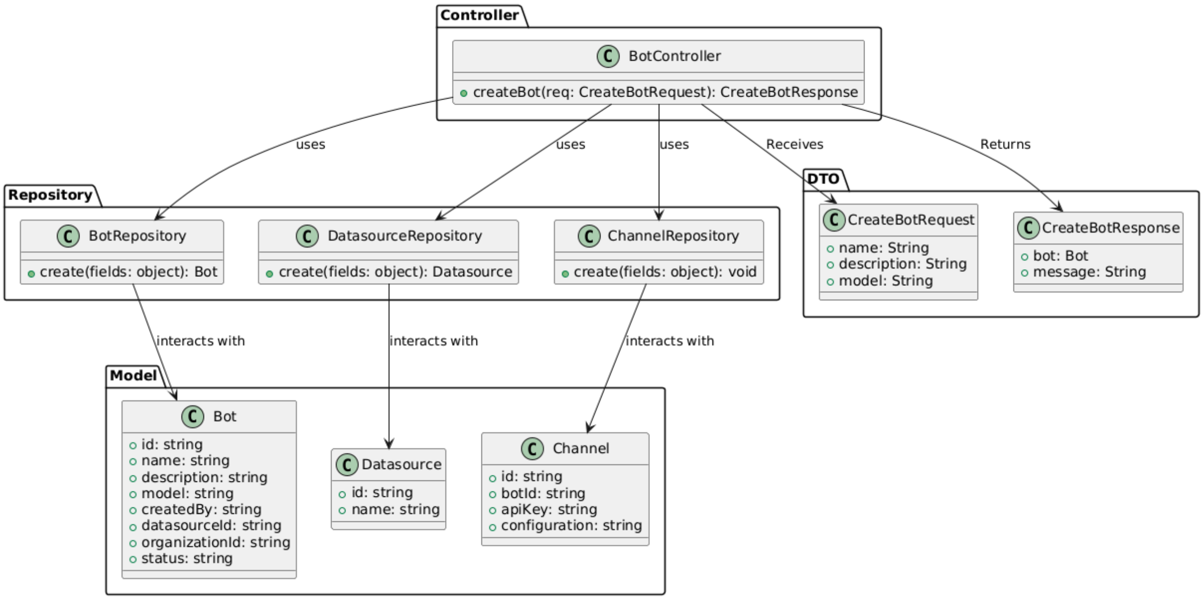


b, Sequence Diagram

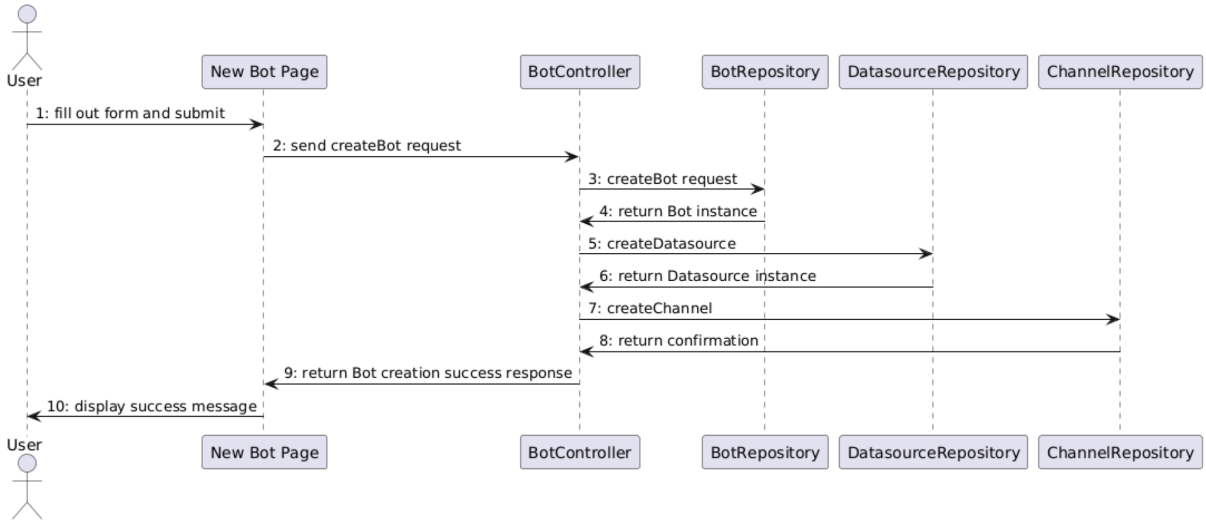


3.3.2 Add New Bot

a, Class Diagram

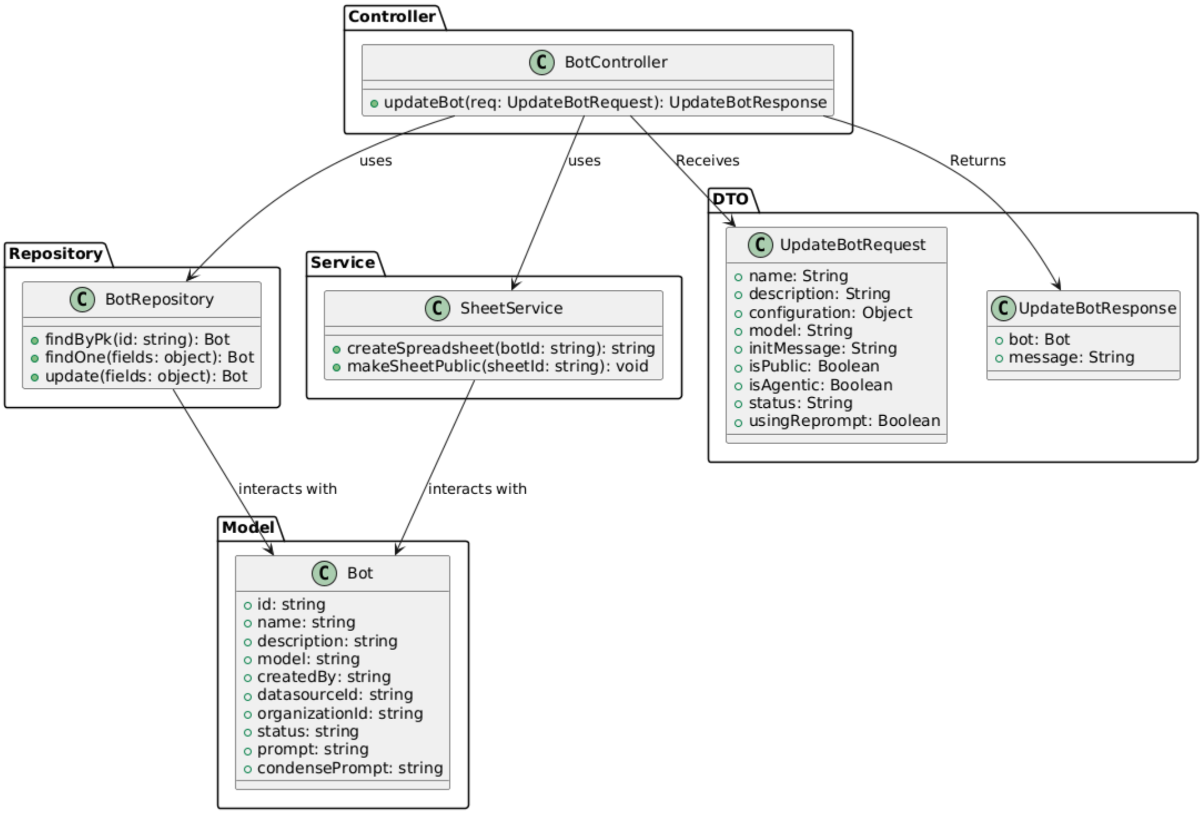


b, Sequence Diagram

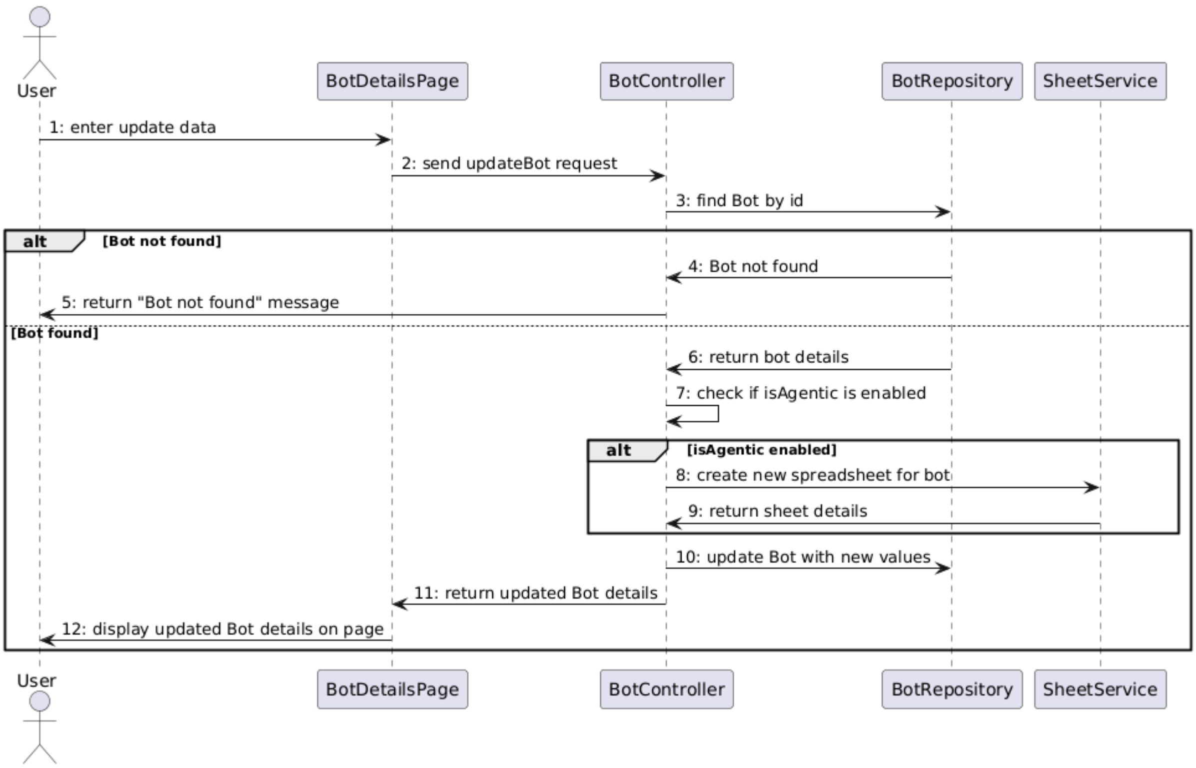


3.3.3 Update Bot Details

a, Class Diagram

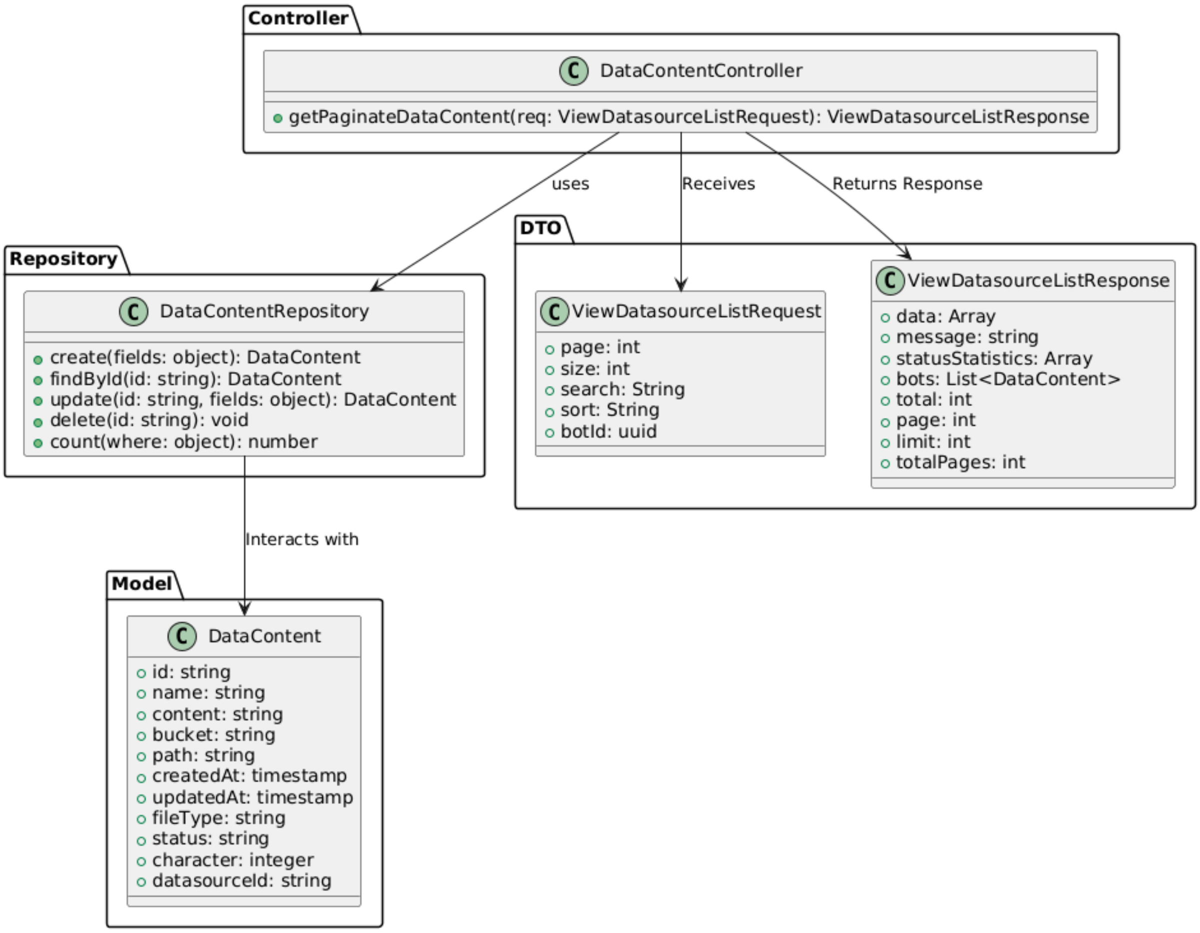


b, Sequence Diagram

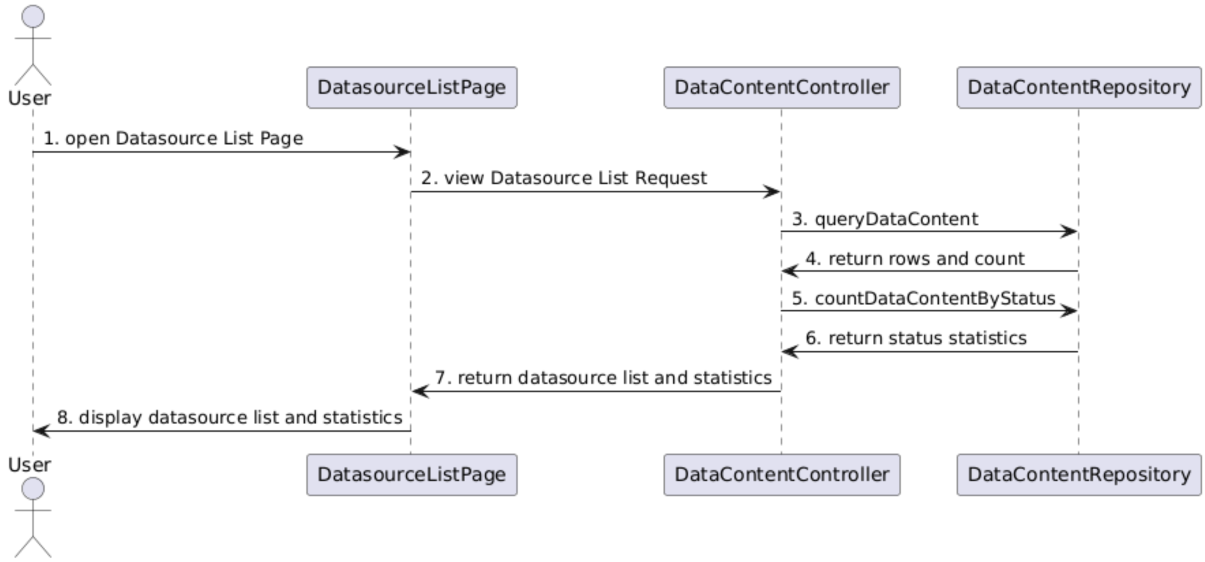


3.3.4 View Data Source List

a, Class Diagram

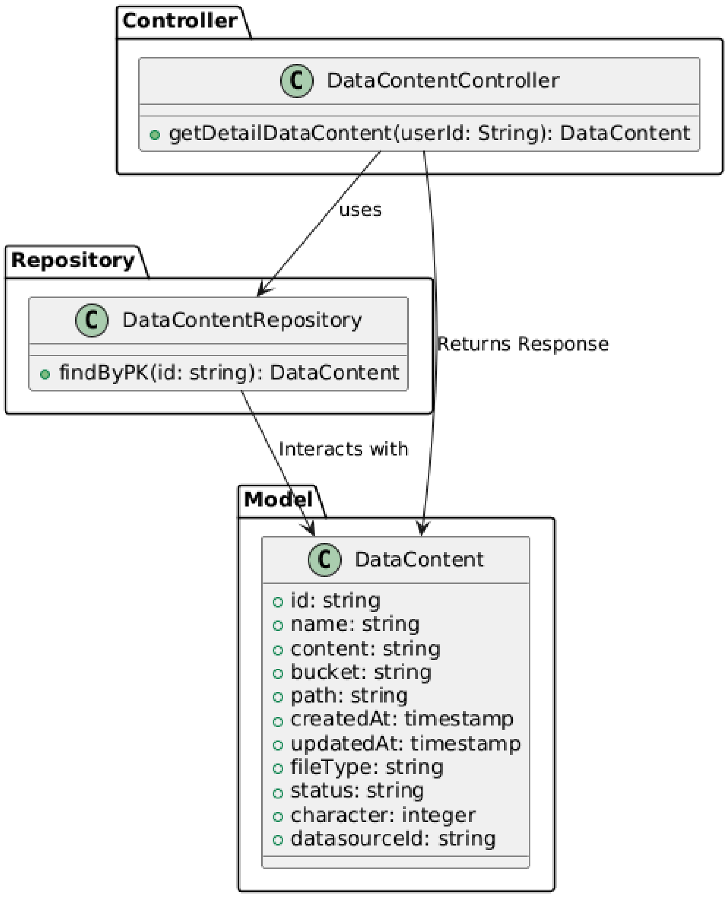


b, Sequence Diagram

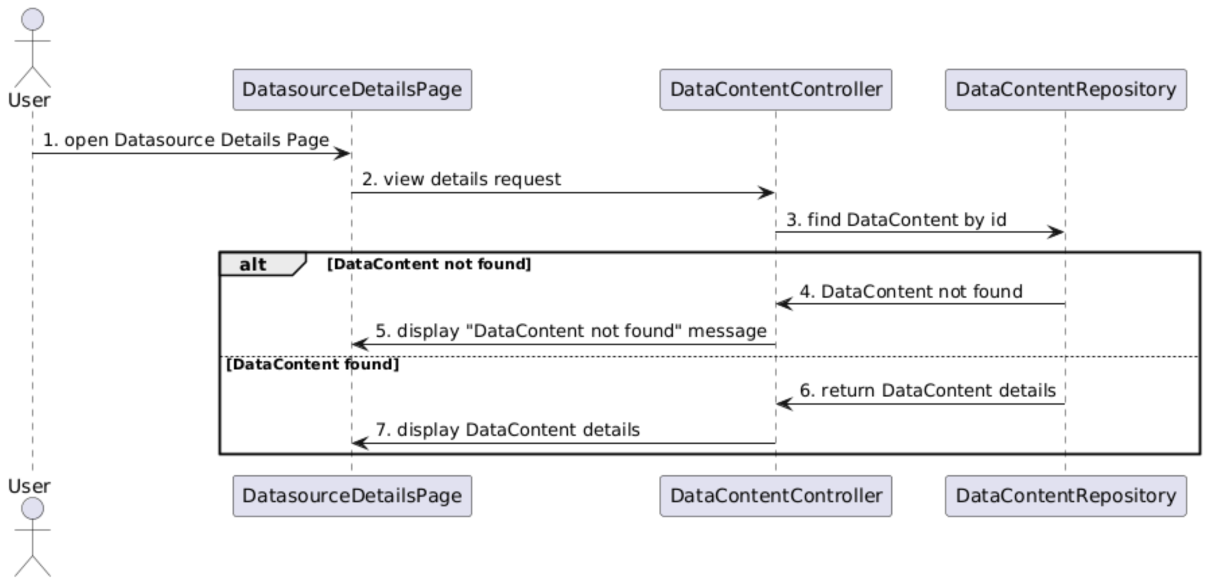


3.3.5 View Data Source Details

a, Class Diagram

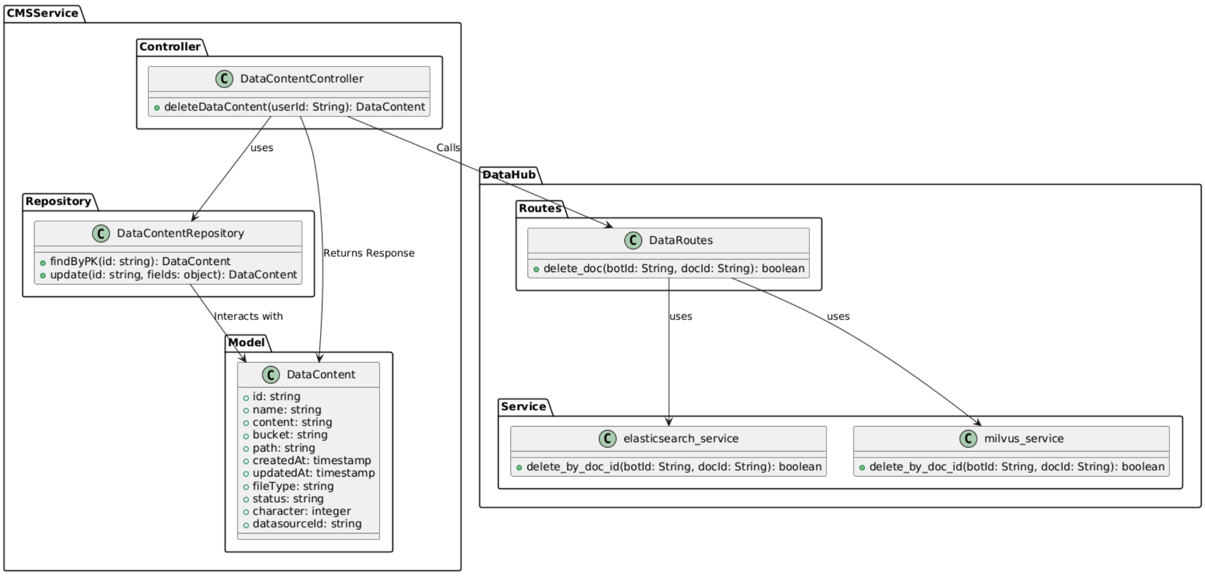


b, Sequence Diagram

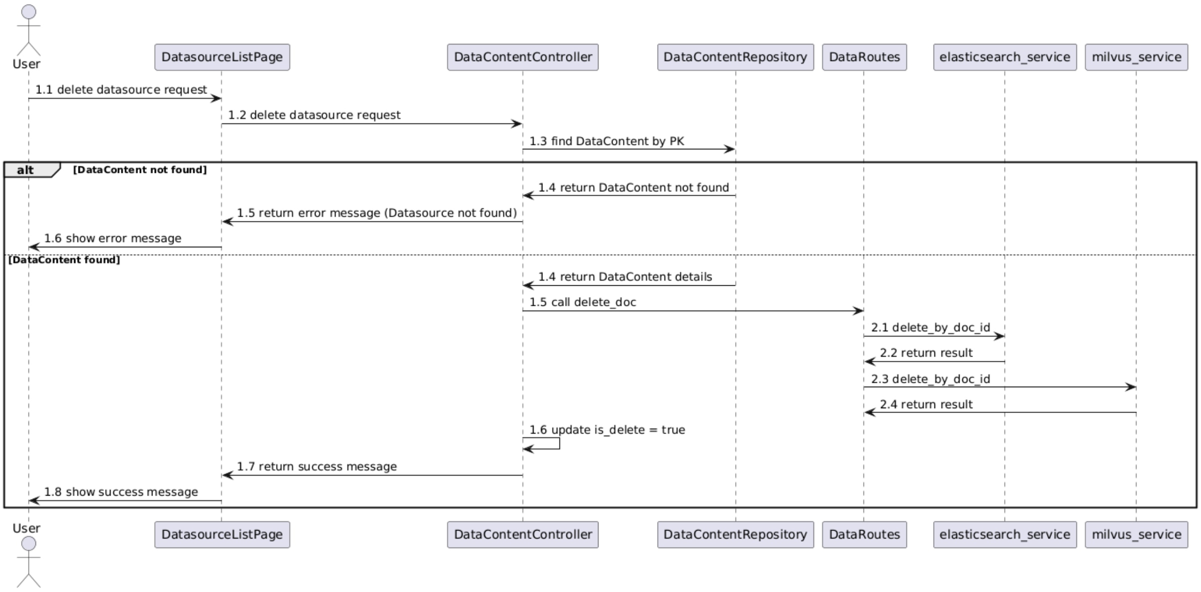


3.3.6 Delete Data Source

a, Class Diagram

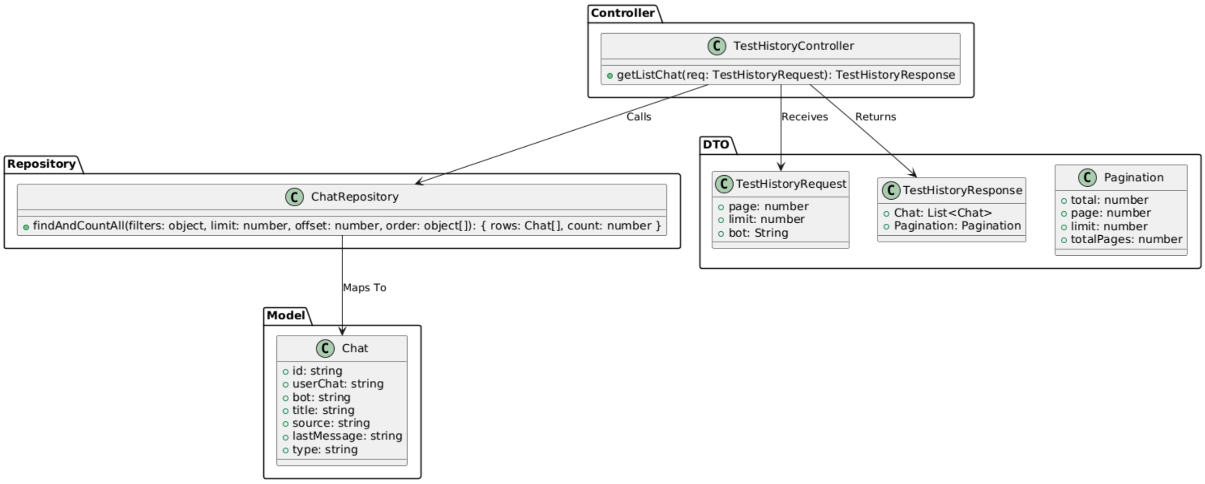


b, Sequence Diagram

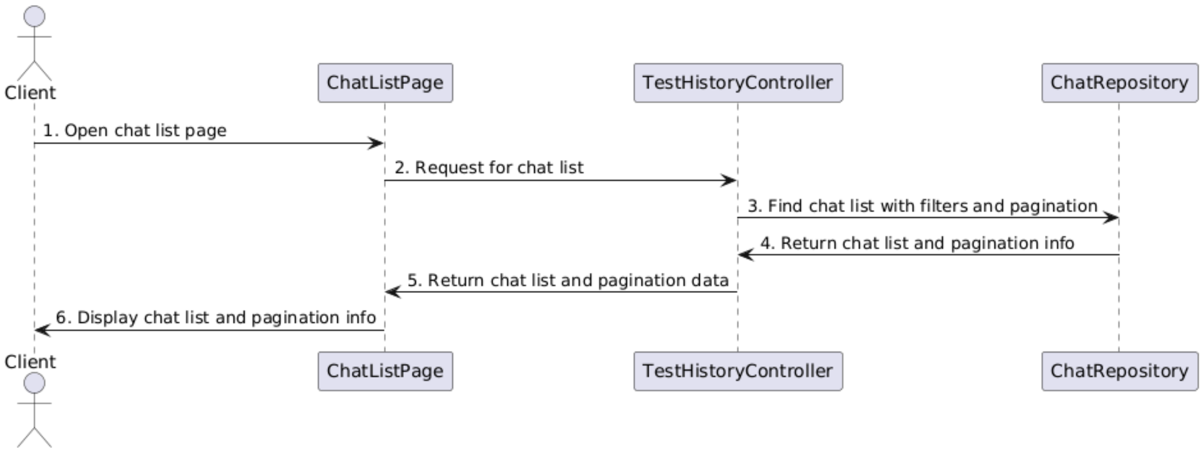


3.3.7 View Test History

a, Class Diagram



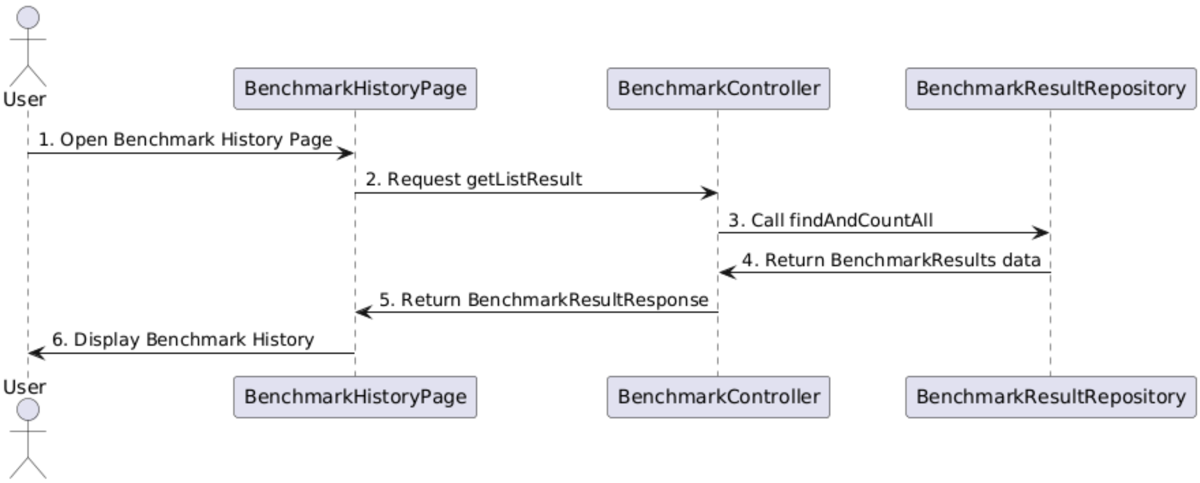
b, Sequence Diagram



3.3.8 View Benchmark History

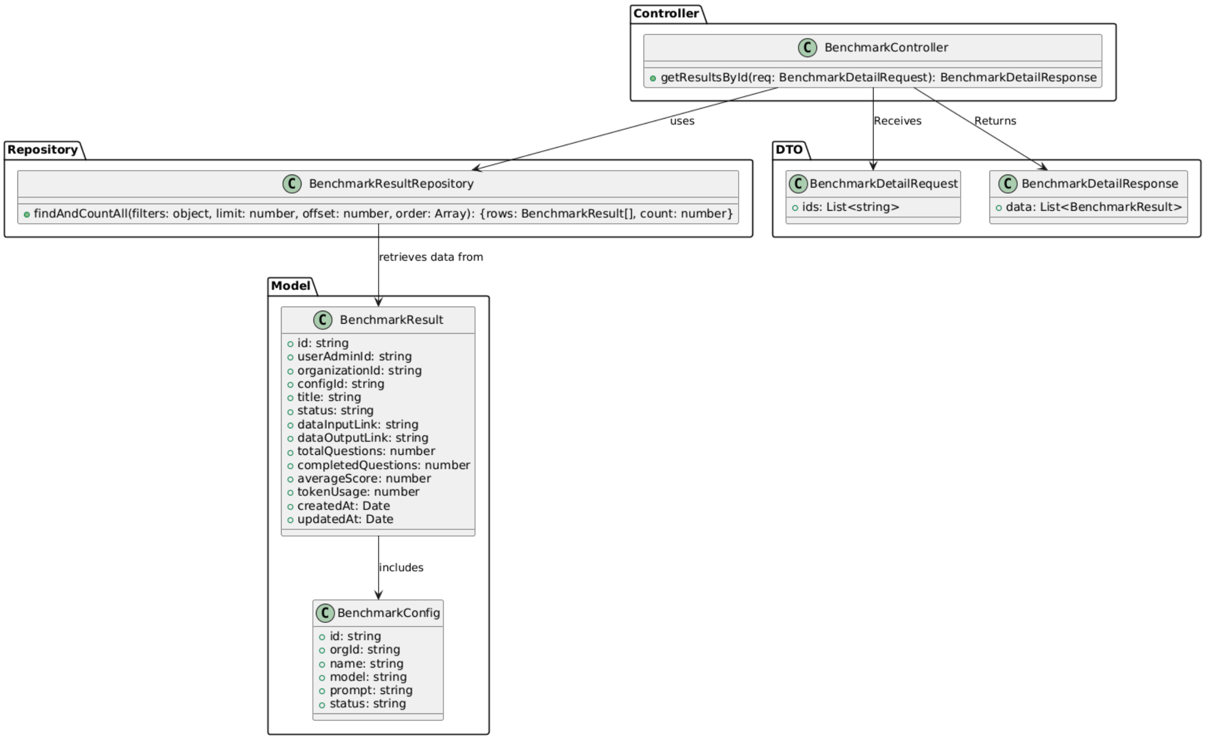
a, Class Diagram

b, Sequence Diagram

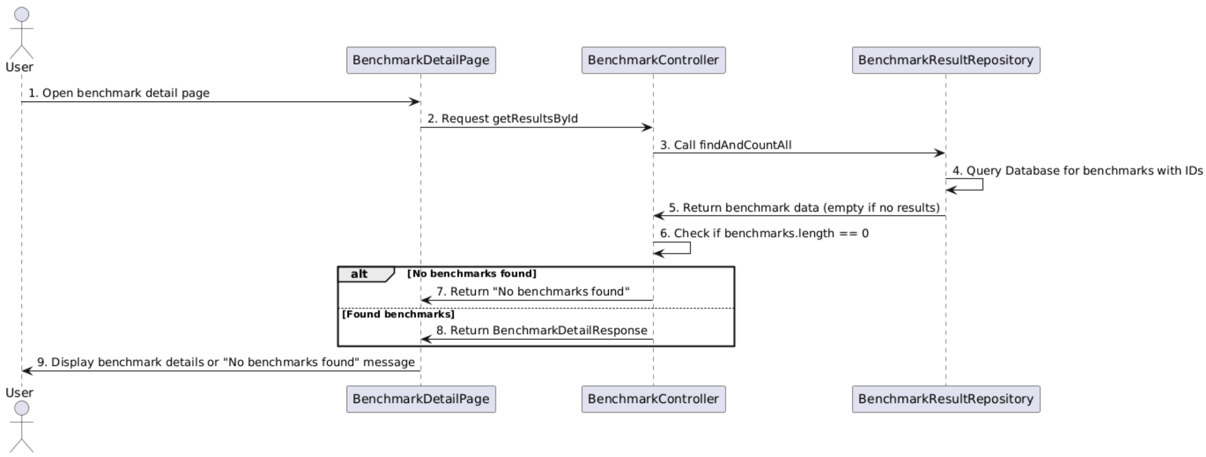


3.3.9 View Benchmark Details

a, Class Diagram



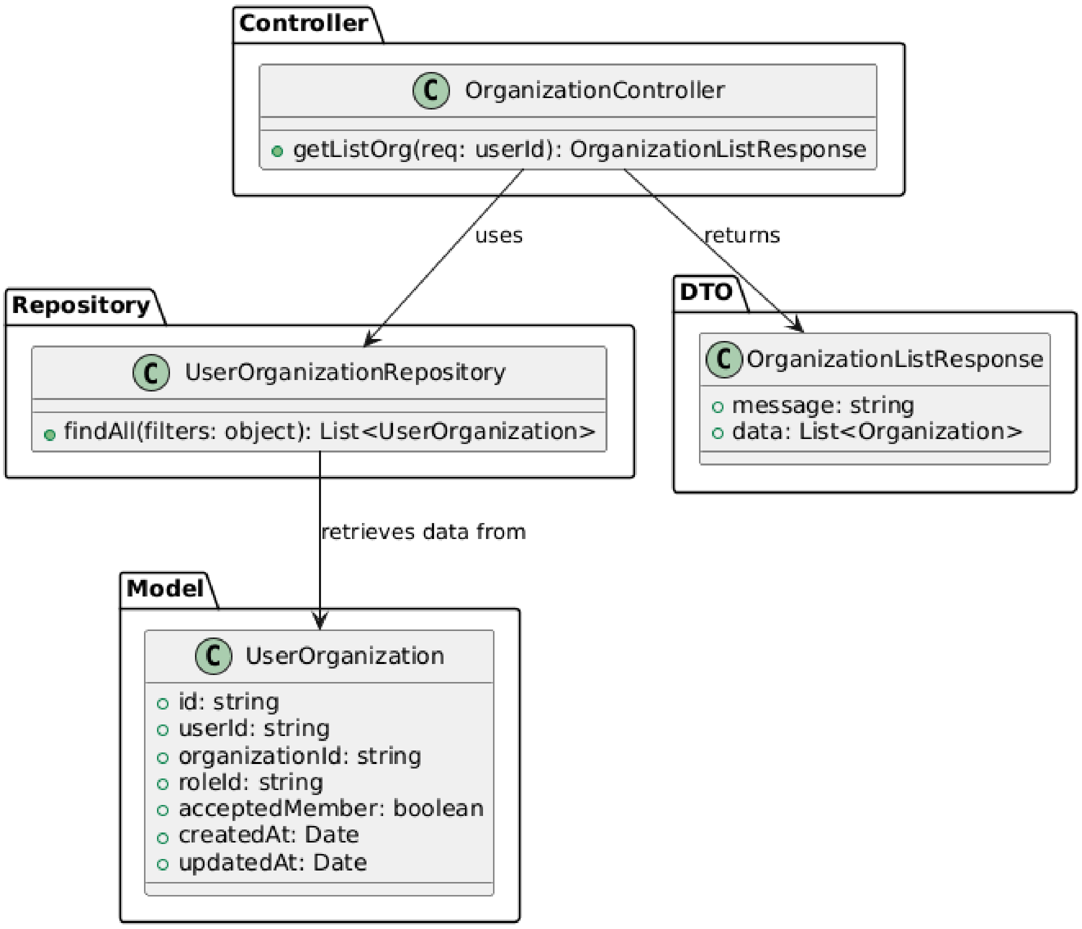
b, Sequence Diagram



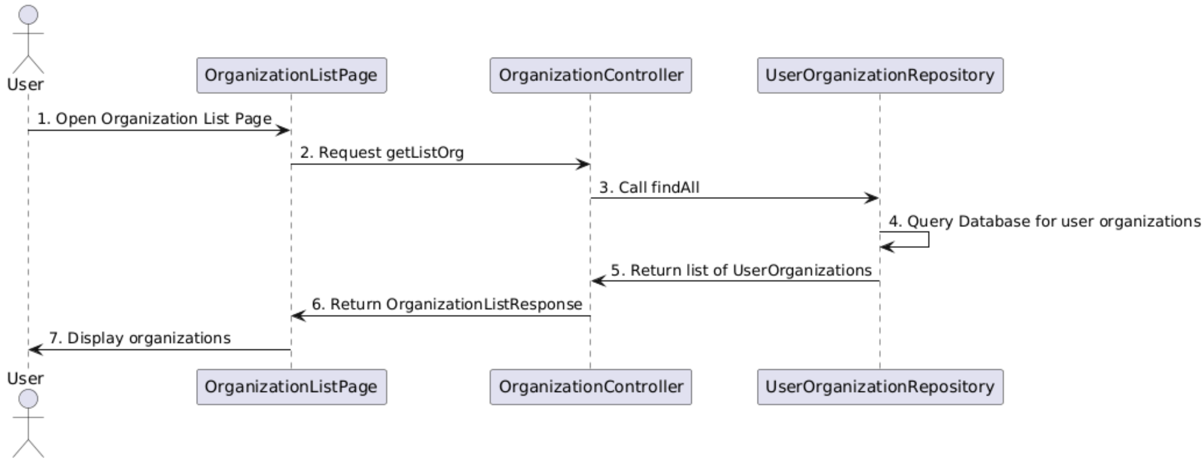
3.4 Business Owner Feature

3.4.1 View Organization List

a, Class Diagram

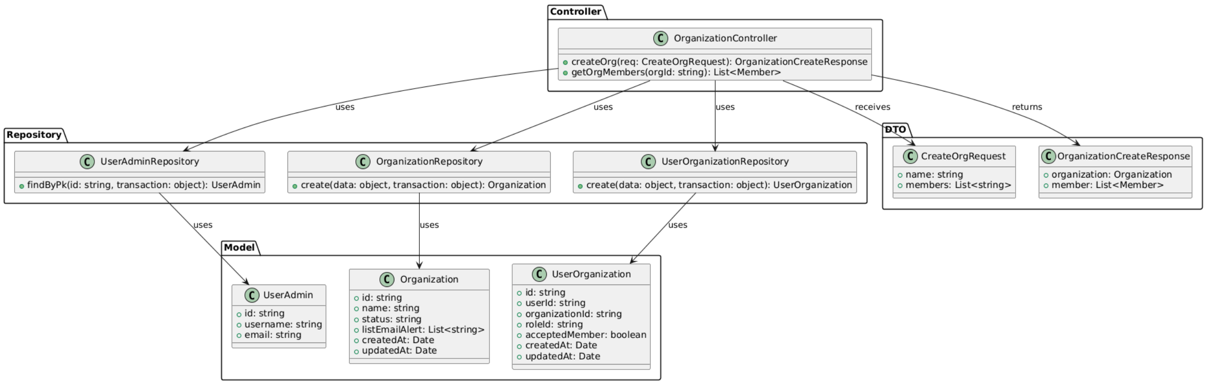


b, Sequence Diagram

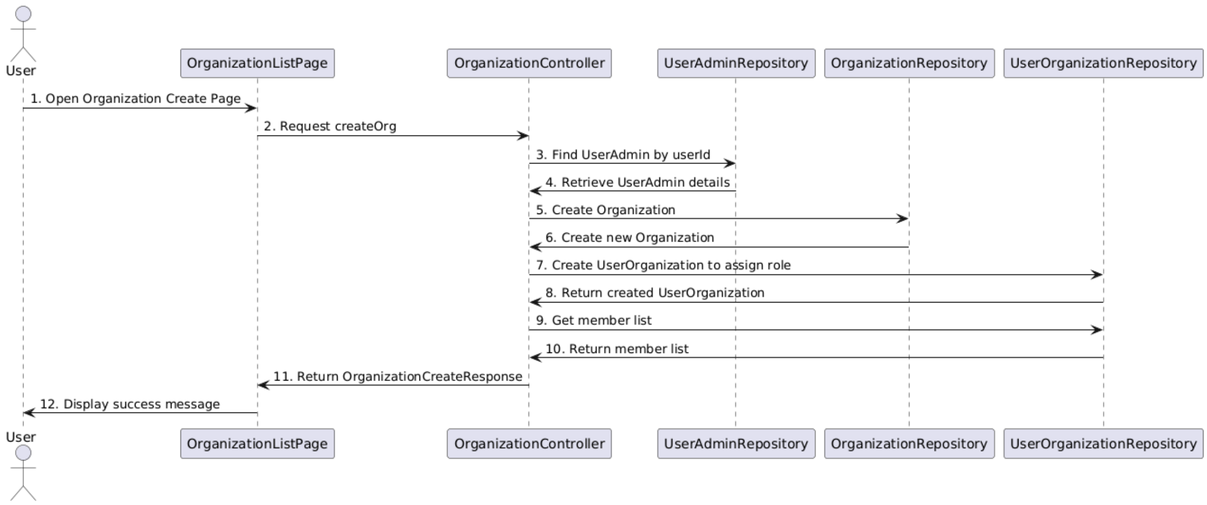


3.4.2 Add New Organization

a, Class Diagram

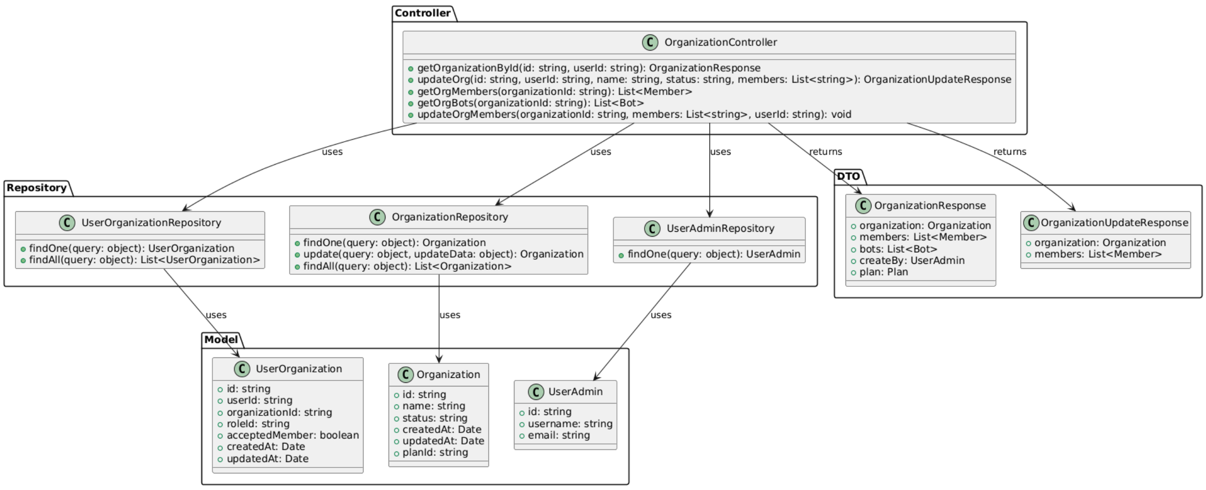


b, Sequence Diagram

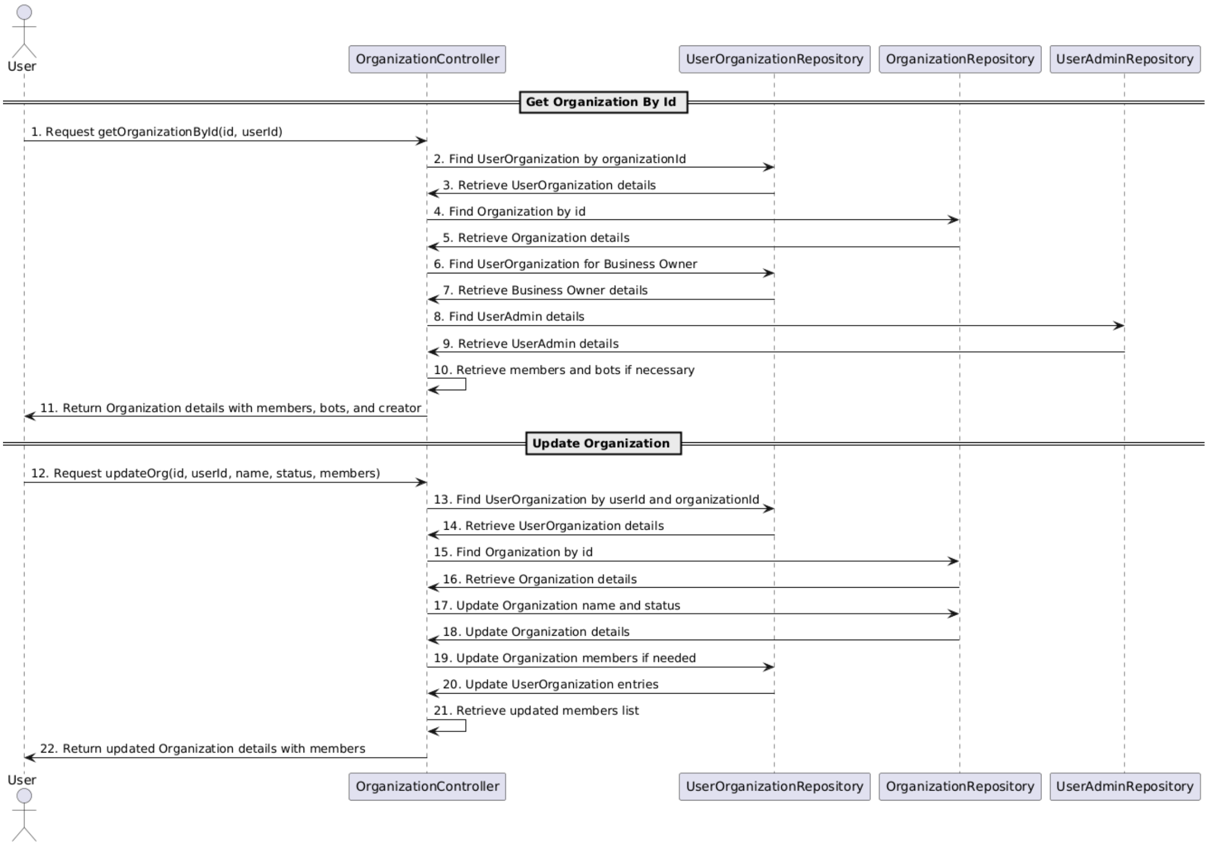


3.4.3 Update Organization Details

a, Class Diagram

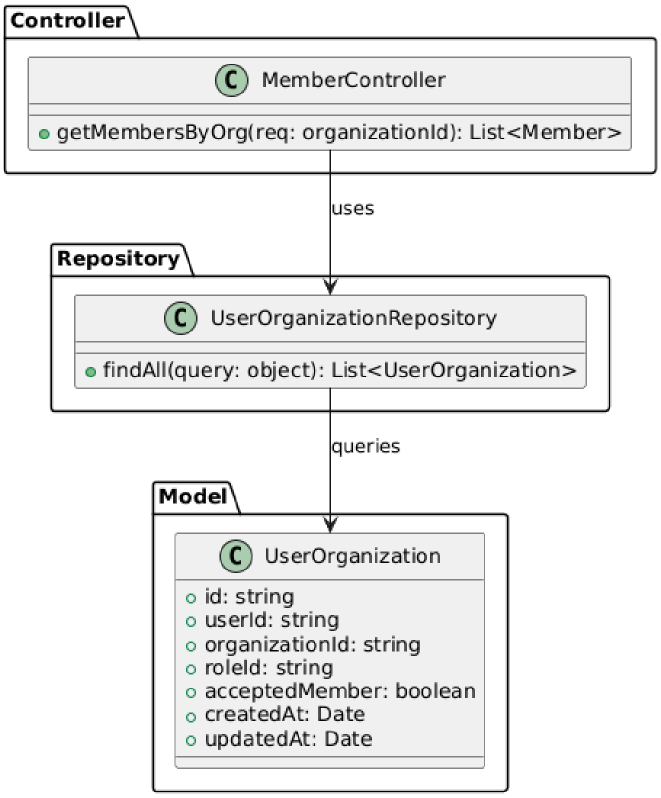


b, Sequence Diagram

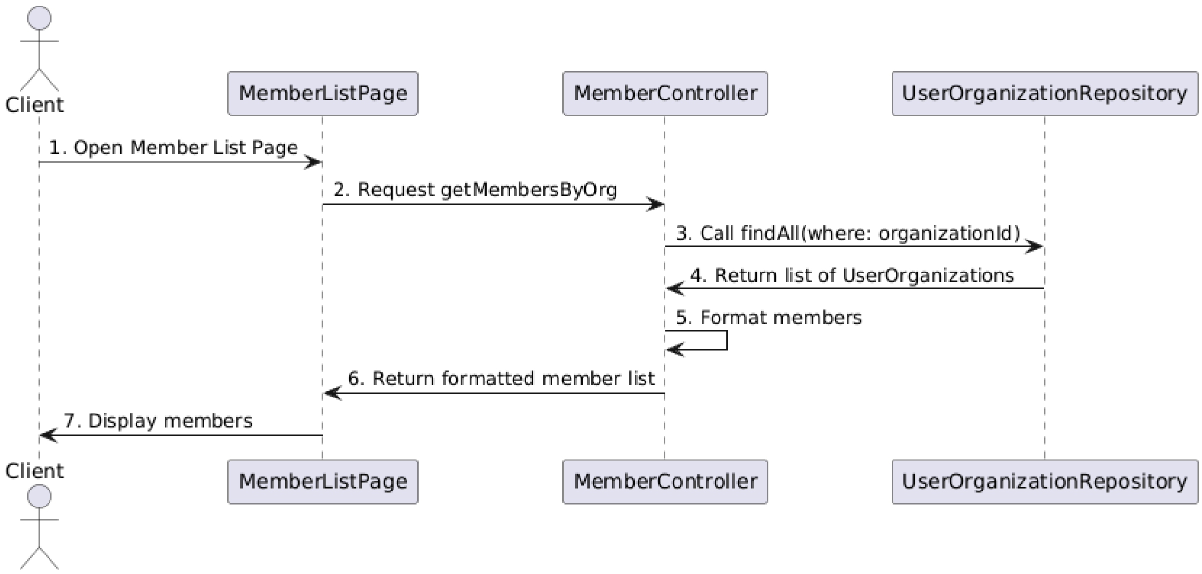


3.4.4 View Member List

a, Class Diagram

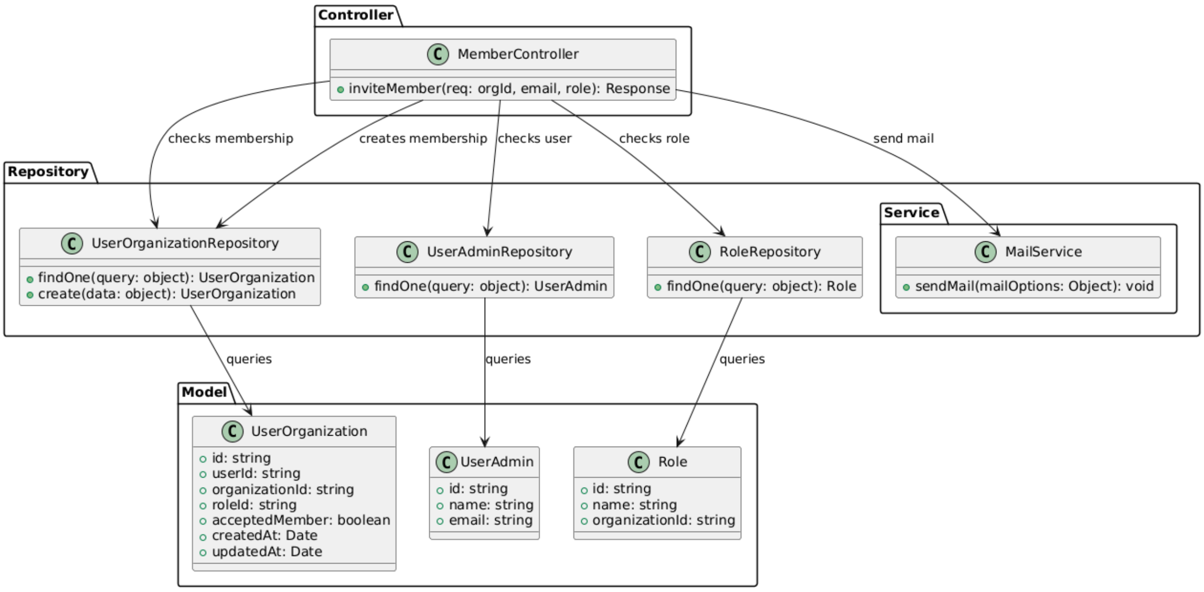


b, Sequence Diagram

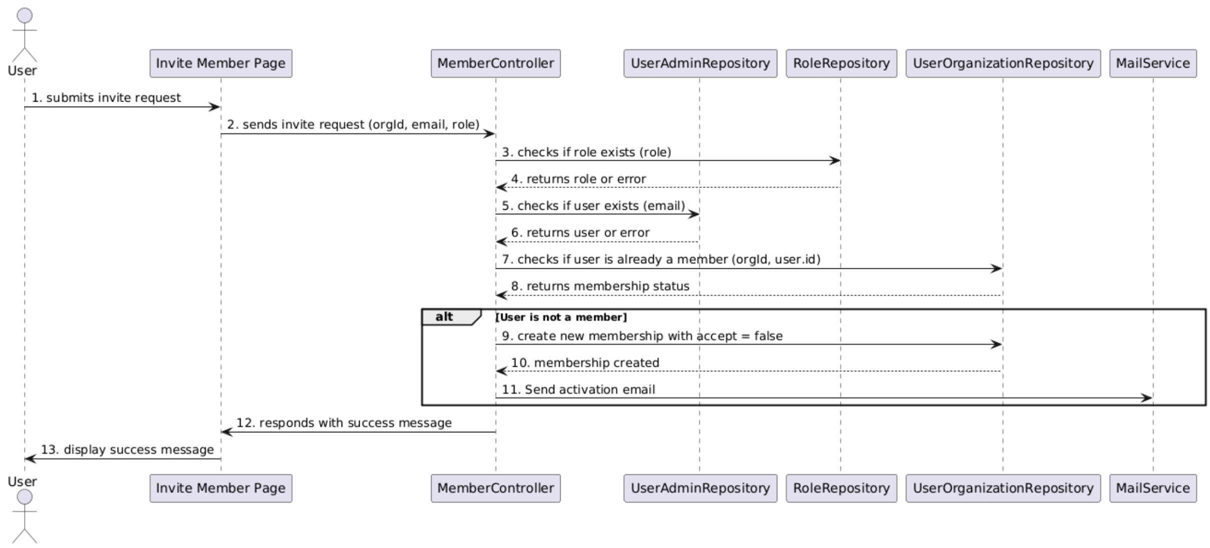


3.4.5 Add New Member

a, Class Diagram

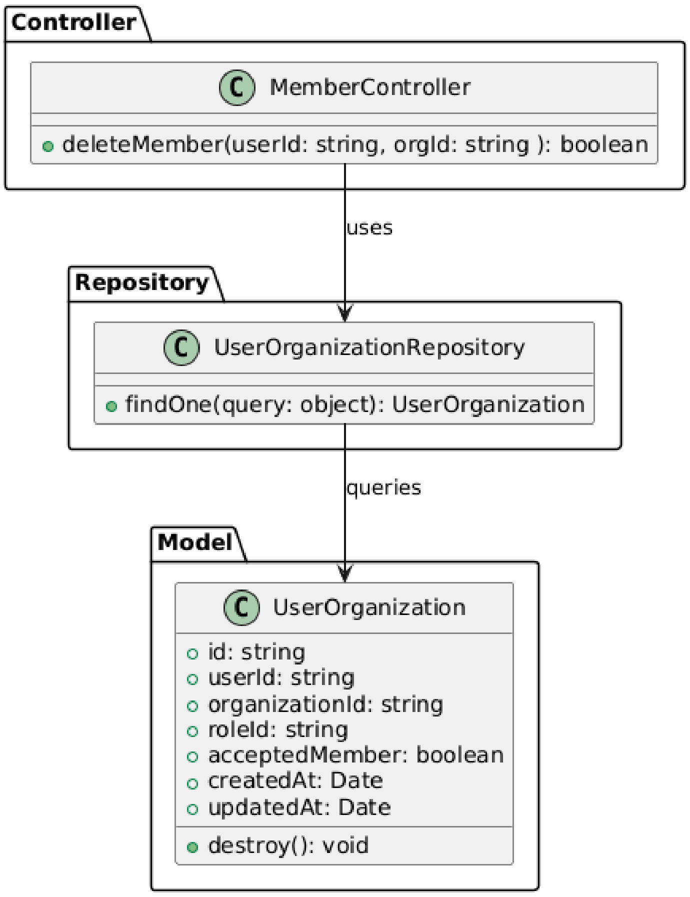


b, Sequence Diagram

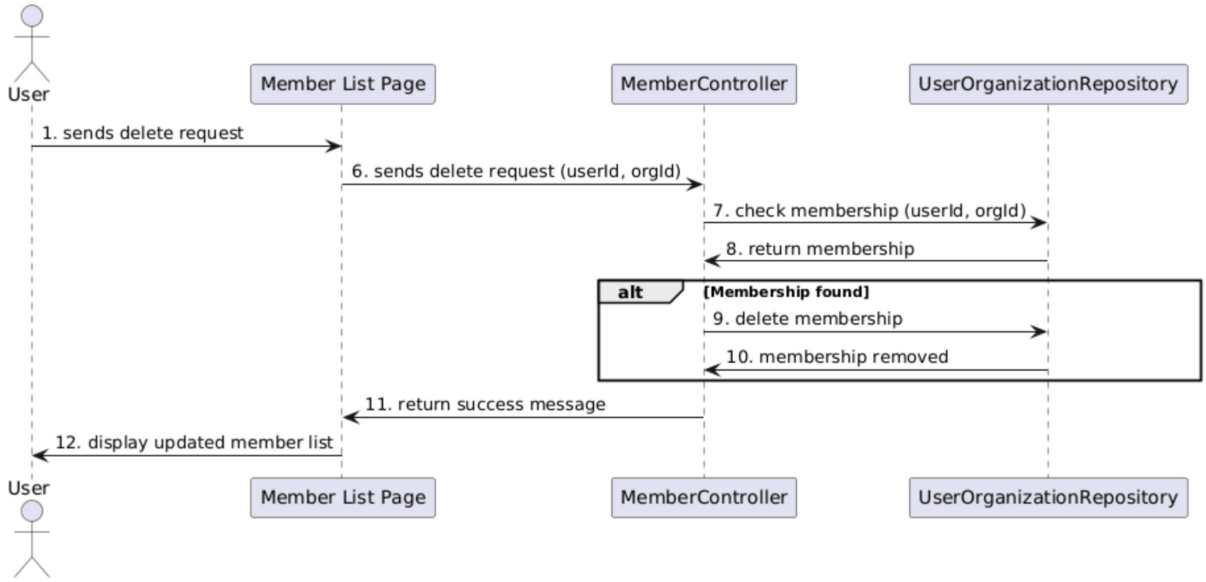


3.4.6 Delete Member

a, Class Diagram

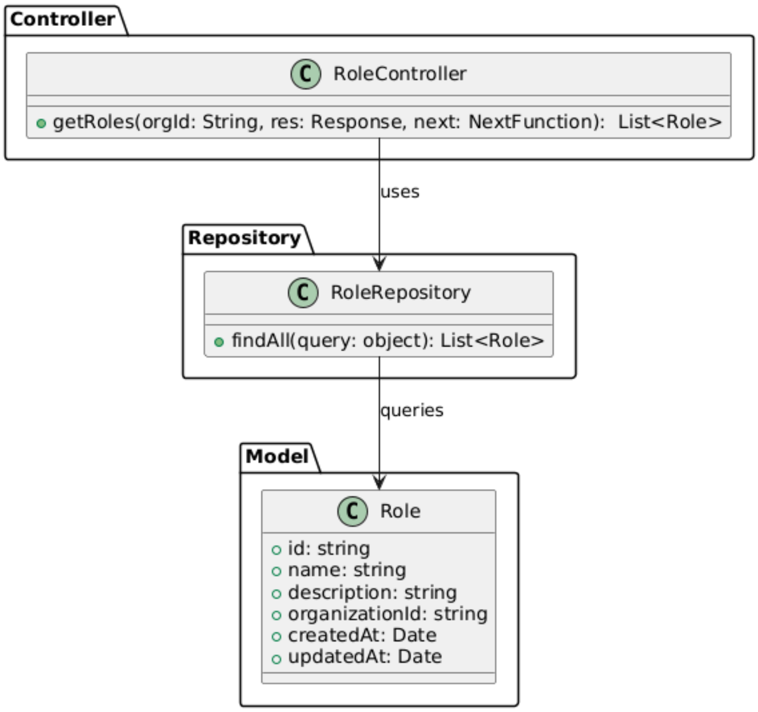


b, Sequence Diagram

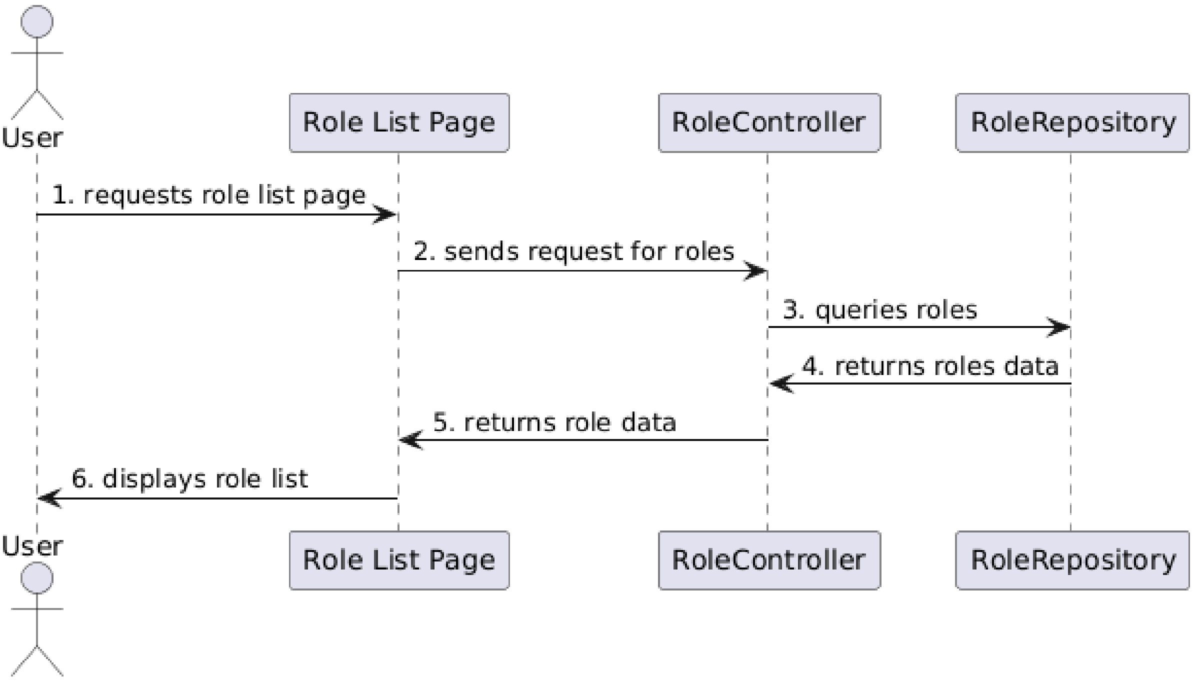


3.4.7 View Role List

a, Class Diagram

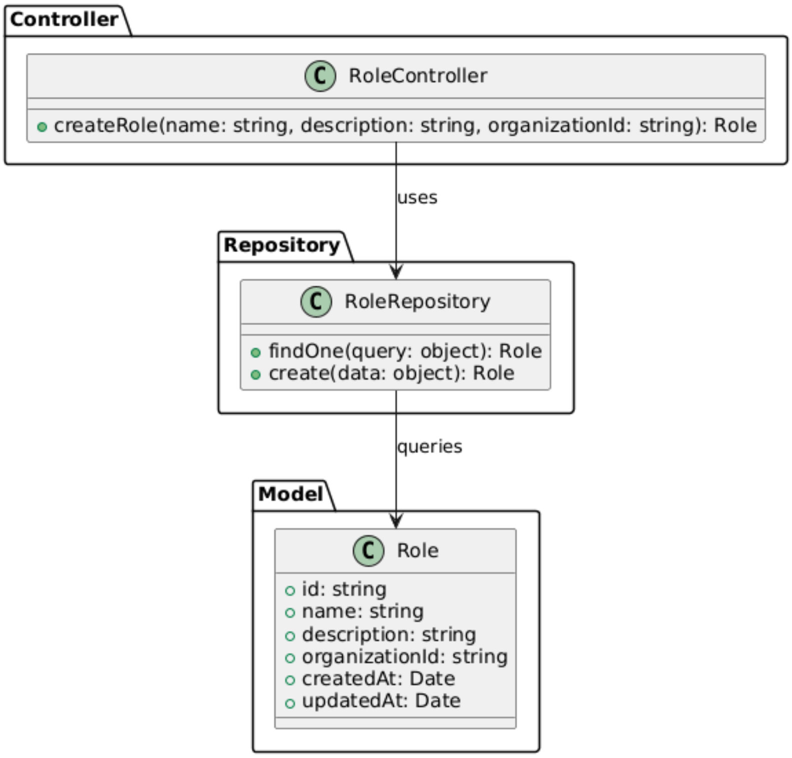


b, Sequence Diagram

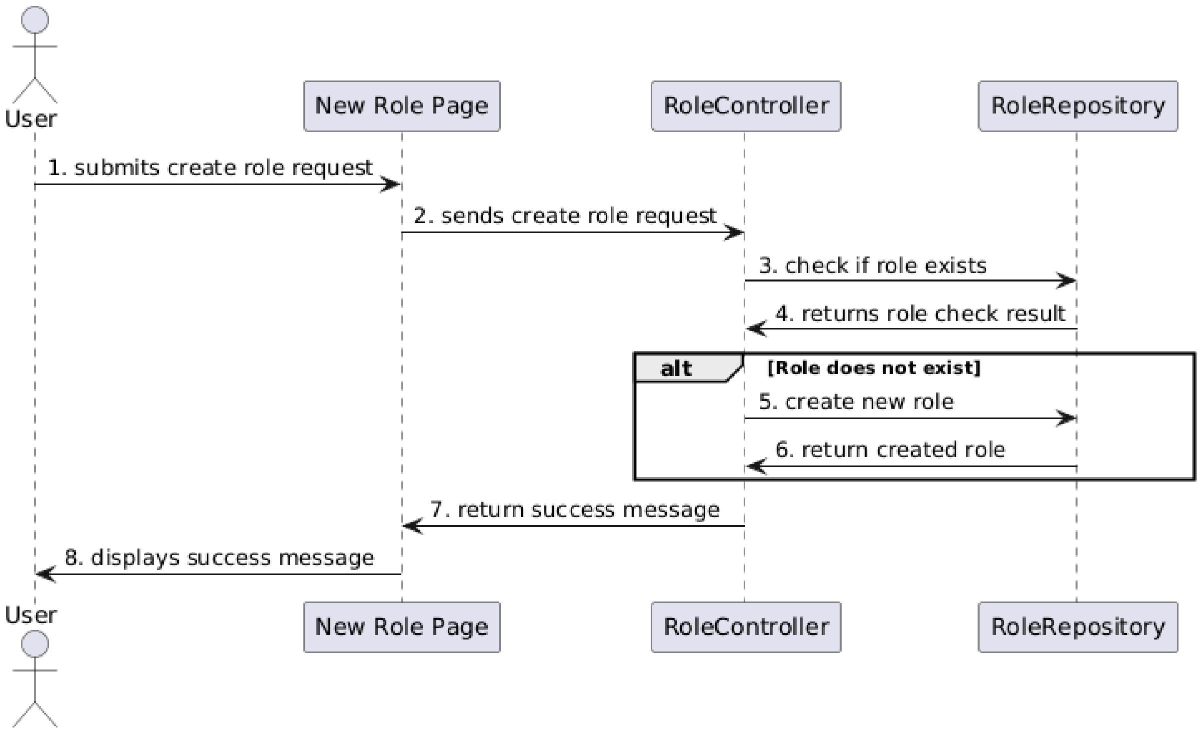


3.4.8 Add New Role

a, Class Diagram

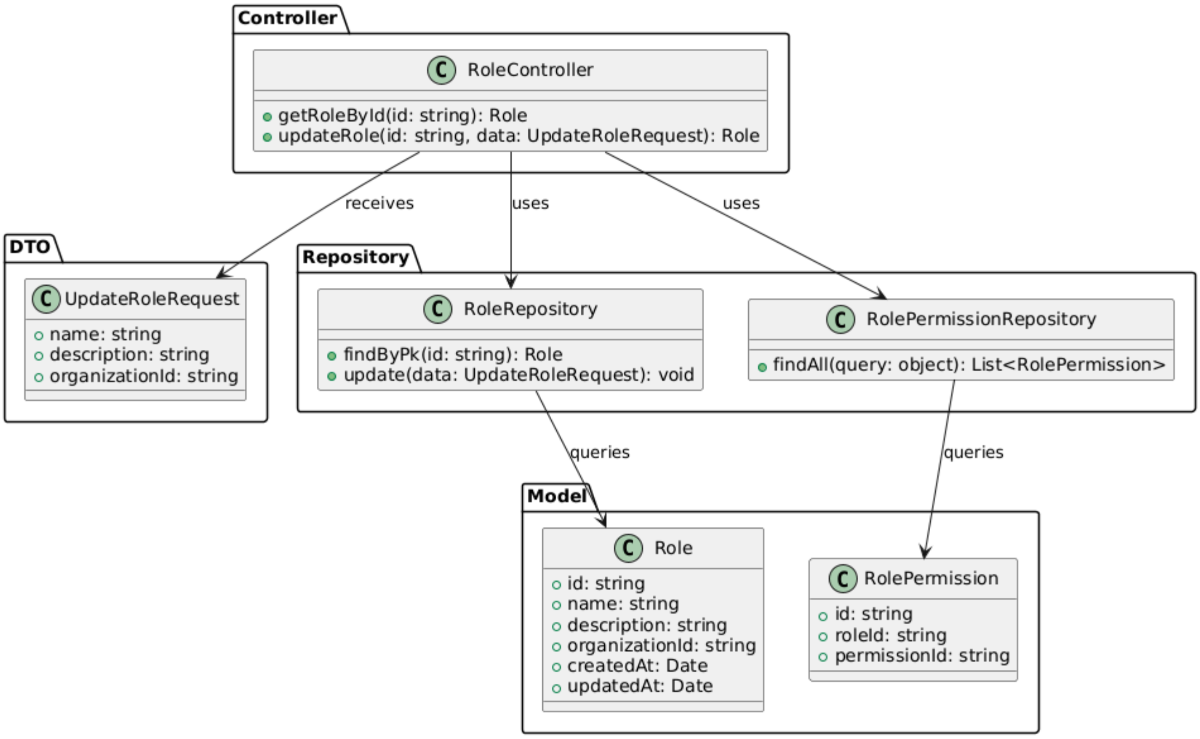


b, Sequence Diagram

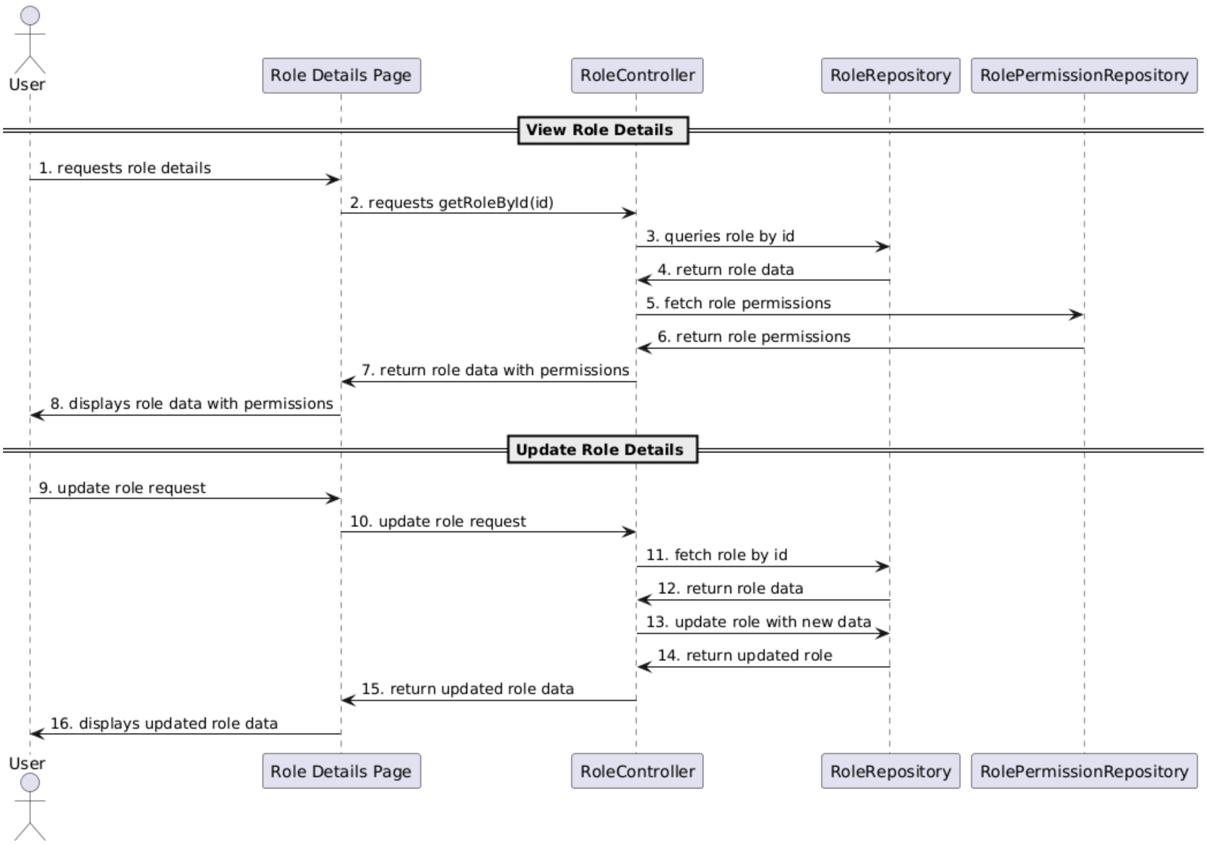


3.4.9 Update Role Details

a, Class Diagram

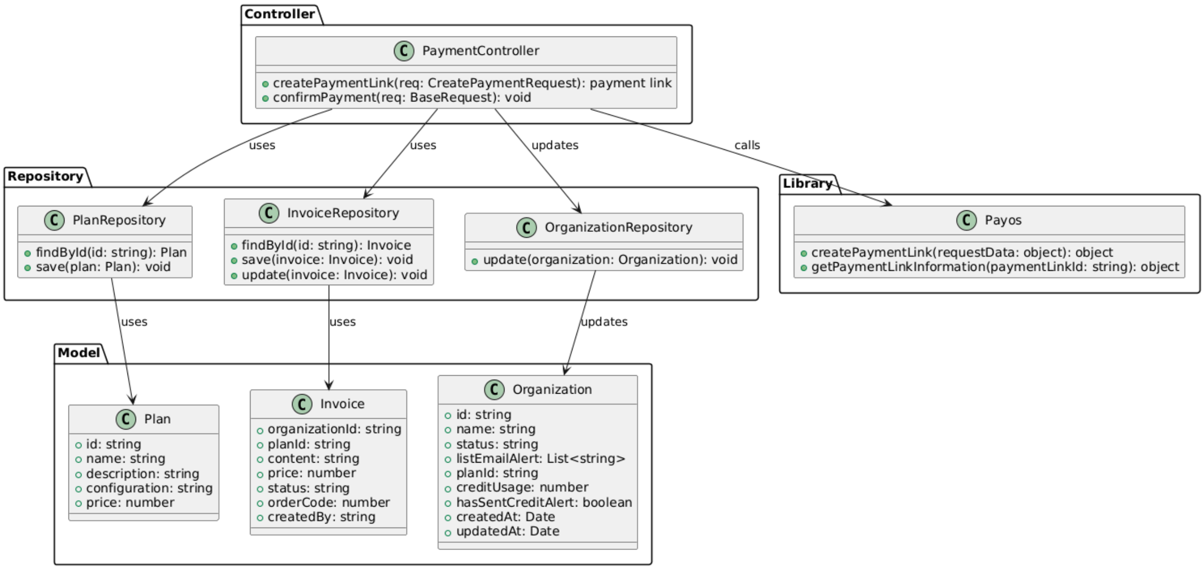


b, Sequence Diagram

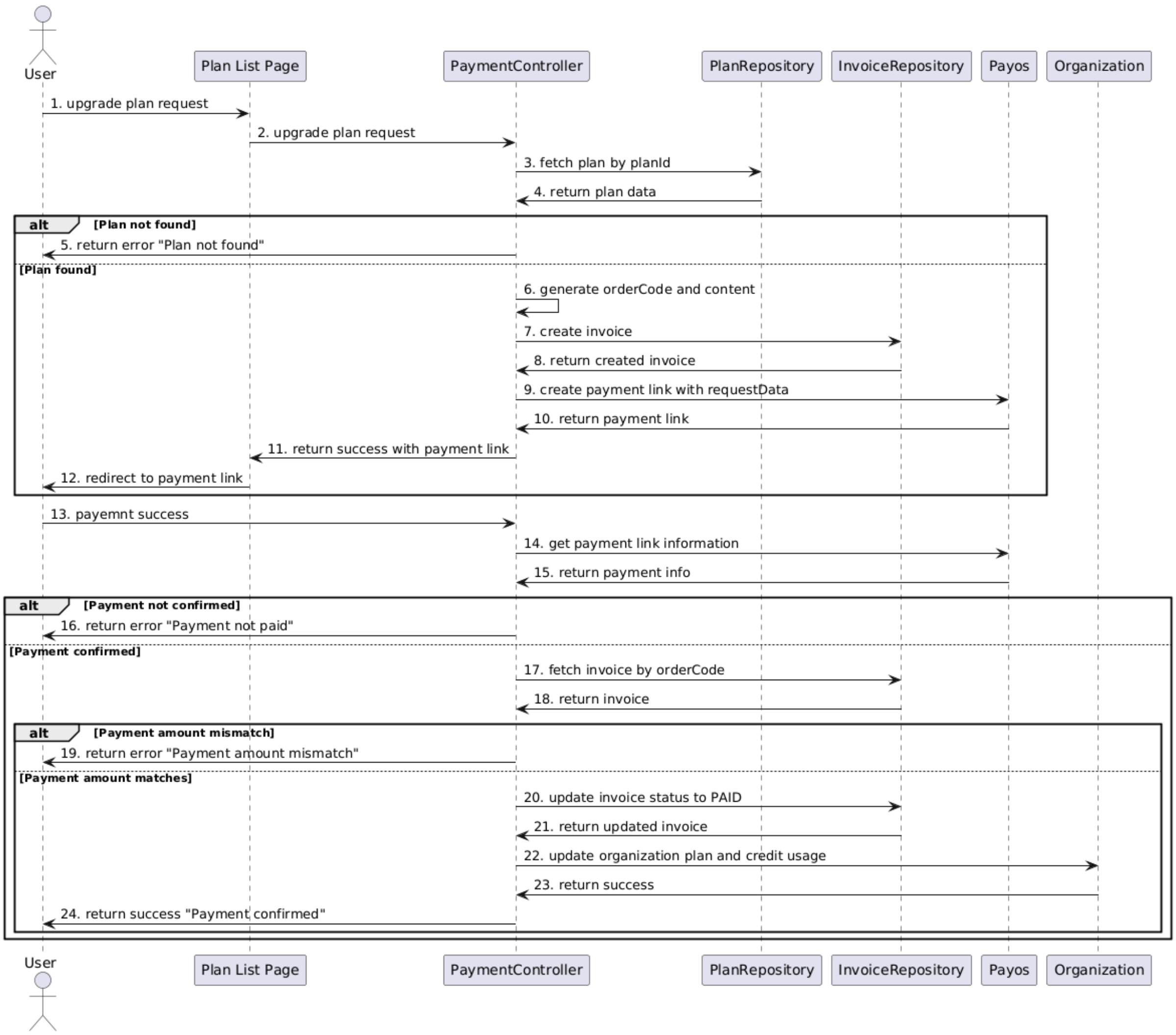


3.4.10 Upgrade Plan

a, Class Diagram

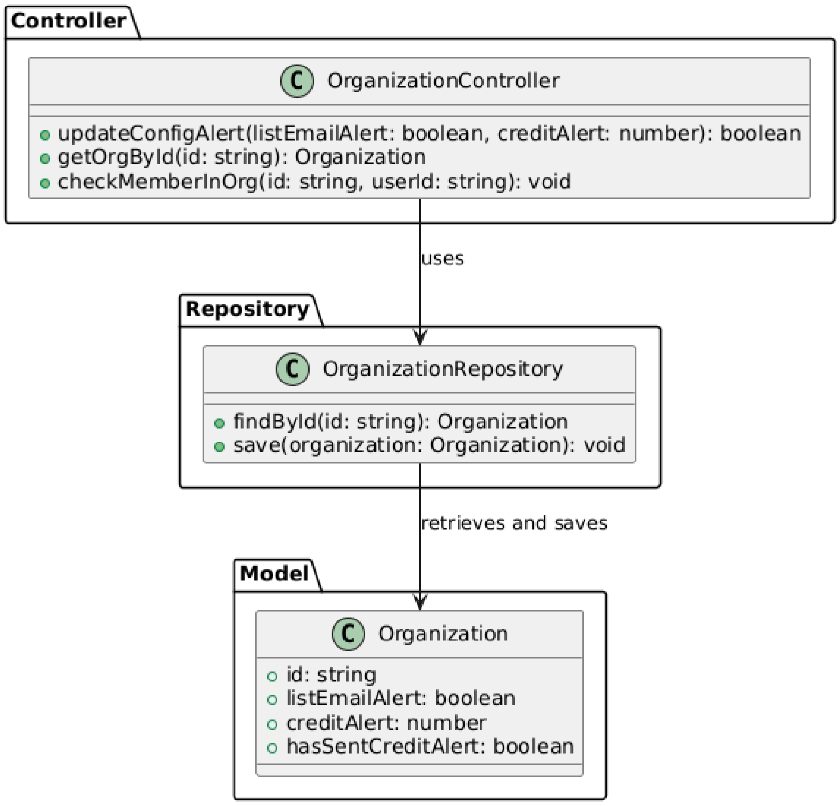


b, Sequence Diagram

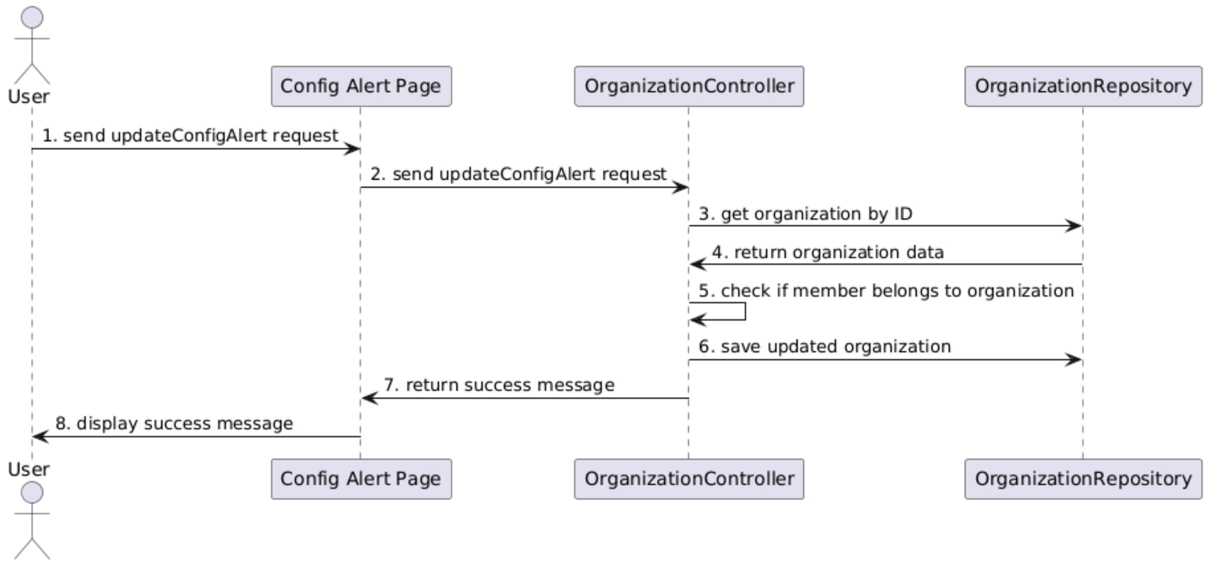


3.4.11 Setup Credit Alerts

a, Class Diagram



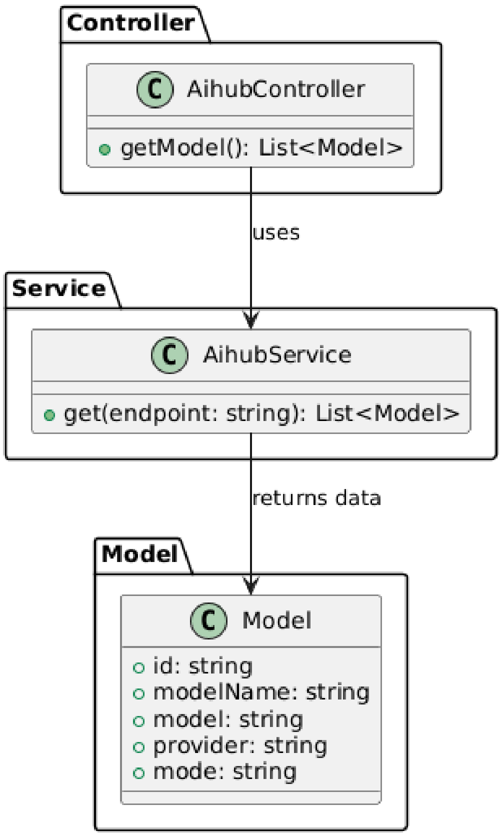
b, Sequence Diagram



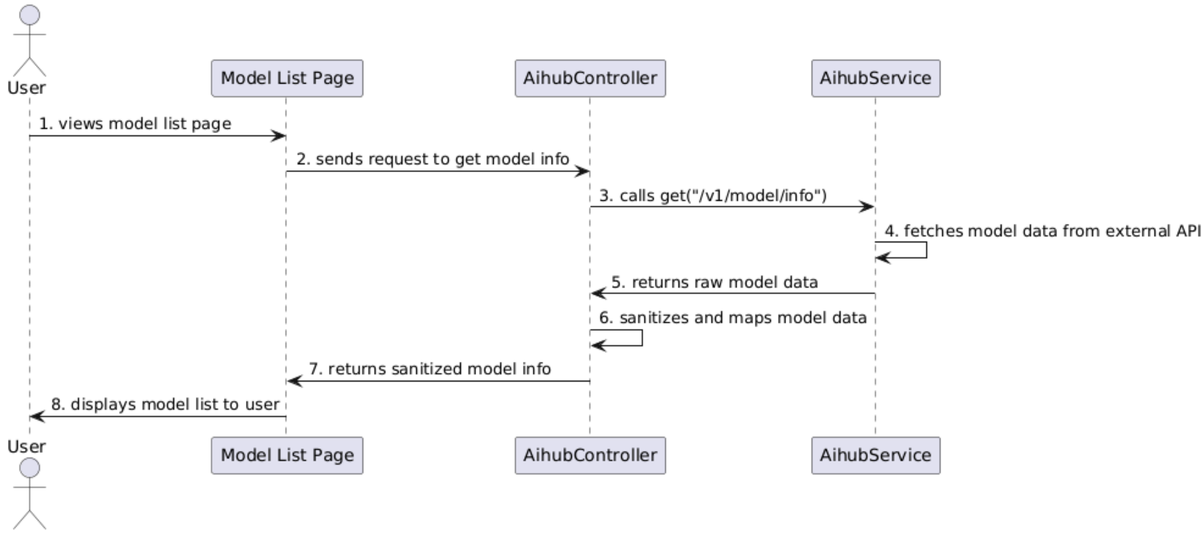
3.5 Admin Staff Feature

3.5.1 View Model List

a, Class Diagram

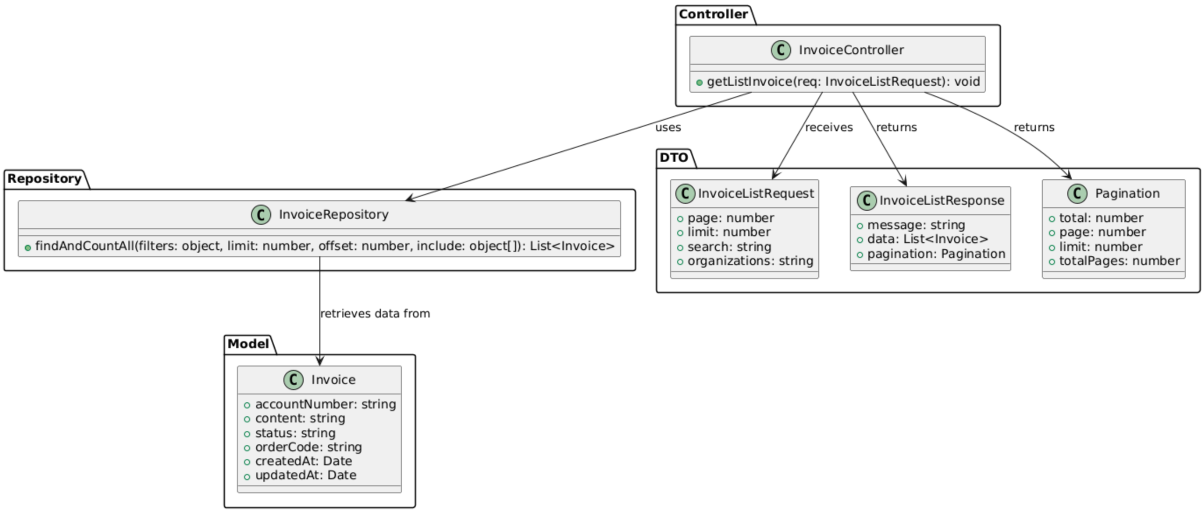


b, Sequence Diagram

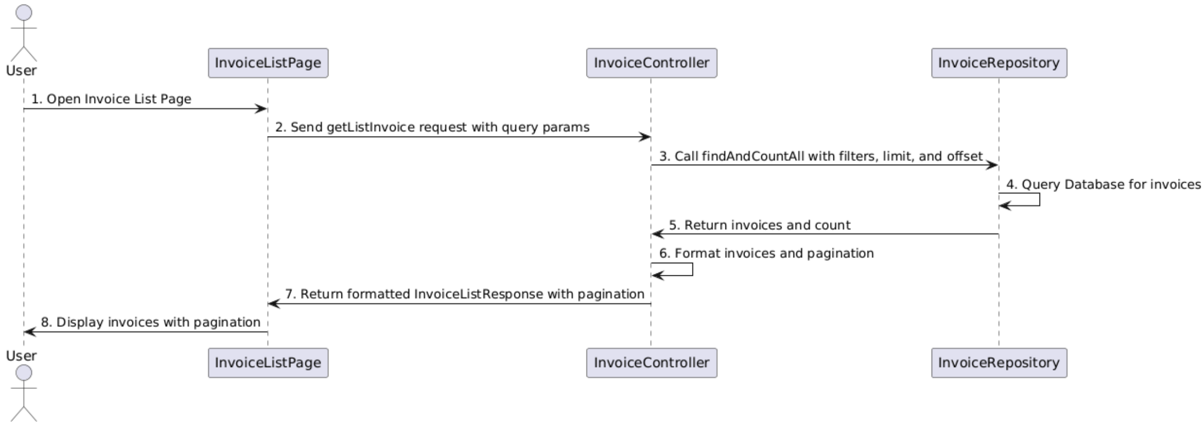


3.5.2 View Invoice List

a, Class Diagram

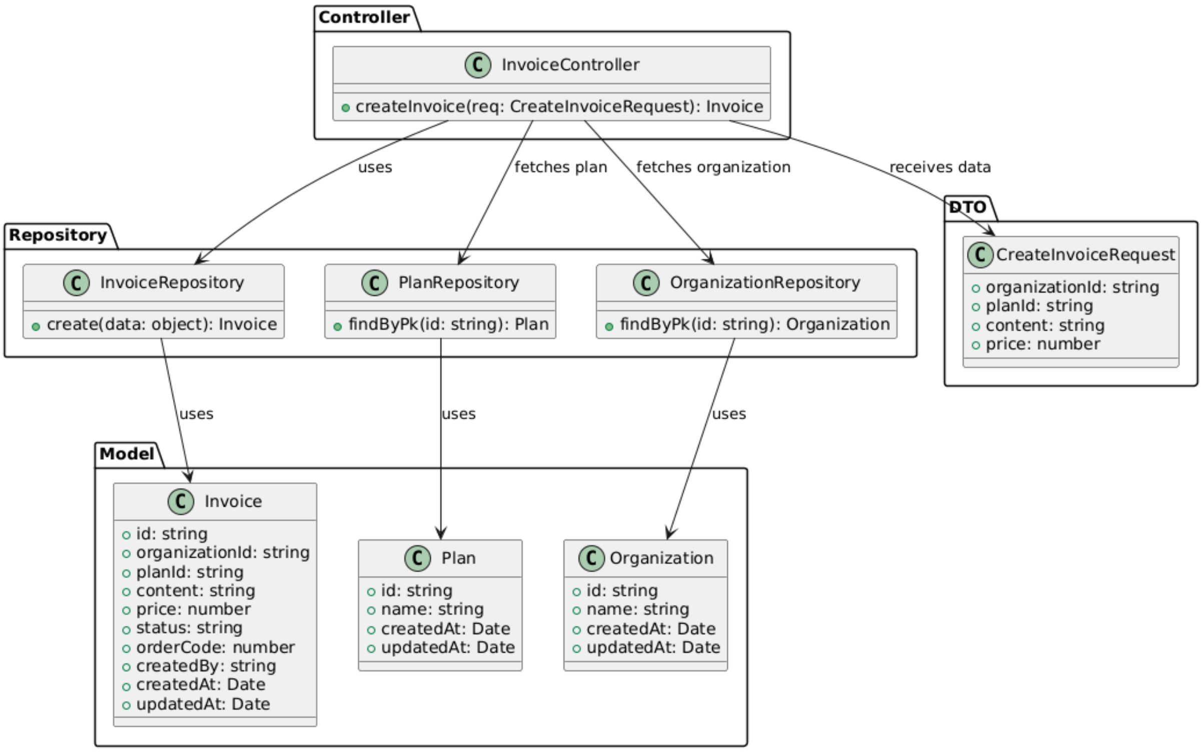


b, Sequence Diagram

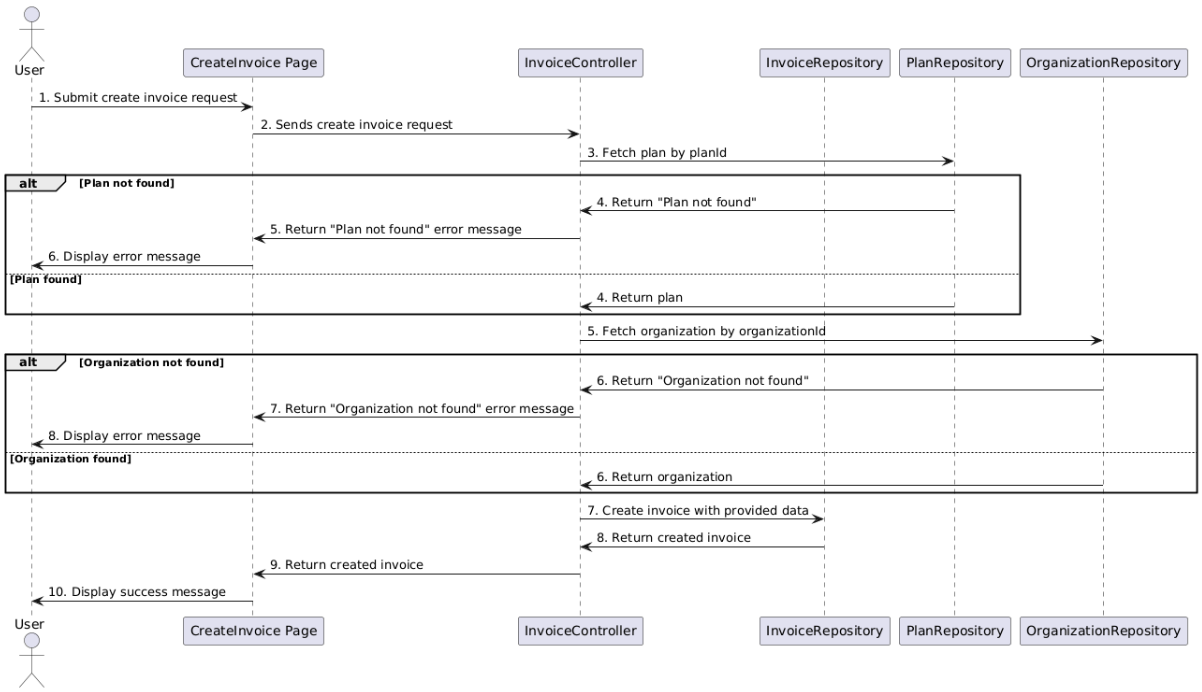


3.5.3 Add New Invoice

a, Class Diagram

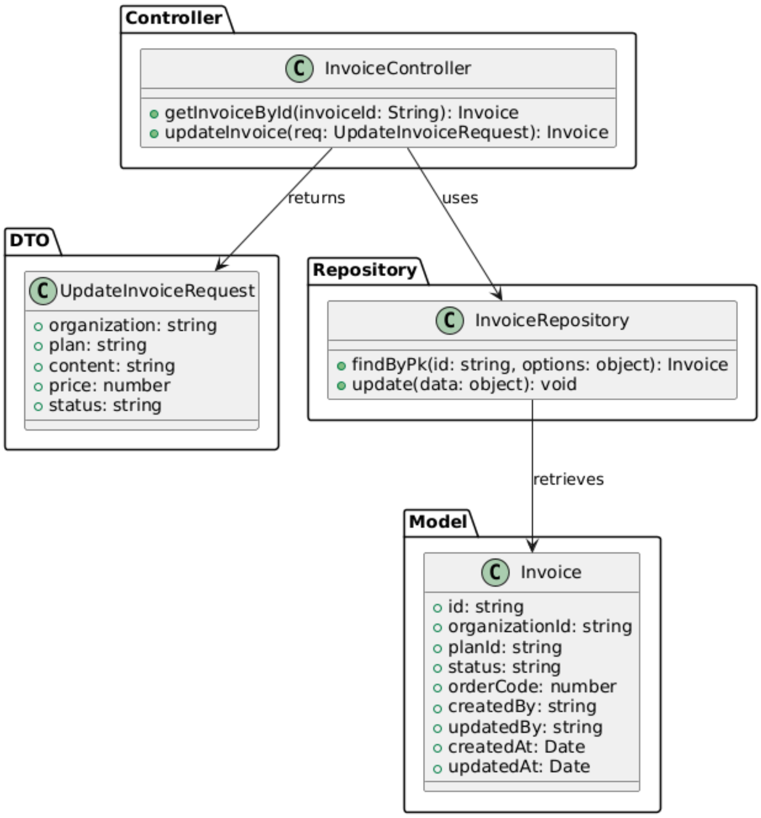


b, Sequence Diagram

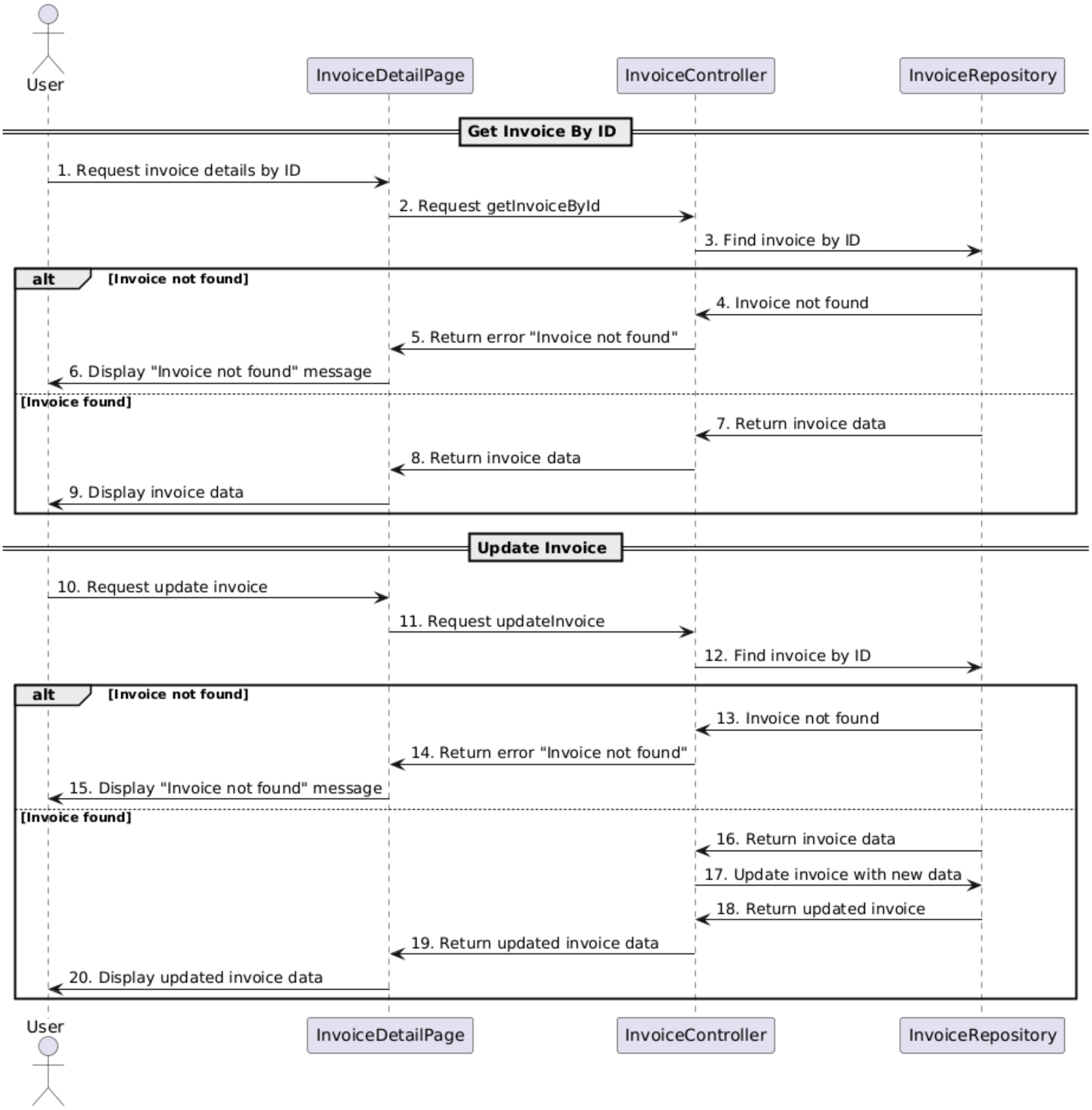


3.5.4 Update Invoice Details

a, Class Diagram



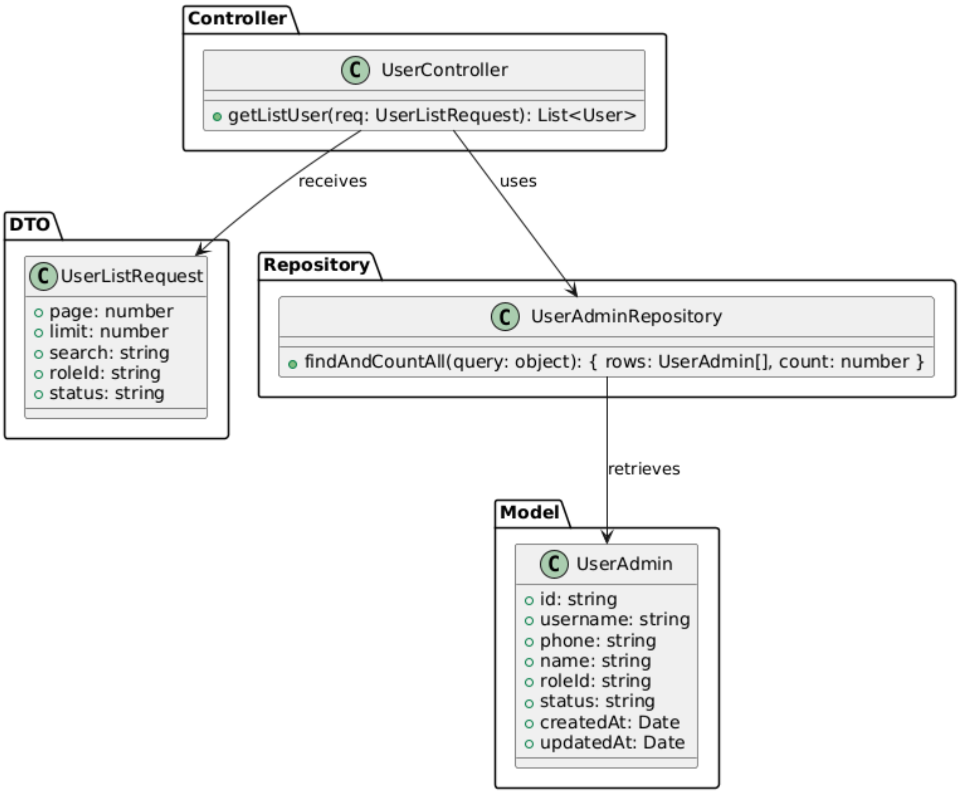
b, Sequence Diagram



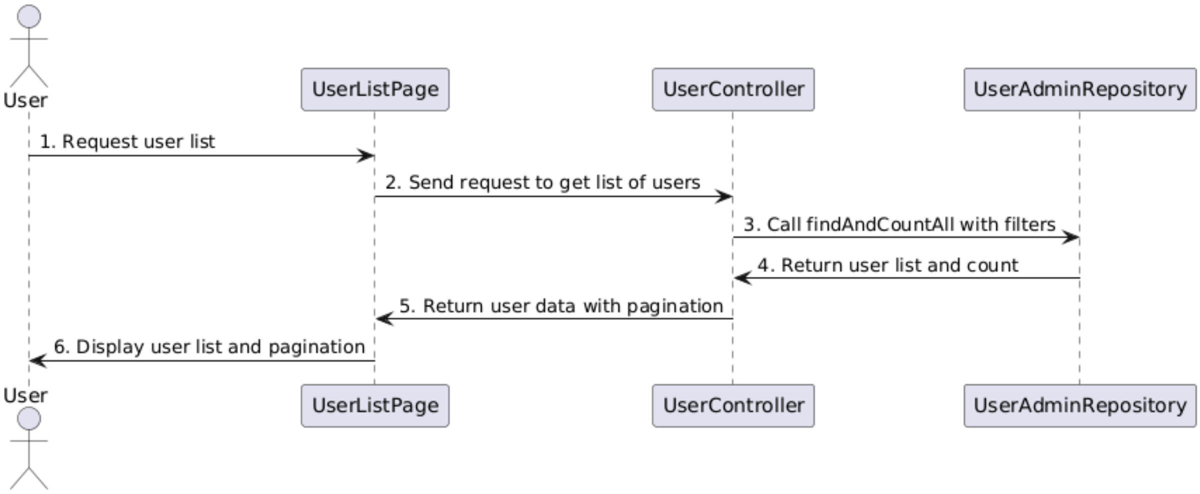
3.6 Admin Feature

3.6.1 View User List

a, Class Diagram

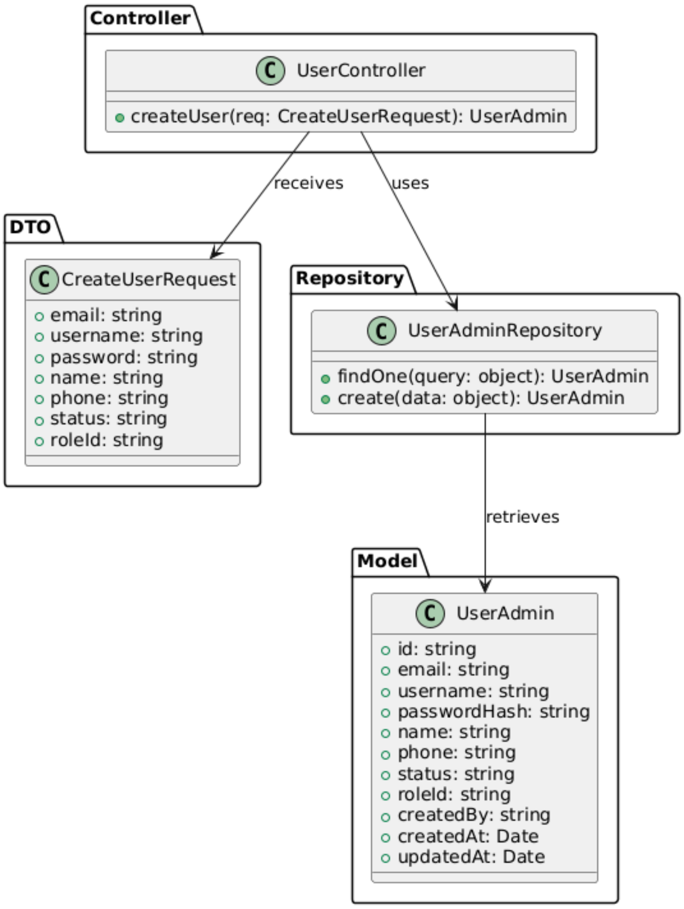


b, Sequence Diagram

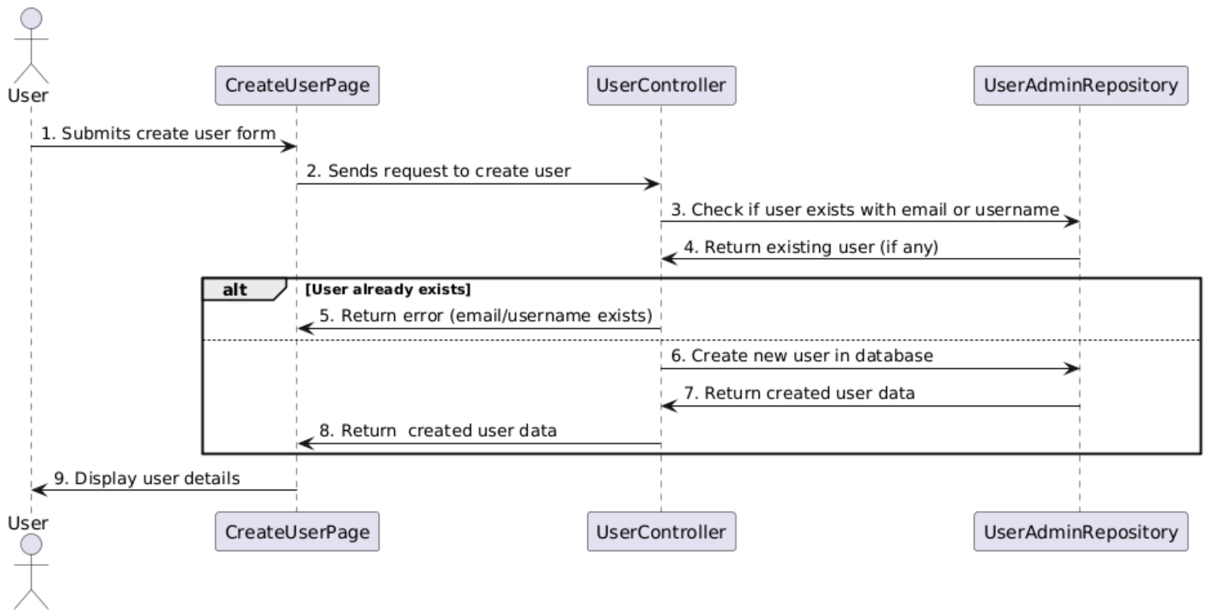


3.6.2 Add New User

a, Class Diagram

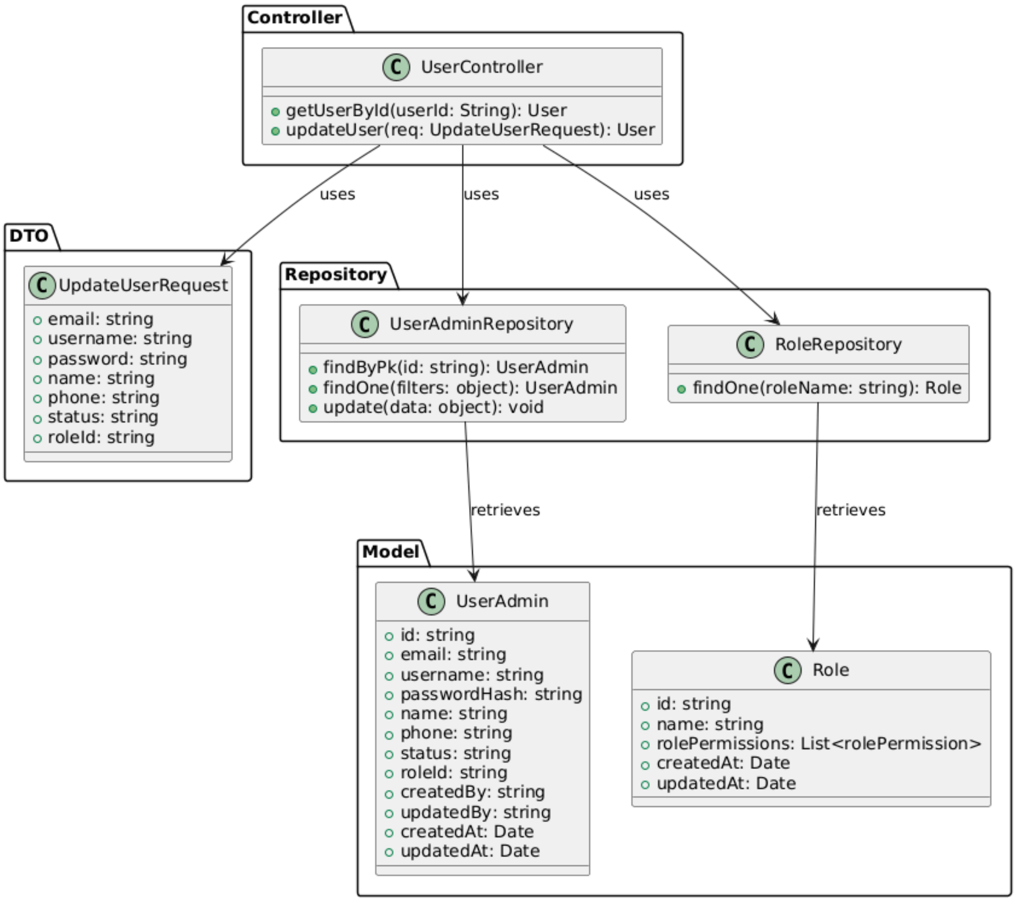


b, Sequence Diagram

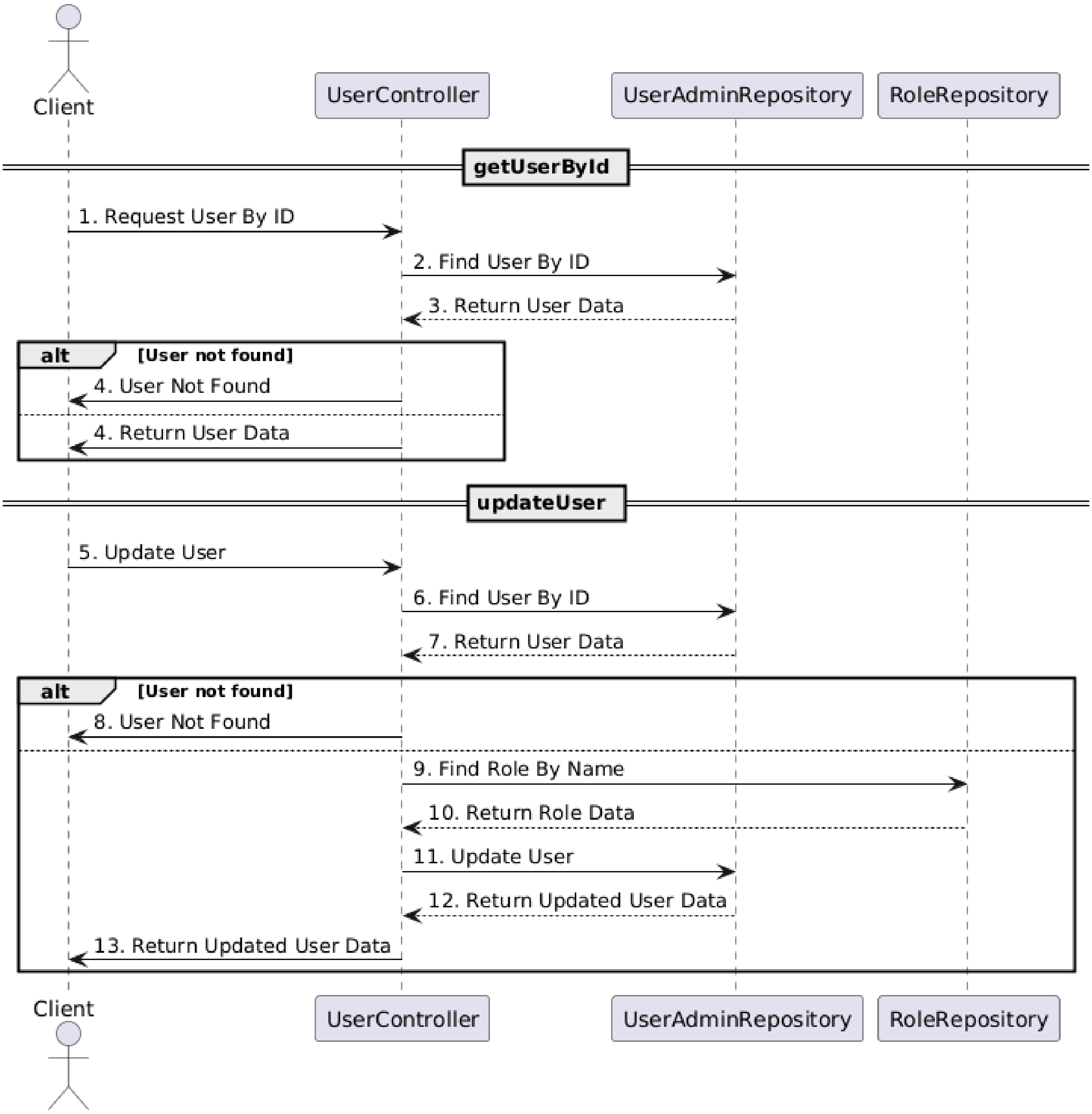


3.6.3 Update User Details

a, Class Diagram

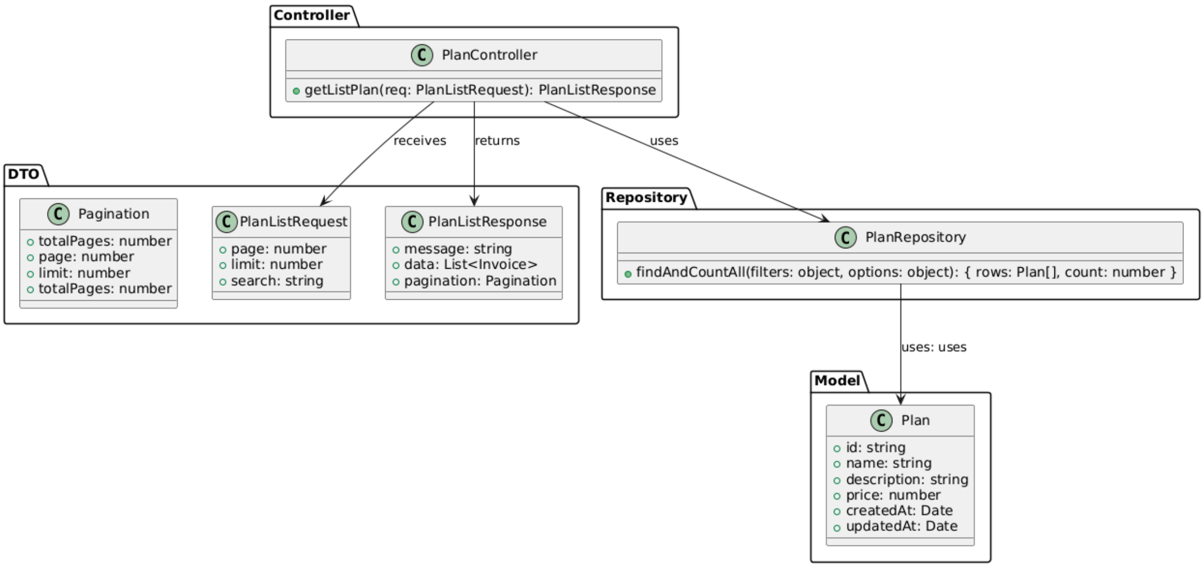


b, Sequence Diagram

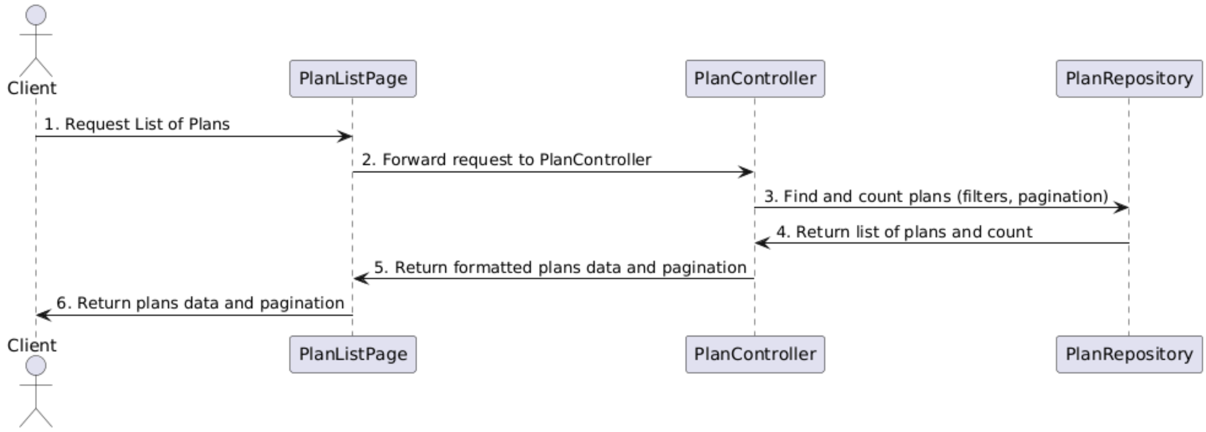


3.6.4 View Plan List

a, Class Diagram

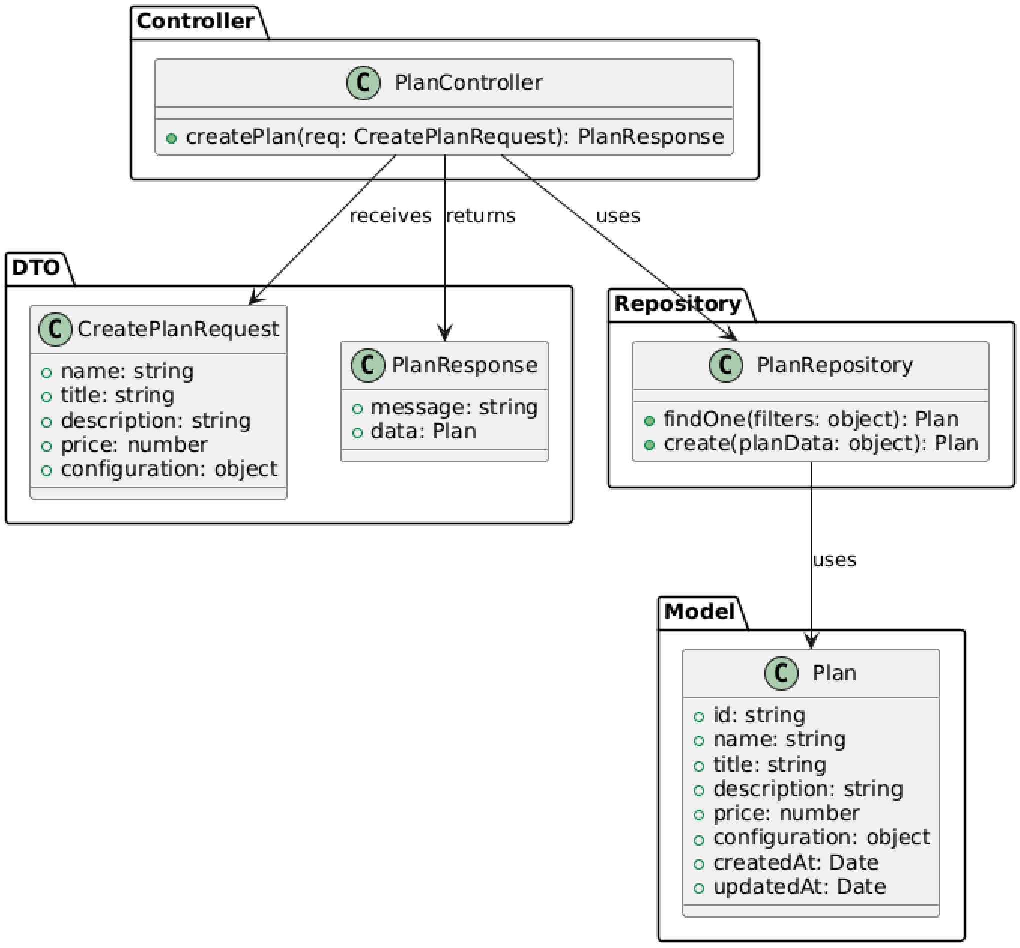


b, Sequence Diagram

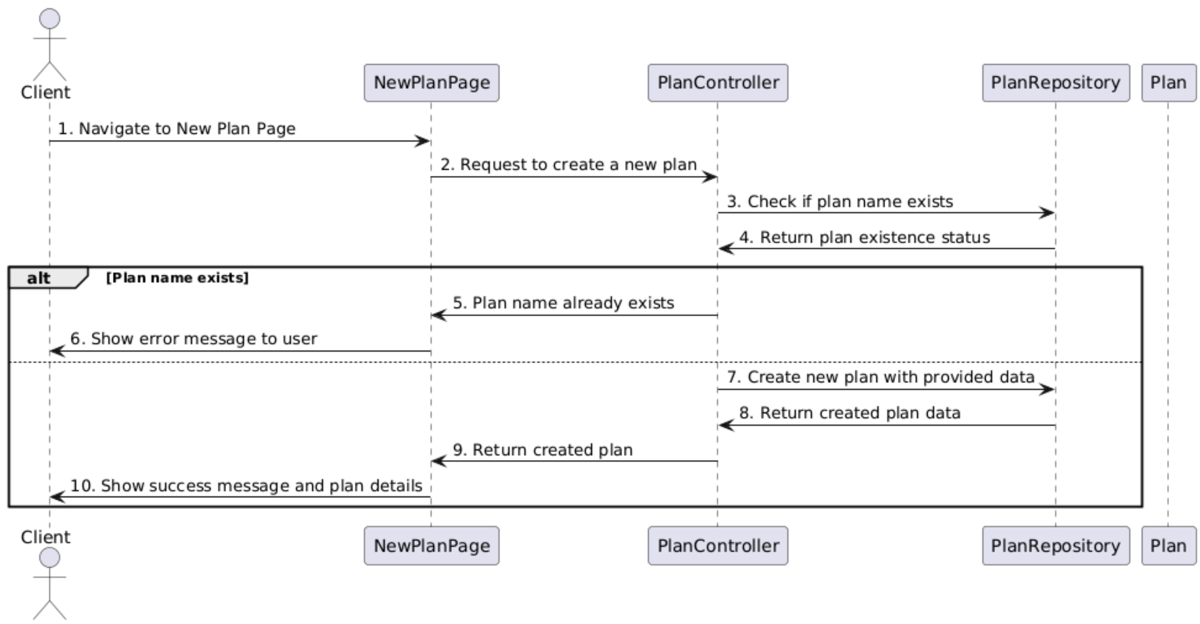


3.6.5 Add New Plan

a, Class Diagram

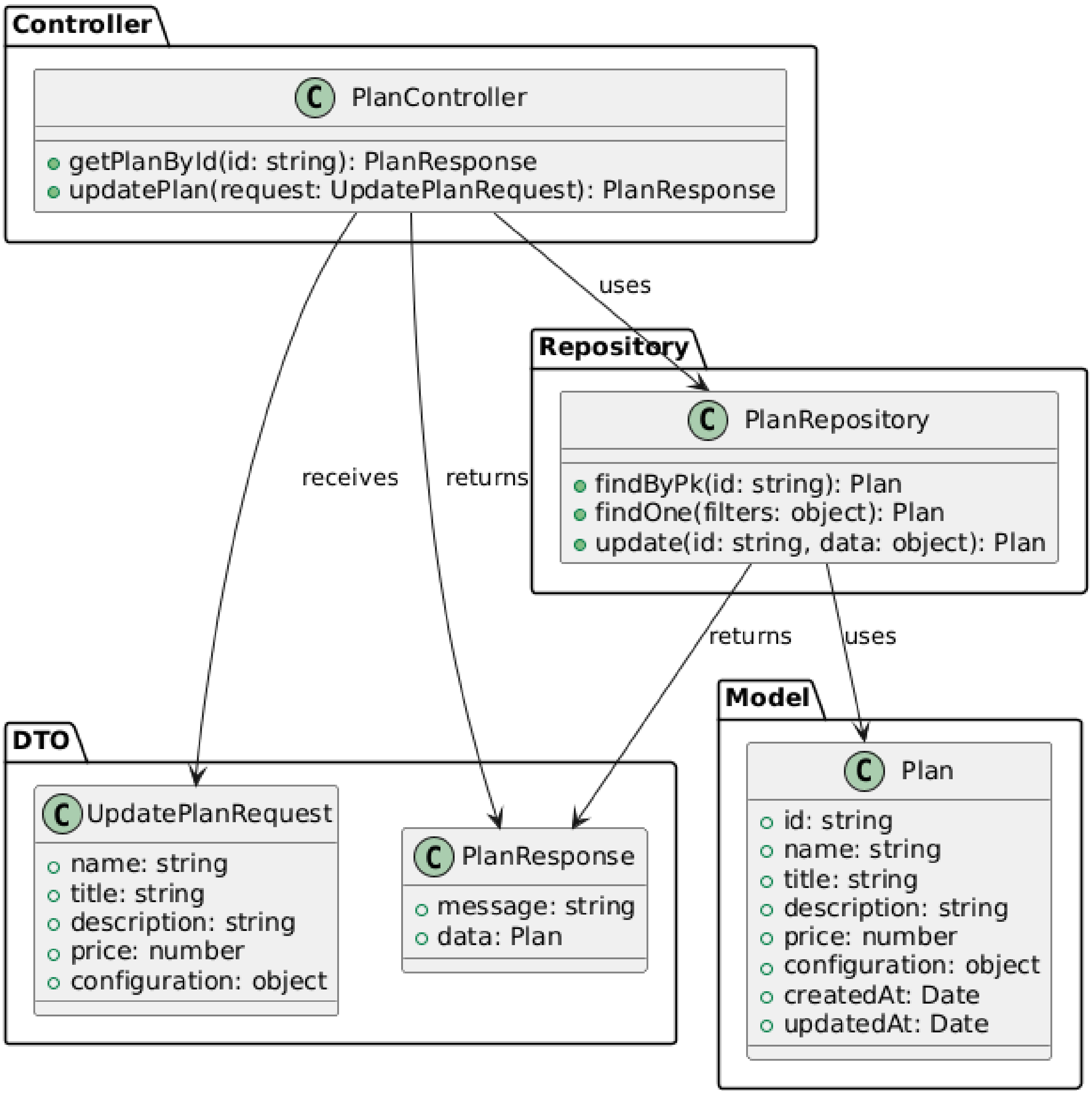


b, Sequence Diagram

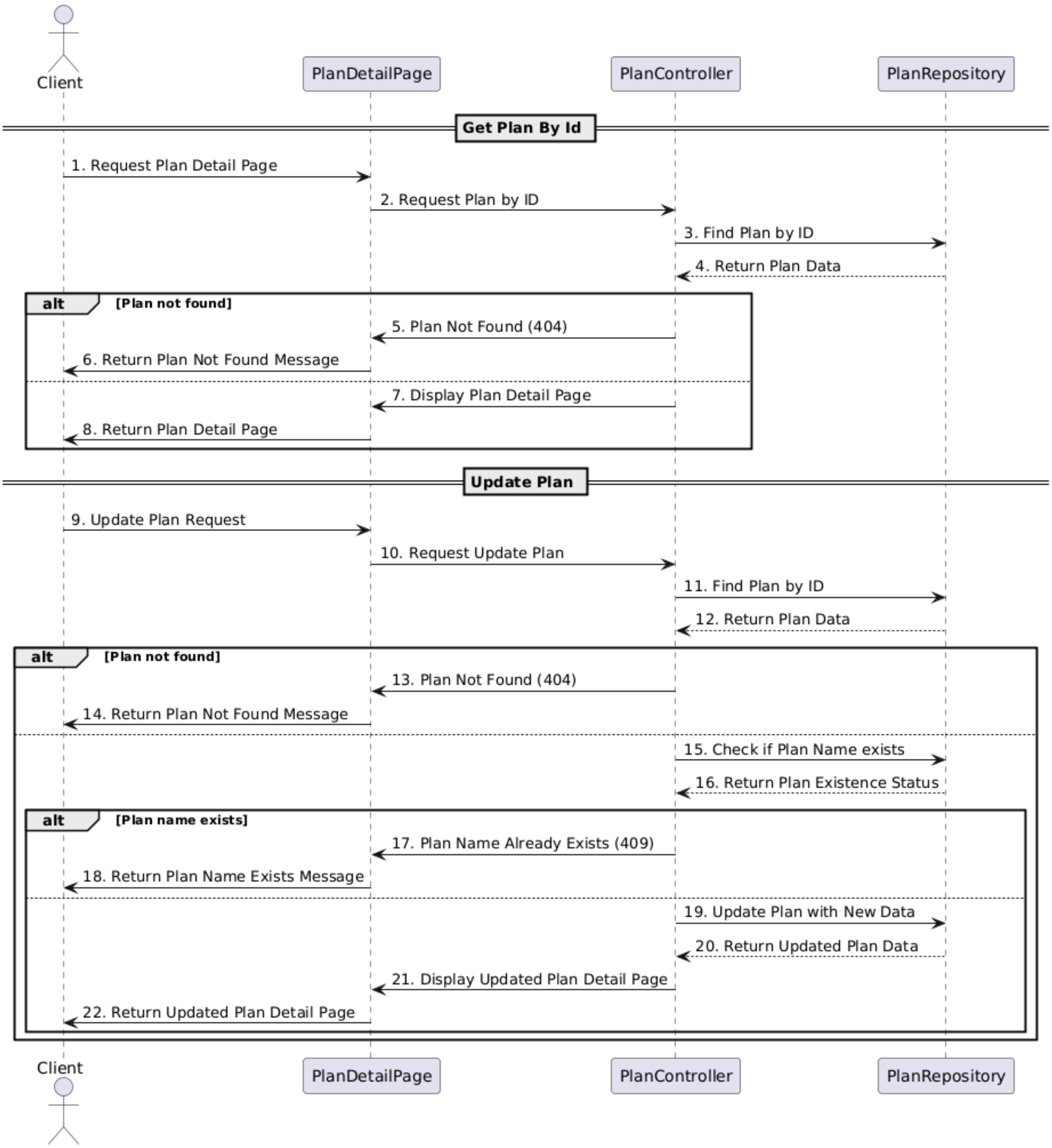


3.6.6 Update Plan Details

a, Class Diagram



b, Sequence Diagram



V. Software Testing Documentation

1. Scope of Testing

1.1. Testing Targets

1.1.1 Functional requirement

- The test scope includes all the features and functions defined in [SEP490_G47_Report3_SRS]
- Develop test cases to verify each functional requirement.

- Ensure that the software behaves as expected and meets the functional specifications.

1.1.2 Test coverage

- Enumerate major features and functions to be tested.
- Verify that the test coverage aligns with the requirements specified in [SEP490_G47_Report3_SRS].

1.1.3 Non - functional

Refer [SEP490_G47_Report3_SRS] - IV. Non-functional requirements

Refer [SEP490_G47_Report3_SRS] - V. Other Requirements

1.2. Testing Levels

We have 3 levels of testing: Unit Test, Integration Test, System Test

2. Test Strategy

2.1 Testing Types

The test team has to test the following types on Postman, Google Chrome browser, and Edge browser:

- Unit Testing
- GUI Testing
- API Testing
- Functional Testing
- Non-Functional Testing
- Regression Testing

2.2 Test Levels

Type of Tests	Test Level		
	Unit	Integration	System
Unit Test	X		
GUI Test		X	X
API Test		X	
Functional Testing	X	X	X
Non-functional Testing			X
Usability Testing			X
Regression Testing	X	X	X

Table 2.2.1. Test Environment

2.3 Supporting Tools

Purpose	Tool	Vendor/In-house	Version
Test Report	Microsoft Word	Microsoft	Office 16
Manage test case and data index	Google Sheet	Google	Latest
Manage bugs and the status of bugs	Jira Software	Atlassian	Latest
Unit Test	Jest	Openjsf Foundation	Latest
Testing API	Postman	Postman	2016

Table 2.3.1. Supporting Tools

3. Test Plan

3.1 Human Resources

Worker/Doer	Role	Specific Responsibilities/Comments
DucNM	PM	• Responsible for project schedule and overall success of the project. • Review test cases and reports.
DungHM	Tester	• Performing the actual system testing. • Manage test resources and assign test tasks. • Create a test plan. • Create test cases. • Create a test report. • Execute test. • Test log report.
DucNM LongPV DucTA AnhNMT	Team member	• Create unit tests. • Fix bugs

Table 3.1.1. Human Resources

3.2 Test Environment

Purpose	Tool	Provider
Write unit test documents	Excel	Microsoft
Write test report documents	Excel	Microsoft
Run UT	Postman	
Run IT	Chrome, Edge, Postman	
Run ST	Chrome, Edge	
Tracking defects, issues, and Q&A	Jira, Gitlab, Excel	

Table 3.2.1. Test Environment

3.3 Test Milestones

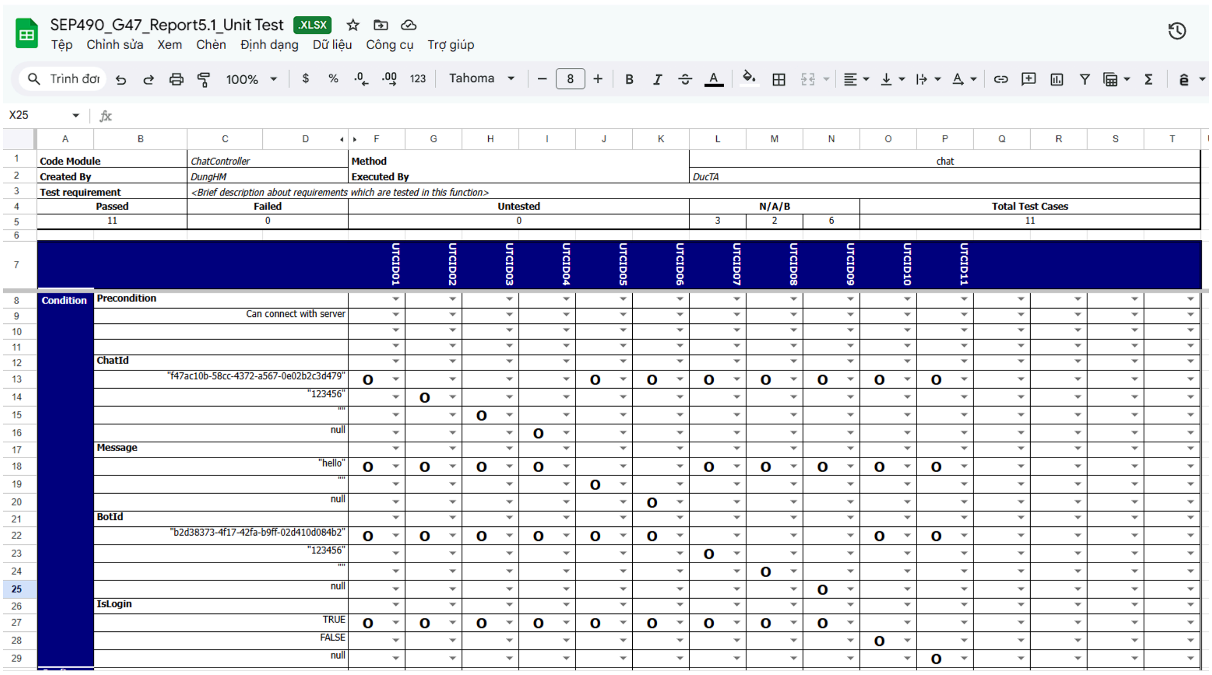
Milestone Task	Start Date	End Date
Create test plan	05/02/2025	09/02/2025
Create Test Report	05/02/2025	18/04/2025
Create, update, and execute UT test cases	19/02/2025	03/03/2025
Create, update, and execute IT test cases	22/03/2025	03/03/2025
Create, update, and execute UT test cases	04/03/2025	17/03/2025
Create, update, and execute IT test cases	08/03/2025	17/03/2025
Create, update, and execute UT test cases	18/03/2025	31/03/2025
Create, update, and execute IT test cases	22/03/2025	31/03/2025
Create, update, and execute ST test cases	01/04/2025	05/04/2025
Full system testing	01/04/2025	22/04/2025

Table 3.3.1. Test milestones plan

4. Test Cases

4.1 Unit Test

Functional testing will be done on the server development side by the backend developer:


SEP490_G47_Report5.1_Unit Test .XLSX

Code Module	ChatController	Method	chat									
Created By	DungHM	Executed By	DucTA									
Test requirement	<Brief description about requirements which are tested in this function>											
Passed	Failed	Untested	N/A/B			Total Test Cases						
11	0	0	3	2	6	11						
		UTCID01	UTCID02	UTCID03	UTCID04	UTCID05	UTCID06	UTCID07	UTCID08	UTCID09	UTCID10	UTCID11
Condition	Precondition											
	Can connect with server											
	ChatId											
	"f47ac10b-58cc-4372-a567-0e02b2c3d479"	O				O	O	O	O	O	O	O
	"123456"		O									
	""			O								
	null				O							
	Message											
	"hello"	O	O	O	O			O	O	O	O	O
	""					O						
	null						O					
	BotId											
	"b2d38373-4f17-42fa-b9ff-02d410d084b2"	O	O	O	O	O	O				O	O
	"123456"							O				
	""								O			
	null									O		
	IsLogin											
	TRUE	O	O	O	O	O	O	O	O	O		
	FALSE										O	
	null											O

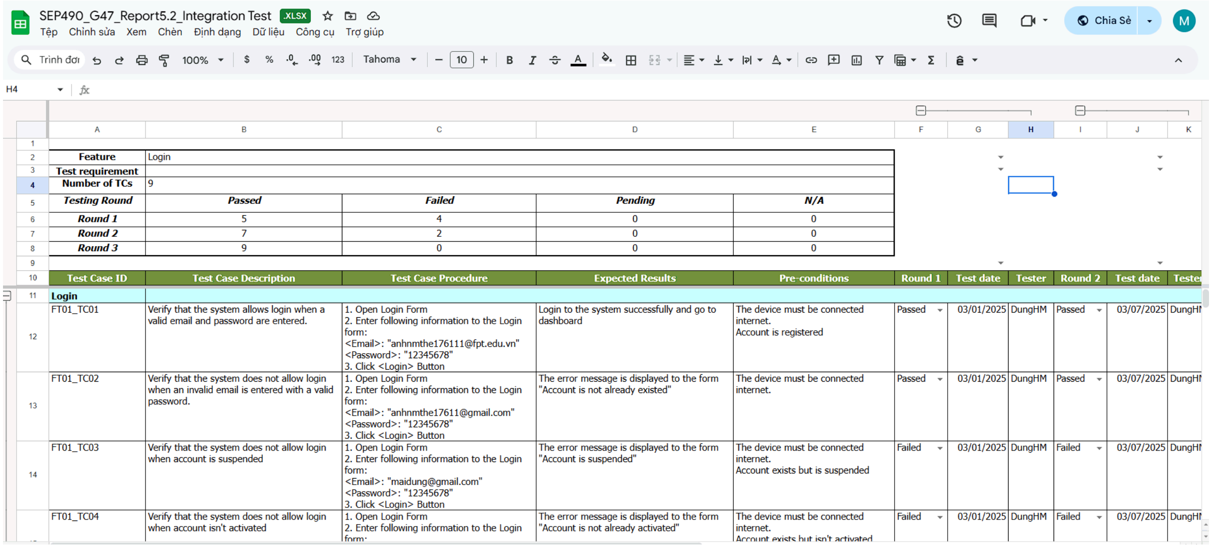
Figure 4.1.1. Test cases for Unit Test

Detail test cases for Unit Test will be described in:

SEP490-G47_Report5.1_UnitTest.xlxs

4.2 Integration Test

Integration testing will involve incorporating the API into the app's interface to assess the data representation for user The tester will select test cases from the Test Report wherein actions are performed between functions or data is exchanged between modules.



Feature	Login			
Test requirement				
Number of TCs	9			
Testing Round	***Passed***	***Failed***	***Pending***	***N/A***
Round 1	5	4	0	0
Round 2	7	2	0	0
Round 3	9	0	0	0

Test Case ID	Test Case Description	Test Case Procedure	Expected Results	Pre-conditions	Round 1	Test date	Tester	Round 2	Test date	Tester
Login										
FT01_TC01	Verify that the system allows login when a valid email and password are entered.	1. Open Login Form 2. Enter following information to the Login form: <Email>: "anhnmthe176111@fpt.edu.vn" <Password>: "12345678" 3. Click <Login> Button	Login to the system successfully and go to dashboard	The device must be connected internet. Account is registered	Passed	03/01/2025	DungHM	Passed	03/07/2025	DungH
FT01_TC02	Verify that the system does not allow login when an invalid email is entered with a valid password.	1. Open Login Form 2. Enter following information to the Login form: <Email>: "anhnmthe17611@gmail.com" <Password>: "12345678" 3. Click <Login> Button	The error message is displayed to the form "Account is not already existed"	The device must be connected internet.	Passed	03/01/2025	DungHM	Passed	03/07/2025	DungH
FT01_TC03	Verify that the system does not allow login when account is suspended	1. Open Login Form 2. Enter following information to the Login form: <Email>: "maidung@gmail.com" <Password>: "12345678" 3. Click <Login> Button	The error message is displayed to the form "Account is suspended"	The device must be connected internet. Account exists but is suspended	Failed	03/01/2025	DungHM	Failed	03/07/2025	DungH
FT01_TC04	Verify that the system does not allow login when account isn't activated	1. Open Login Form 2. Enter following information to the Login form:	The error message is displayed to the form "Account is not already activated"	The device must be connected internet. Account exists but isn't activated	Failed	03/01/2025	DungHM	Failed	03/07/2025	DungH

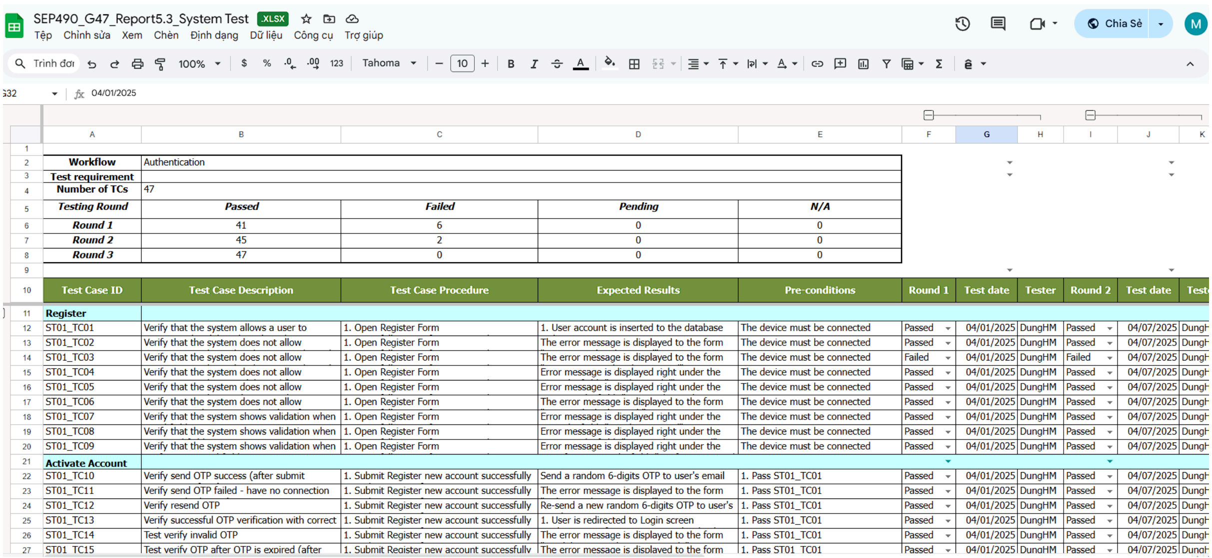
Figure 4.2.1. Test cases for Integration Test

Detail test cases for Integration Test will be described in:

SEP490-G47_Report5.2_IntegrationTest.xlxs

4.3 System Test

This testing stage occurs after the functions outlined in the requirements have been completely implemented. At this stage, the entire system is tested as a whole to ensure that all components work correctly and that the overall workflow behaves as designed



Workflow	Authentication			
Test requirement				
Number of TCs	47			
Testing Round	***Passed***	***Failed***	***Pending***	***N/A***
Round 1	41	6	0	0
Round 2	45	2	0	0
Round 3	47	0	0	0

Test Case ID	Test Case Description	Test Case Procedure	Expected Results	Pre-conditions	Round 1	Test date	Tester	Round 2	Test date	Tester
Register										
ST01_TC01	Verify that the system allows a user to	1. Open Register Form	1. User account is inserted to the database	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC02	Verify that the system does not allow	1. Open Register Form	The error message is displayed to the form	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC03	Verify that the system does not allow	1. Open Register Form	The error message is displayed to the form	The device must be connected	Failed	04/01/2025	DungHM	Failed	04/07/2025	DungH
ST01_TC04	Verify that the system does not allow	1. Open Register Form	Error message is displayed right under the	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC05	Verify that the system does not allow	1. Open Register Form	Error message is displayed right under the	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC06	Verify that the system does not allow	1. Open Register Form	The error message is displayed to the form	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC07	Verify that the system shows validation when	1. Open Register Form	Error message is displayed right under the	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC08	Verify that the system shows validation when	1. Open Register Form	Error message is displayed right under the	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC09	Verify that the system shows validation when	1. Open Register Form	Error message is displayed right under the	The device must be connected	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
Activate Account										
ST01_TC10	Verify send OTP success (after submit	1. Submit Register new account successfully	Send a random 6-digits OTP to user's email	1. Pass ST01_TC01	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC11	Verify send OTP failed - have no connection	1. Submit Register new account successfully	The error message is displayed to the form	1. Pass ST01_TC01	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC12	Verify resend OTP	1. Submit Register new account successfully	Re-send a new random 6-digits OTP to user's	1. Pass ST01_TC01	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC13	Verify successful OTP verification with correct	1. Submit Register new account successfully	1. User is redirected to Login screen	1. Pass ST01_TC01	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC14	Test verify invalid OTP	1. Submit Register new account successfully	The error message is displayed to the form	1. Pass ST01_TC01	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH
ST01_TC15	Test verify OTP after OTP is expired (after	1. Submit Register new account successfully	The error message is displayed to the form	1. Pass ST01_TC01	Passed	04/01/2025	DungHM	Passed	04/07/2025	DungH

Figure 4.3.1. Test cases for System Test

Detail test cases for System Test will be described in:

SEP490-G47_Report5.3_SystemTest.xlxs

5. Test Reports

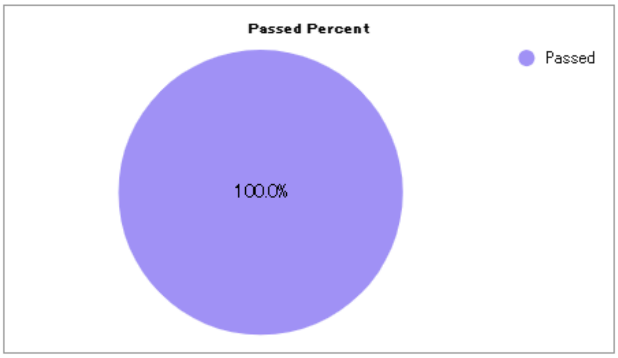
5.1 Unit Test

No	Function code	Passed	Failed	Untested	N	A	B	Total Test Cases
1	chat	11	0	0	3	2	6	11
2	getLiveChatList	20	0	0	1	2	17	20
3	getMessagesByLiveChatId	12	0	0	3	1	8	12
4	getListChat	12	0	0	1	1	10	12
5	getChatMessage	12	0	0	3	1	8	12
6	chat	9	0	0	3	2	6	11
7	getPromptDocument	11	0	0	1	2	8	11
8	getHistoryMessages	8	0	0	2	1	5	8
9	getListBot	11	0	0	4	0	7	11
10	getBotById	4	0	0	1	0	3	4
11	getBotConfigurationById	4	0	0	1	0	3	4
12	createBot	8	0	0	1	1	6	8
13	updateBot	8	0	0	1	1	6	8S
14	getPaginateDataContent	17	0	0	1	2	14	17
15	getDetailDataContent	4	0	0	1	0	3	4
16	createDataContent	6	0	0	1	1	4	6
17	createDataContentURL	9	0	0	1	2	6	9
18	createDataContentText	8	0	0	1	1	6	8
19	updateDataContent	13	0	0	1	2	10	13
20	updateStatusDataContent	8	0	0	1	1	6	8
21	createPaymentLink	4	0	0	1	1	2	4
22	confirmPayment	4	0	0	1	1	2	4
23	insert_data	10	0	0	1	2	7	10
24	insert_data_url	12	0	0	1	3	8	12
25	insert_data_text	11	0	0	1	2	8	11
26	query_data	9	0	0	1	2	6	9
27	HandleUserMessage	17	0	0	2	4	11	17
	Sub total	270	0	0	40	40	190	270
	Test coverage		**100.00**	%				
	Test successful coverage		**100.00**	%				
	Normal case		**14.55**	%				
	Abnormal case		**10.91**	%				
	Boundary case		**74.55**	%				

Table 5.1.1. Unit Test Report

Sub total	55	0	0	8	6	41	55

Test coverage	100.00	%
Test successful coverage	100.00	%
Normal case	14.55	%
Abnormal case	10.91	%
Boundary case	74.55	%



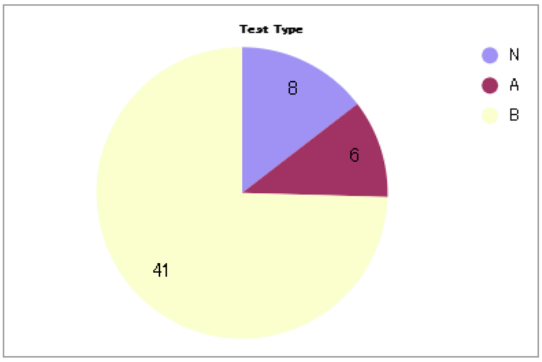


Figure 5.1.1. Unit Test Statistics

5.2 Integration Test

No	Module code	Passed	Failed	Pending	N/A	Number of test cases
1	Login	9	0	0	0	9
2	Register	17	0	0	0	17
3	Password Reset	11	0	0	0	11
4	Profile	10	0	0	0	10
5	User List	33	0	0	0	33
6	Chat	6	0	0	0	6
7	Live chat	4	0	0	0	4
8	Bot	16	0	0	0	16
9	Bot Details	40	0	0	0	40
10	Model	12	0	0	0	12
11	Organization	14	0	0	0	14
12	Organization Details	24	0	0	0	24
13	Role	18	0	0	0	18
14	Plan	24	0	0	0	24
15	Invoice	23	0	0	0	23
	Sub total	261	0	0	0	261
	Test coverage		100.00	%		
	Test successful coverage		100.00	%		

Table 5.2.1. Integration Test Report

5.3 System Test

No	Module code	Passed	Failed	Pending	N/A	Number of test cases
1	Authentication	47	0	0	0	47
2	User Management	51	0	0	0	51
3	Plan Process & Payment	47	0	0	0	47
4	Bot Config	47	0	0	0	47

5	Chat	19	0	0	0	19
6	Business Management	24	0	0	0	24
	Sub total	235	0	0	0	235
	Test coverage		**100.00**	%		
	Test successful coverage		**100.00**	%		

Figure 5.3.1. System Test Report

VI. Release Package & User Guides

1. Deliverable Package

No.	Deliverable Item	Description
1	Project Schedule/Tracking	SEP490_G47_Report2_Project_Tracking.xlxs
2	Source Codes	SEP490_G47_CMS_BE.zip
		SEP490_G47_CMS_FE.zip
		SEP490_G47_Job_Server.zip
		SEP490_G47_Socket_Server.zip
		SEP490_G47_Data_Hub.zip
		SEP490_G47_AI_Hub.zip
		SEP490_G47_Report.zip
3	Database Script(s)	SEP490_G47_Database_script.sql
4	Final Report Document	SEP490_G47_Report7_Final_Project_Report.pdf
5	Unit Test	SEP490_G47_Report5.1_Unit_Test.xlxs
5	Integration Test	SEP490_G47_Report5.2_Integration_Test.xlxs
6	System Test	SEP490_G47_Report5.3_System_Test.xlxs
7	Defects List	SEP490_G47_Report2_Project_Tracking.xlxs
8	Slide	SEP490_G47_Slide_Final_Presentation.pdf

Table 1.1. Deliverable Packages

2. Installation Guides

2.1 System Requirements

- For Develop CAIP CMS-FE application, the system requirements
  - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB
  - **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
  - **Software:** Visual Studio Code
  - **Technologies:** Node.js v22, Git

- For Develop CAIP CMS-BE application, the system requirements
  - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB
  - **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
  - **Software:** Visual Studio Code
  - **Technologies:** Node.js v22, Git

- For Develop CAIP Job Server application, the system requirements
  - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB

- **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
    - **Software:** Visual Studio Code
    - **Technologies:** Node.js v22, Git

- For Develop CAIP Socket Server application, the system requirements
    - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB
    - **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
    - **Software:** Visual Studio Code
    - **Technologies:** Node.js v22, Git

- For Develop CAIP Report application, the system requirements
    - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB
    - **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
    - **Software:** Visual Studio Code
    - **Technologies:** Node.js v22, Git

- For Develop CAIP Data Hub application, the system requirements
    - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB
    - **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
    - **Software:** Visual Studio Code
    - **Technologies:** Python 3.9, Git

- For Develop CAIP AI Hub application, the system requirements
    - **Hardware:** Intel® CoreTM i5 or higher, RAM at least 4.00GB
    - **Operating System:** Windows 10/11 or Ubuntu 22.04 LTS 64-bit
    - **Software:** Visual Studio Code
    - **Technologies:** Python 3.9, Git

- For Production Environment, the system requirements
    - **Hardware:** CPU at least 8 cores, RAM at least 16.00GB
    - **Storage:** 256GB SSD
    - **Operating System:** Ubuntu 22.04 LTS 64-bit
    - **Technologies:** Docker, Git

2.2 Installation Instruction

2.2.1 Install Docker

1. Setup Docker's apt repository

```
# Add Docker’s official GPG key
sudo apt update
sudo apt install -y apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

# Add Docker repository
```

```
echo "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

2. Install the Docker packages

```
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io
# Add user to Docker group
sudo usermod -aG docker $USER
```

3. Verify Docker installation

```
docker --version
```

4. **Log out** and **Log back** in to apply group changes

2.2.2 Install Docker Compose and Infrastructure Services

1. Install the Docker Compose

```
sudo curl -L
"https://github.com/docker/compose/releases/download/v2.29.2/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

2. Verify installation

```
docker-compose --version
```

3. Create .env file modify environment variables if needed
4. Create file docker-compose.yml

```
version: '3.8'
services:
  rabbitmq:
    image: rabbitmq:3-management-alpine
    container_name: rabbitmq
    ports:
      - 5672:5672
      - 15672:15672
    env_file:
      - .env
    environment:
      RABBITMQ_DEFAULT_USER: ${RABBITMQ_DEFAULT_USER}
      RABBITMQ_DEFAULT_PASS: ${RABBITMQ_DEFAULT_PASS}
    volumes:
      - rabbitmq_data:/var/lib/rabbitmq
    restart: always
    networks:
      - caip-network
```

```yaml
  mongodb:
    image: mongo:6.0
    container_name: mongodb
    ports:
      - 27017:27017
    env_file:
      - .env
    environment:
      MONGO_INITDB_ROOT_USERNAME: ${MONGO_INITDB_ROOT_USERNAME}
      MONGO_INITDB_ROOT_PASSWORD: ${MONGO_INITDB_ROOT_PASSWORD}
    volumes:
      - mongodb_data:/data/db
    restart: always
    networks:
      - caip-network

  postgresql:
    image: postgres:15
    container_name: postgresql
    ports:
      - 5432:5432
    env_file:
      - .env
    environment:
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
      POSTGRES_DB: ${POSTGRES_DB}
    volumes:
      - postgresql_data:/var/lib/postgresql/data
      - ./SEP490_G47_Database_script.sql:/docker-entrypoint-initdb.d/SEP490_G47_Database_script.sql
    restart: always
    networks:
      - caip-network

volumes:
  rabbitmq_data:
  mongodb_data:
  postgresql_data:

networks:
  caip-network:
    driver: bridge
```

5. Start the services

```
docker-compose up -d

# view services
docker ps
```

2.2.3 Install CAIP CMS-FE

1. Create .env file by copy .env.example file content and modify environment variables if needed
2. Start the services

```
docker build -t cms-fe:latest .
docker run -d --name cms-fe -p 5173:5173 cms-fe:latest
```

2.2.4 Install CAIP CMS-BE

1. Create .env file by copy .env.example file content and modify environment variables if needed
2. Start the services

```
docker build -t cms-be:latest .
docker run -d --name cms-be -p 8000:8000 cms-be:latest
```

2.2.5 Install CAIP Job Server

1. Create .env file by copy .env.example file content and modify environment variables if needed
2. Start the services

```
docker build -t job-server:latest .
docker run -d --name job-server -p 8100:8100 cms-be:latest
```

2.2.6 Install CAIP Socket Server

1. Create .env file by copy .env.example file content and modify environment variables if needed
2. Start the services

```
docker build -t socket-server:latest .
docker run -d --name socket-server -p 5001:5001 socket-server:latest
```

2.2.7 Install CAIP Report

1. Create .env file by copy .env.example file content and modify environment variables if needed
2. Start the services

```
docker build -t report:latest .
docker run -d --name report -p 8300:8300 report:latest
```

2.2.8 Install CAIP Data Hub

1. Create .env file by copy .env.example file content and modify environment variables if needed
2. Start the services

```
docker build -t datahub:latest .
```

```
docker run -d --name datahub -p 8200:8200 datahub:latest
```

2.2.9 Install CAIP AI Hub

1. Create litellm_config.yaml file by copy litellm_config.yaml.example file content and modify environment variables if needed
2. Create start.sh file by copy start.sh.example file content and modify environment variables if needed
3. Start the services

```
chmod +x start.sh
./start.sh
```

3. User Guides

3.1 Overview

This document aims to describe and provide detailed instructions for operations on the screens of the Chatbees system.

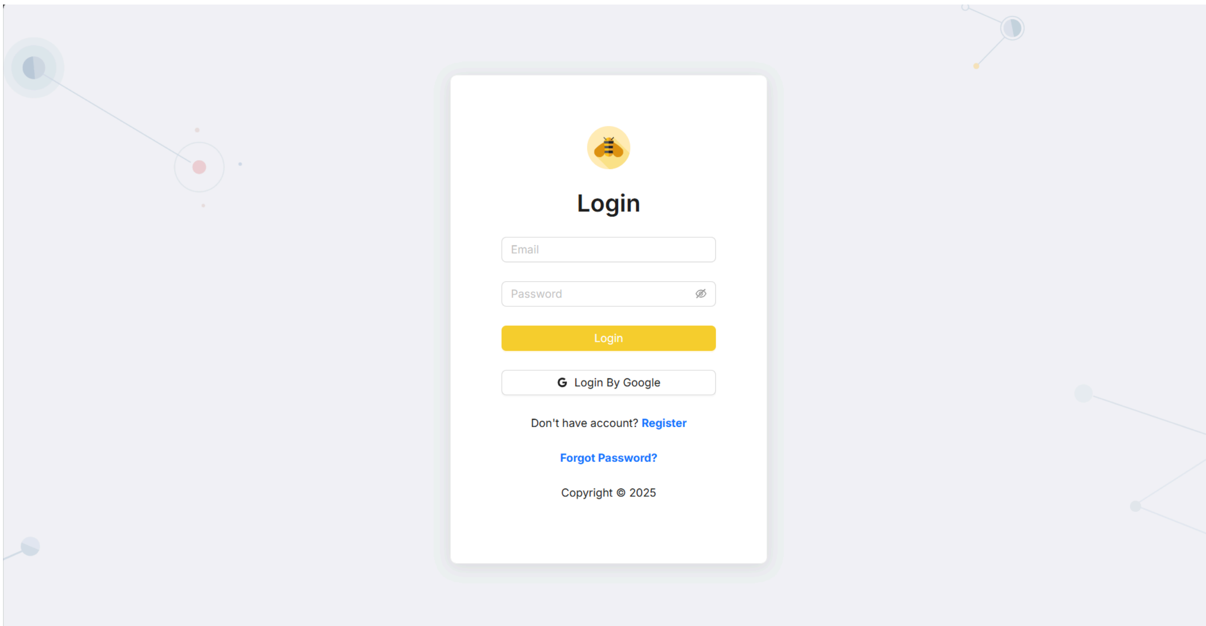
3.2. Authentication

3.2.1 Login

3.2.1.1 Login with System Account

Description: User want to log in to use the system

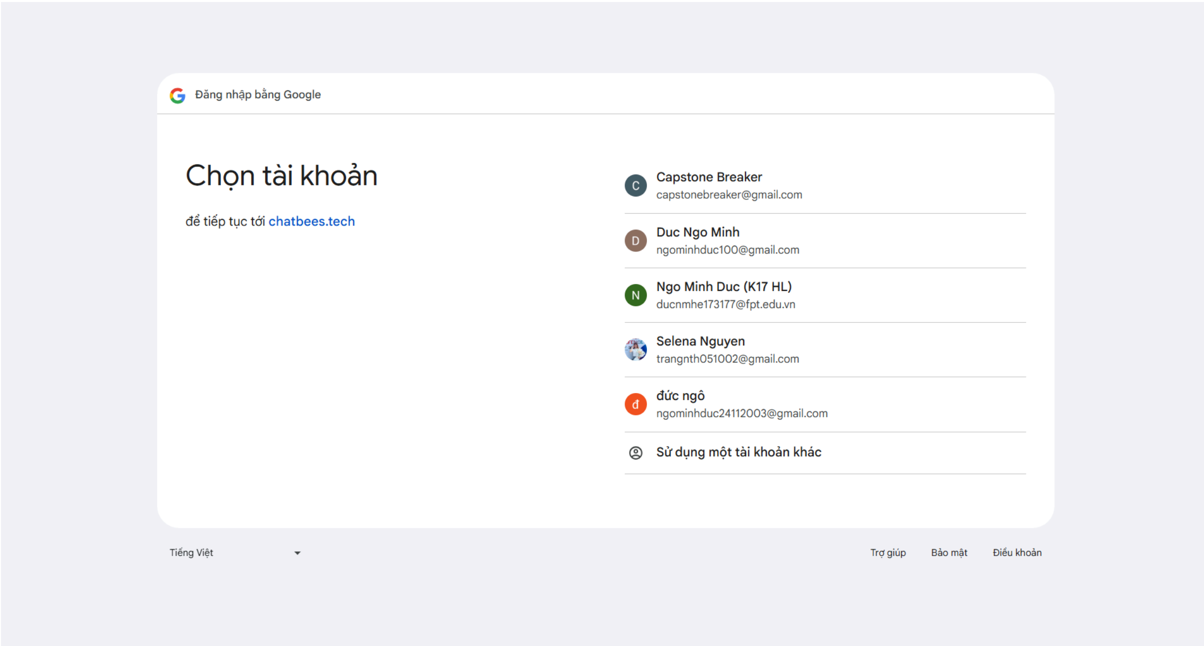
- User fill in their login information, including their registered "Email" and "Password".
- Click "Login"



3.2.1.2 Login with Google

Description: User wants to log in using their Google account.

- User clicks the “Login ByGoogle” button.
- The system will redirect to Google’s authentication page.
- After successful authentication, the user will be logged into the system.

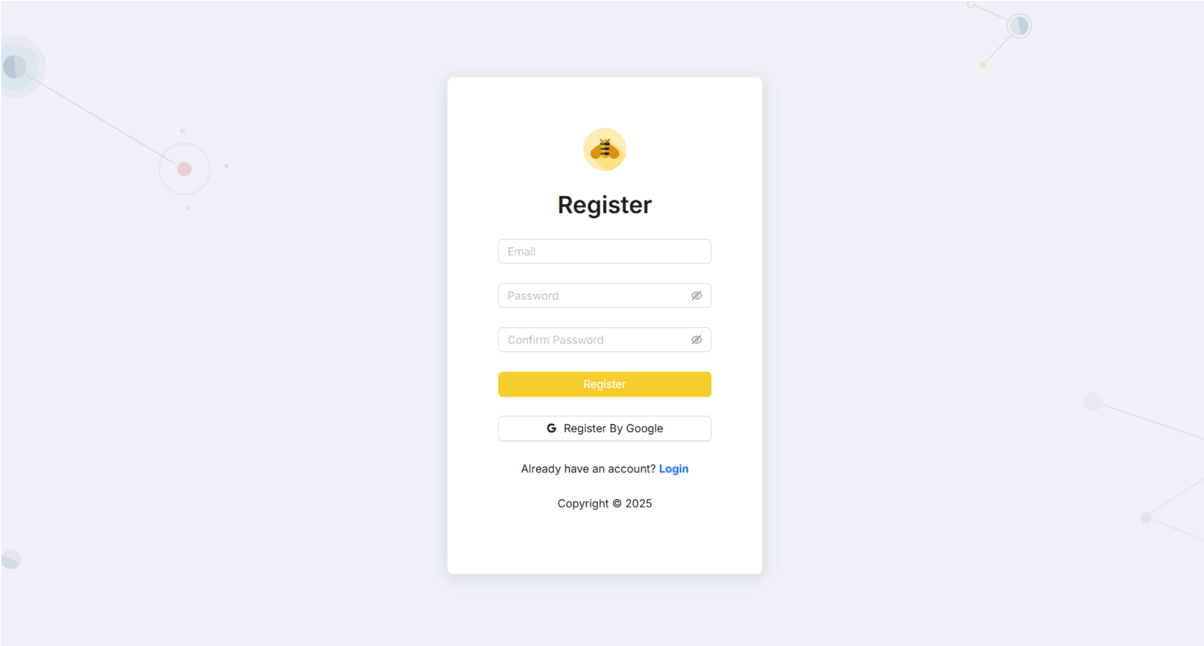


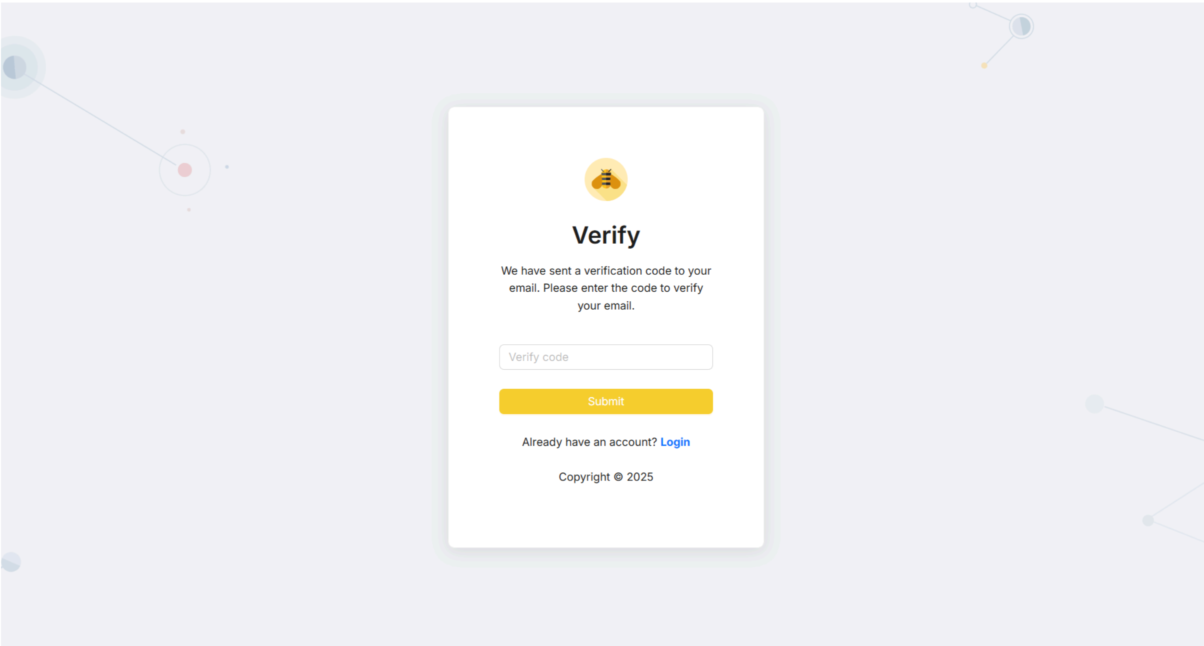
3.2.2 Register

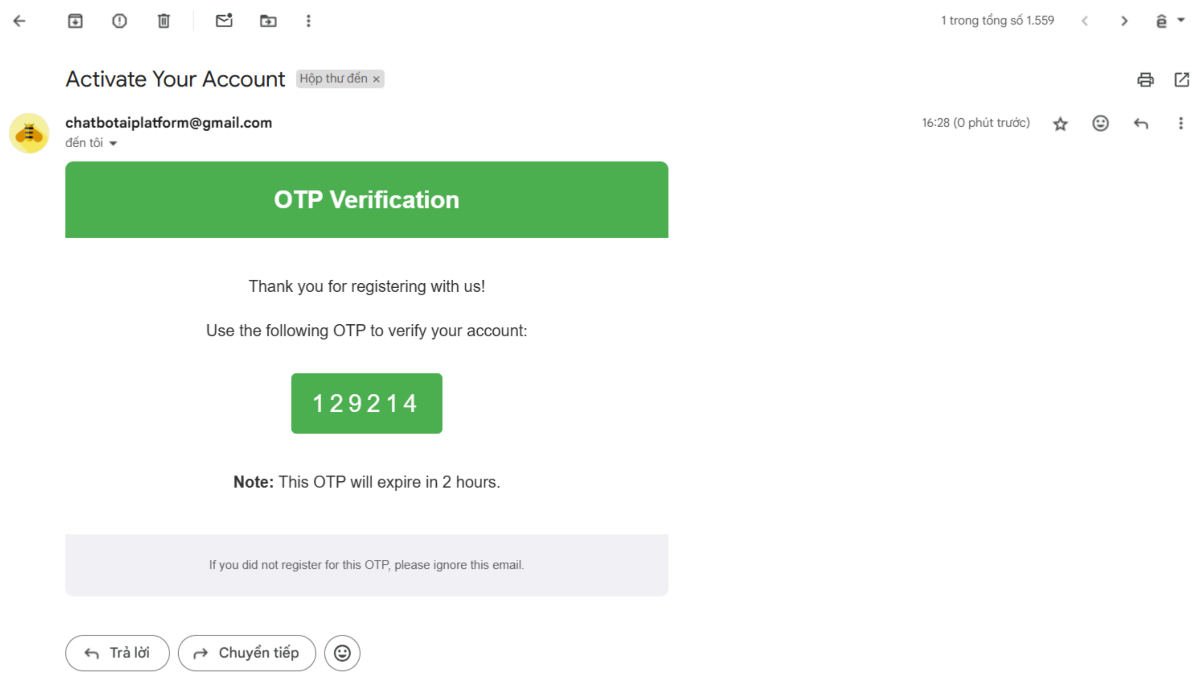
3.2.2.1 Signup with System Account

Description: User wants to use the system to register a new account in the user role.

- User fill in their login information, including their registered "Email", "Password" and "Confirm Password".
- Click the button “Register” on the Login page
- The system sends an OTP code to the user's registered email address.
- User enters the received OTP into the input field and clicks the **Submit** button to verify the account.
- When user is verified, they can login to use Chatbees


Register
Email
Password
Confirm Password
Register
Register By Google
Already have an account? Login
Copyright © 2025

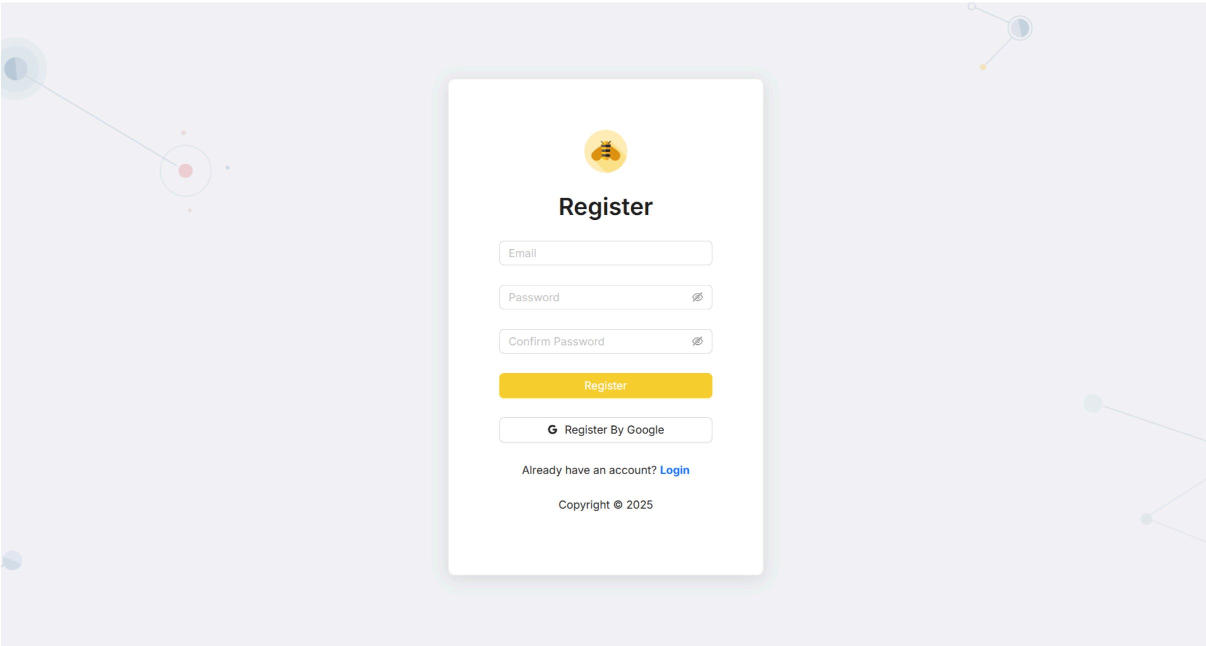

Verify
We have sent a verification code to your email. Please enter the code to verify your email.
Verify code
Submit
Already have an account? Login
Copyright © 2025

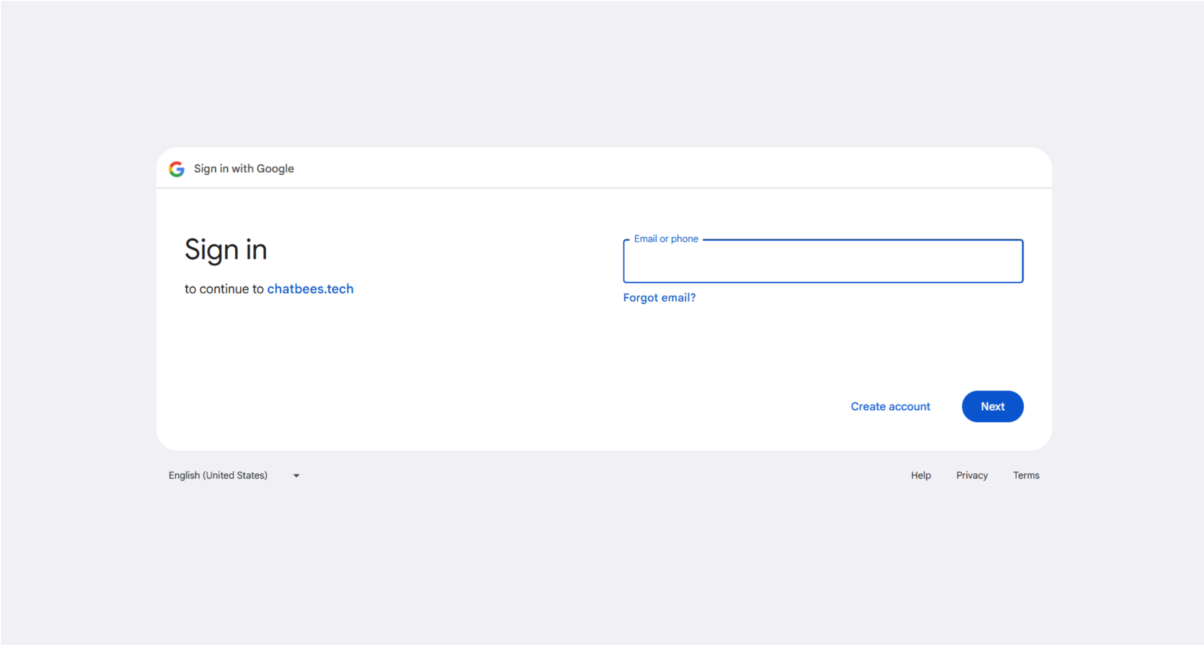


3.2.2.2 Signup with Google

Description: User wants to use google to register a new account in the user role.

- Click the button “Register By Google” on the Login page
- The system will redirect to Google’s authentication page.
- After successful authentication, the user will be logged into the system.

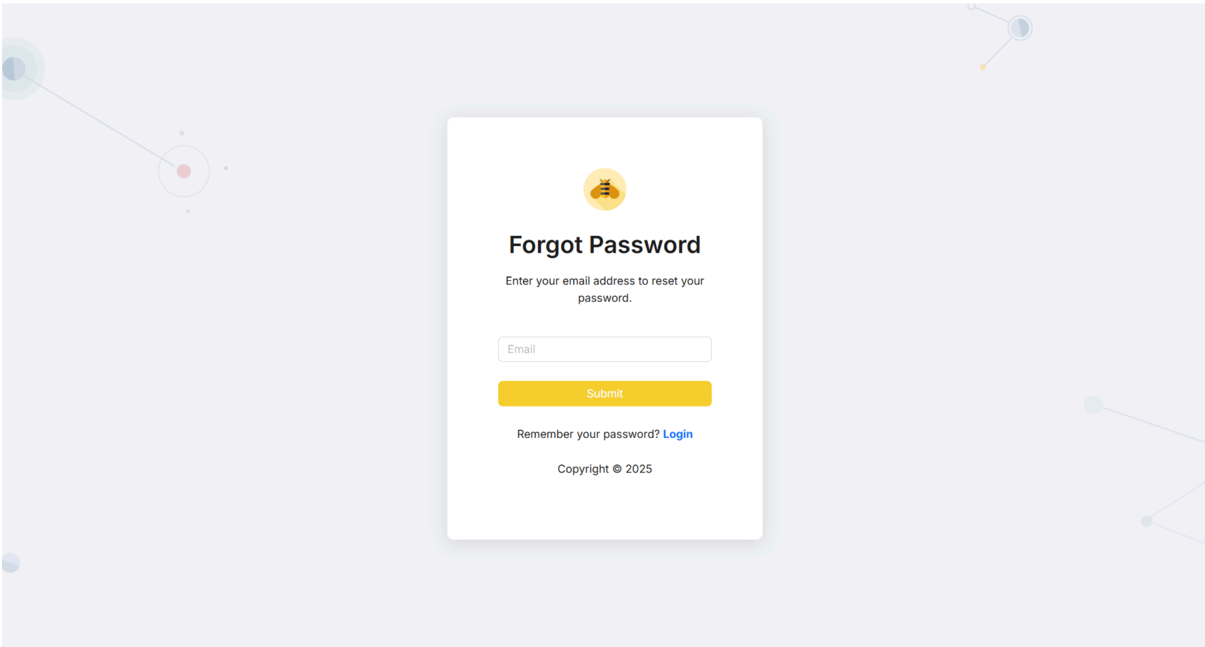




3.2.3 Forgot password

Description: User wants to reset their password because they forgot it.

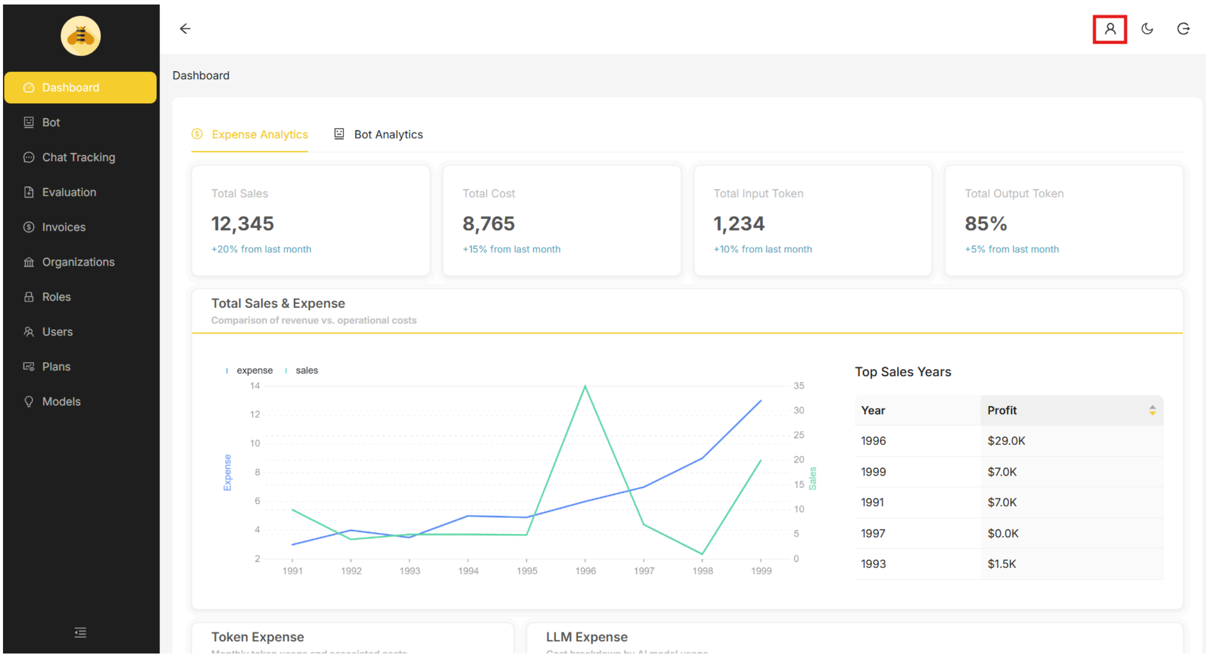
- From the Login page, the user clicks the "Forgot Password" link.

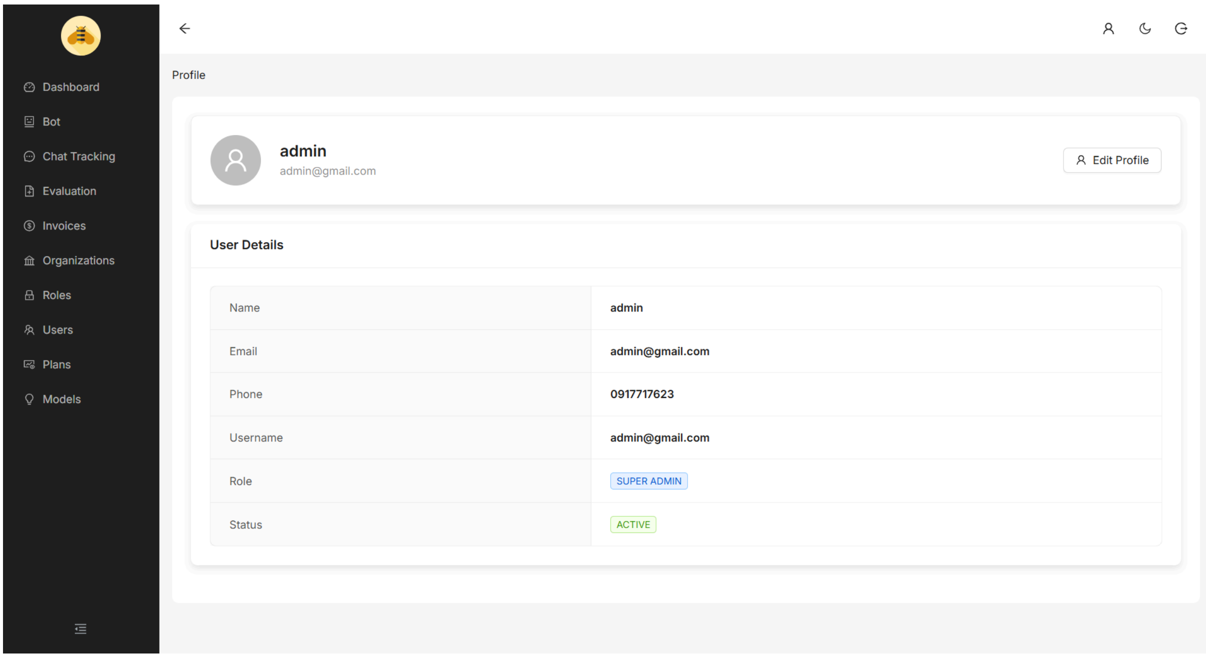


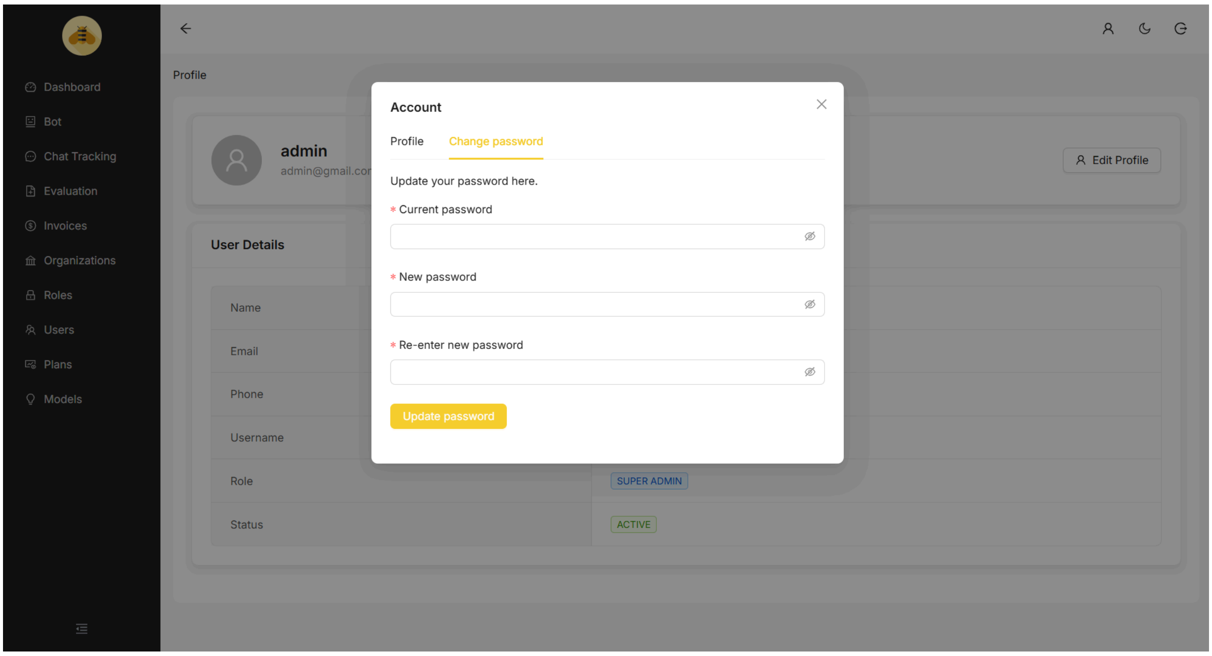
3.2.4 Change password

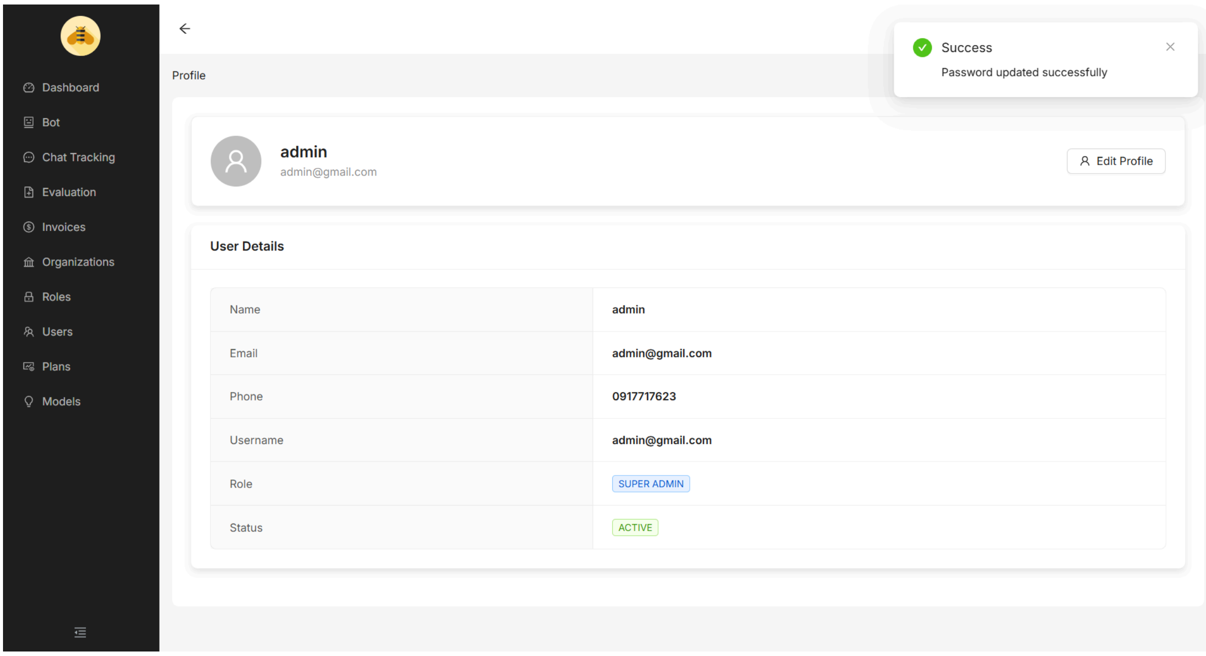
Description: User wants to reset their password because they forgot it.

- After logging into the system, the user is on the Home Page.
- User click on their Profile icon in the header bar.
- On the Profile page, the user clicks the "Edit Profile" button.





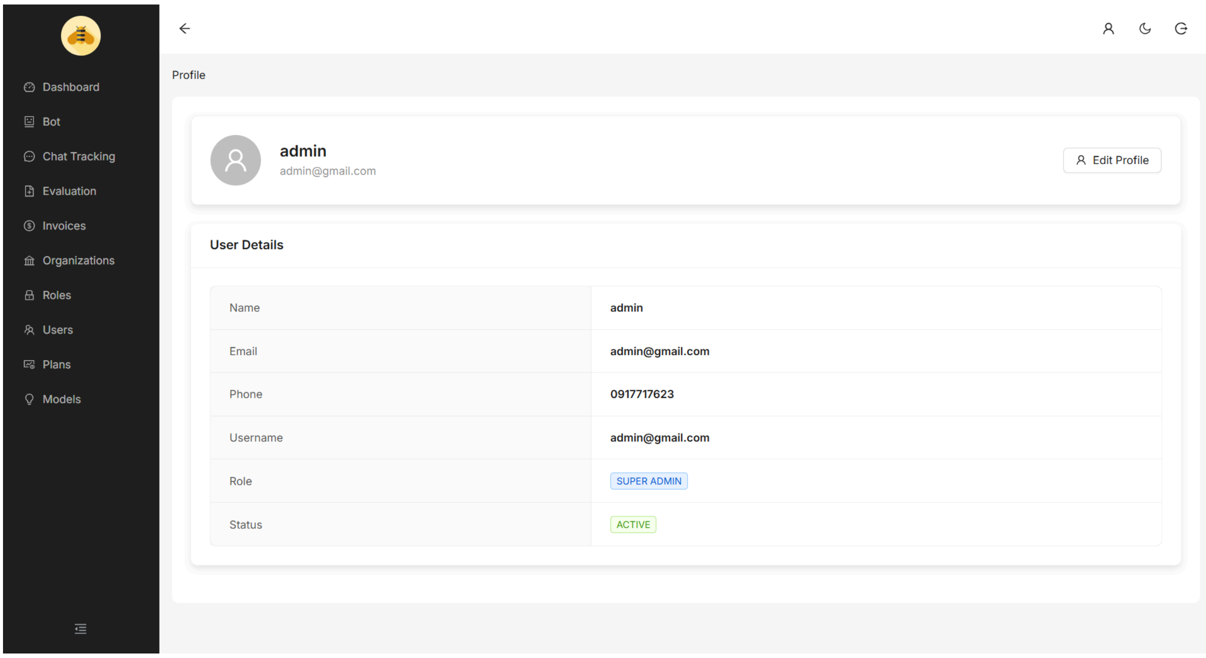


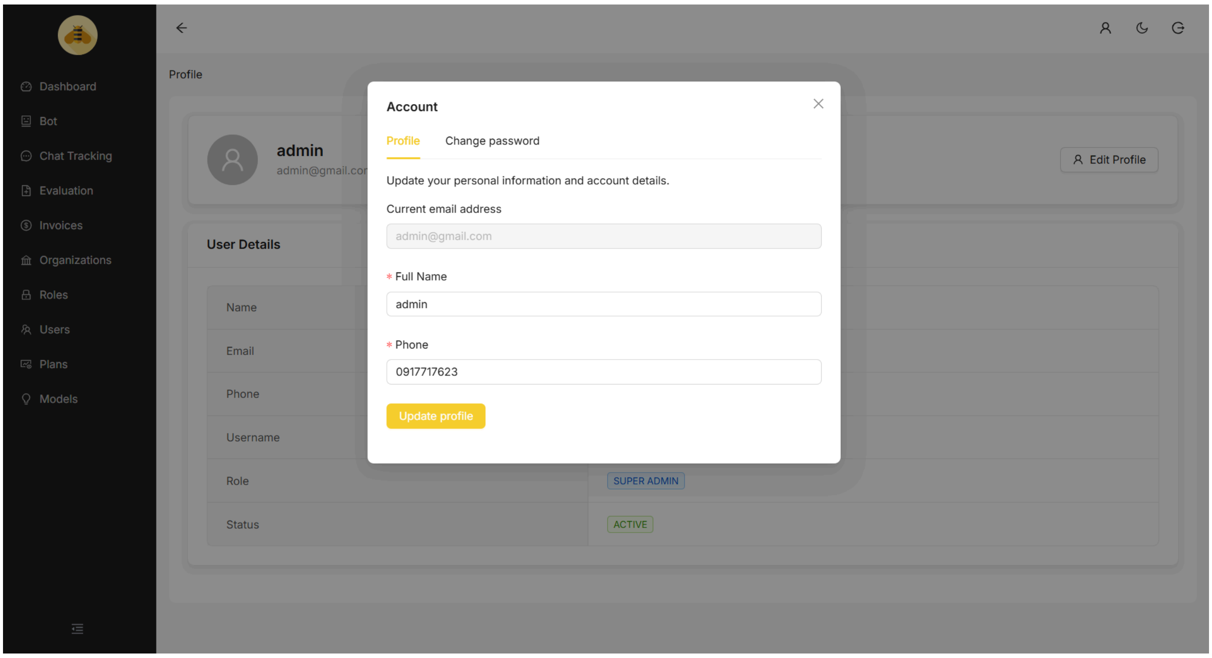


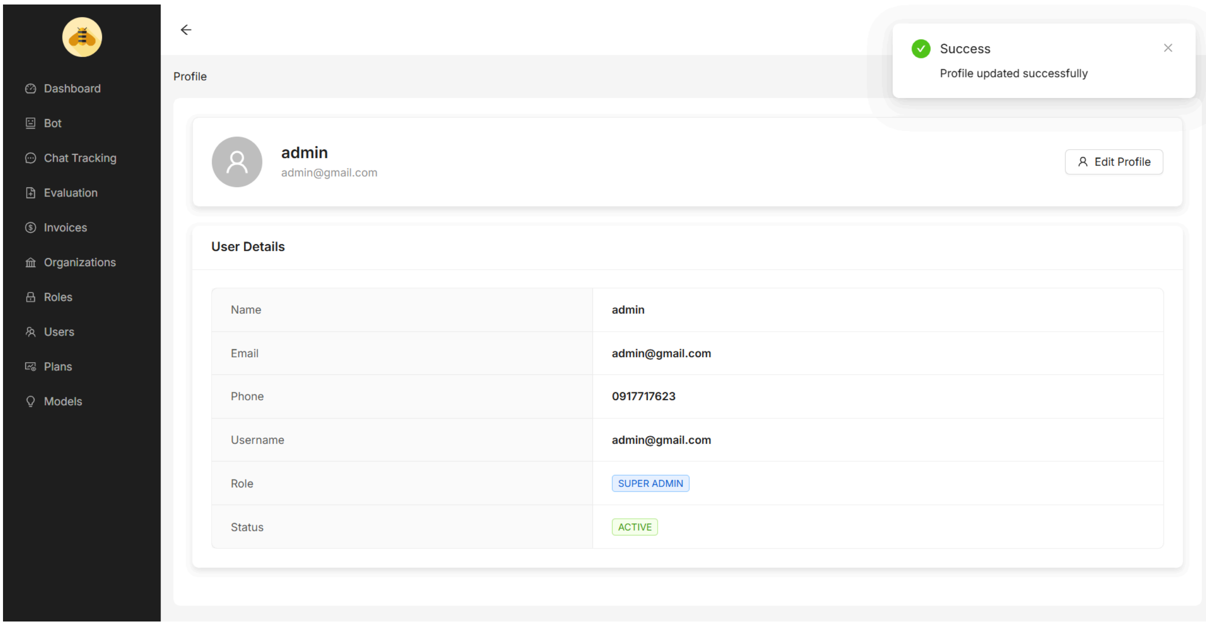
3.2.5 Change profile

Description: User wants to change their profile.

- After logging into the system, the user is on the Home Page.
- User clicks on their Profile icon in the header bar.
- On the Profile page, the user clicks the "Edit Profile" button.
- A popup menu appears, the user selects tab "Profile" from the popup menu.
- User updates information such as Full Name and/or Phone Number.
- Click the “Update profile” button to submit changes.
- The system updates the user's profile and shows a success message.


Dashboard
Bot
Chat Tracking
Evaluation
Invoices
Organizations
Roles
Users
Plans
Models
Profile
admin
admin@gmail.com
Edit Profile
User Details
Name
admin
Email
admin@gmail.com
Phone
0917717623
Username
admin@gmail.com
Role
SUPER ADMIN
Status
ACTIVE


Account
Profile
Change password
Update your personal information and account details.
Current email address
admin@gmail.com
* Full Name
admin
* Phone
0917717623
Update profile



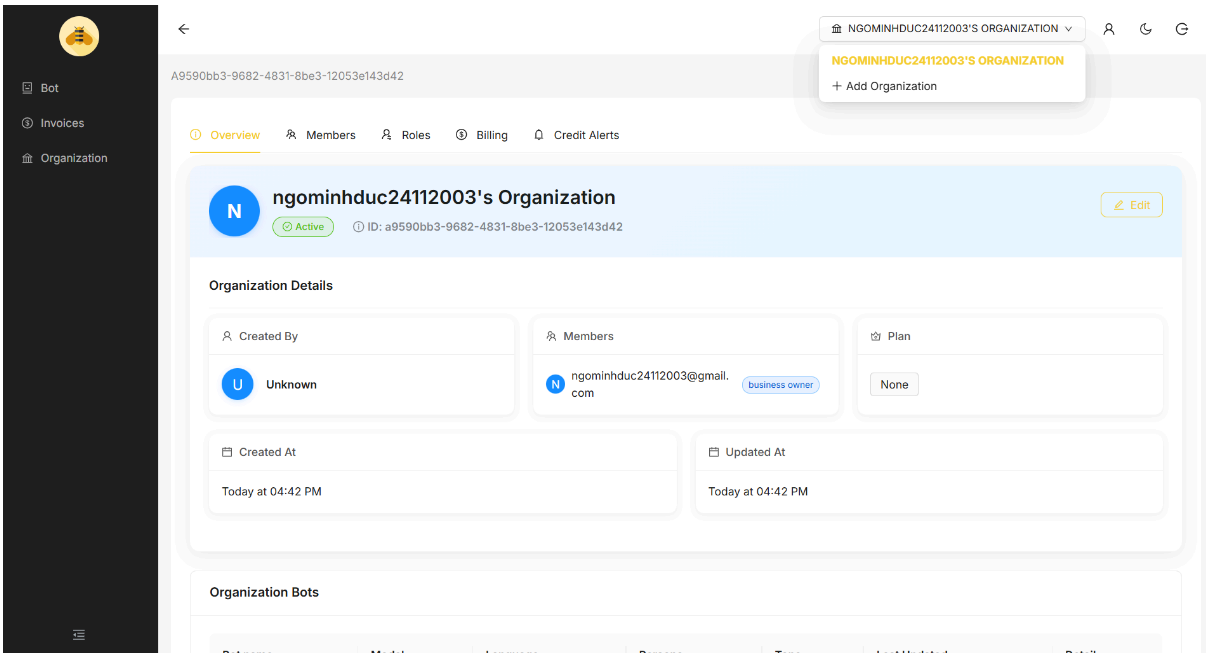
3.3. Setup Organization

3.3.1 Update plan for organization

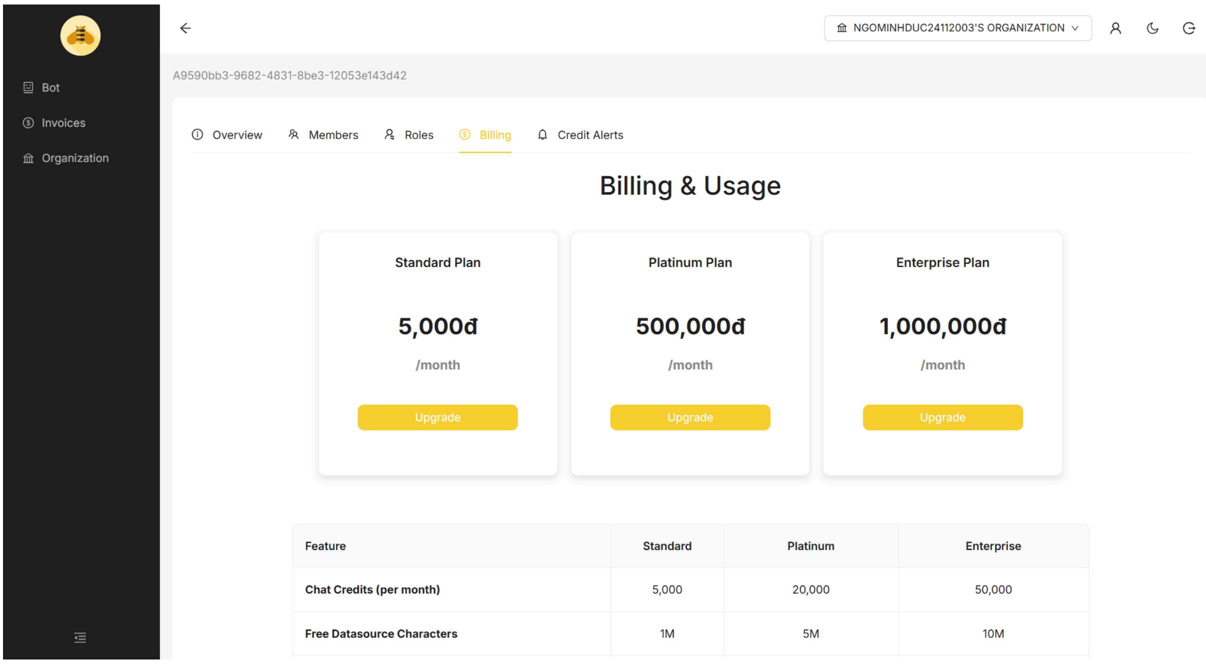
3.3.1.1 View Plan

Description: User wants to view all plans with a selected organization.

- After logging into the system, user is on the Home Page.
- User selects an Organization they want to upgrade the plan for from the organization list in the header.
- User selects an tab "Organization" form the sidebar.



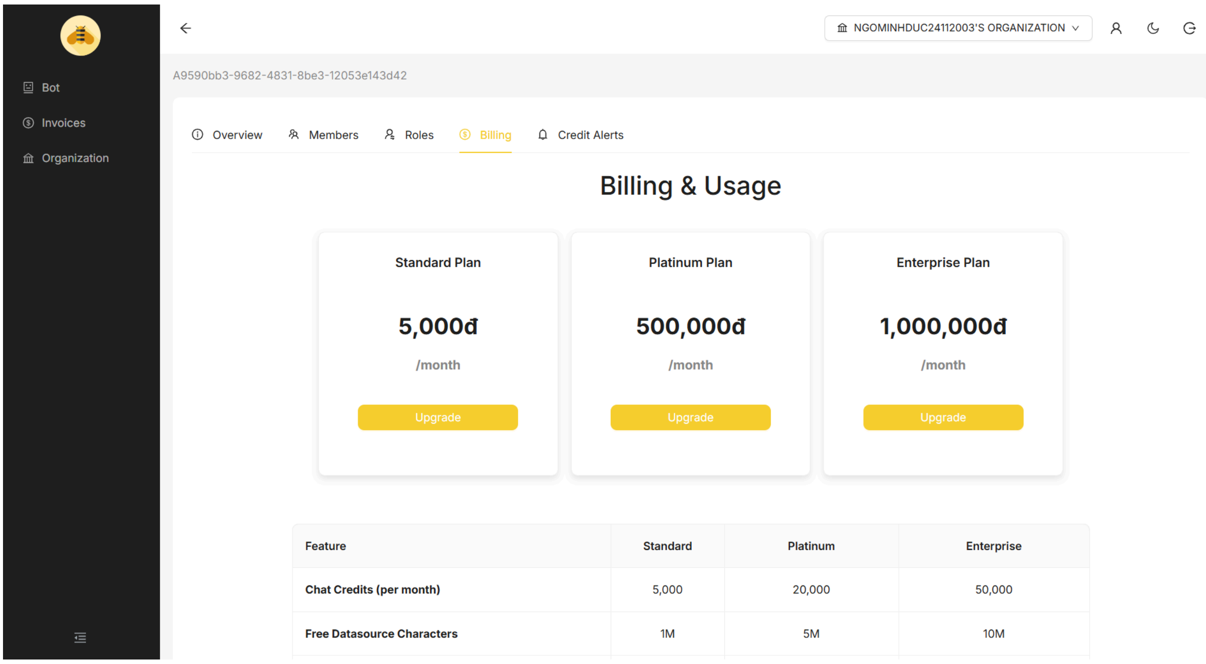
- After entering the selected Organization, user navigates to the "Billing" section.
- In the Billing section, user can view the current plan and available upgrade options.

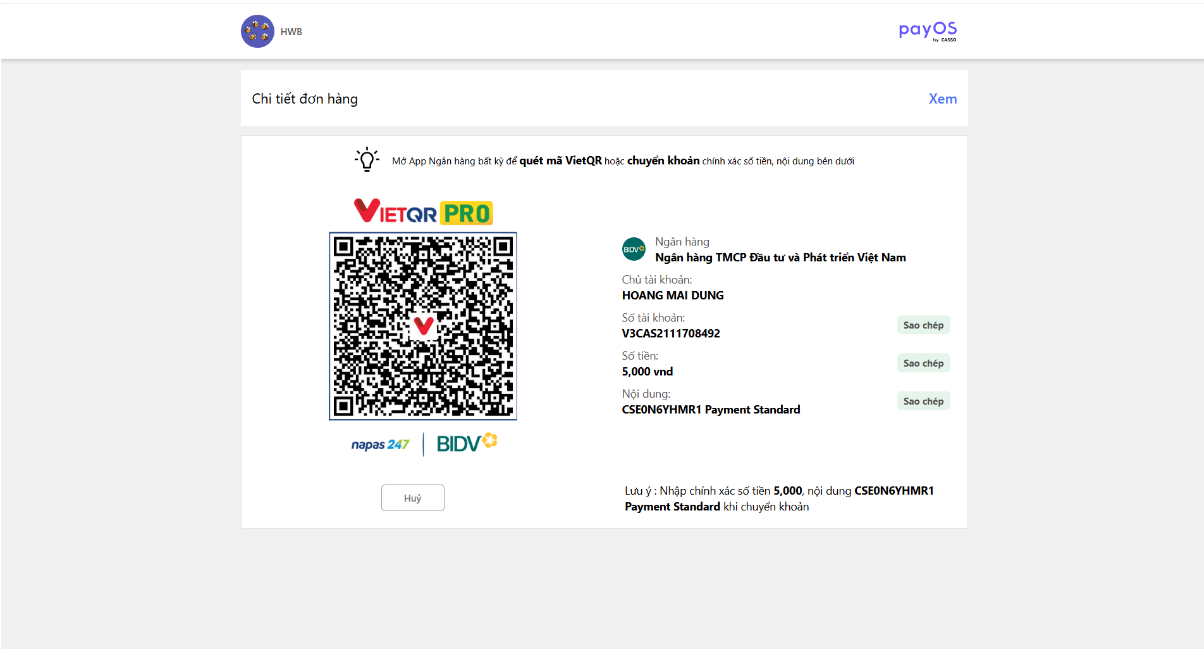


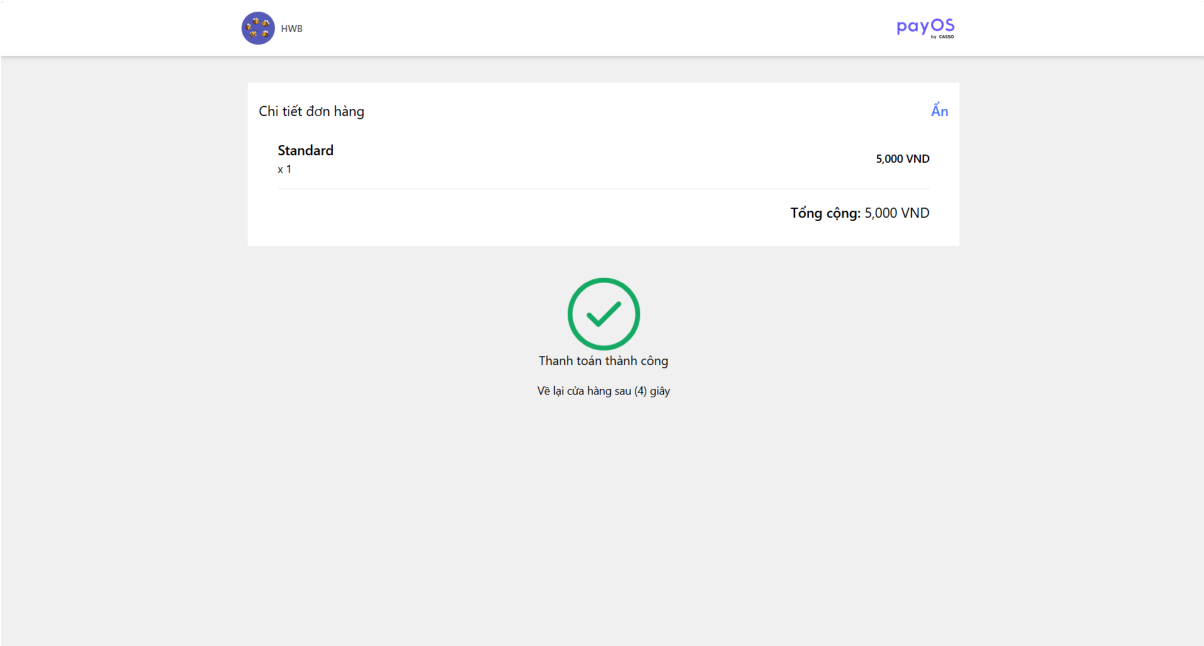
3.3.1.2 Upgrade plan

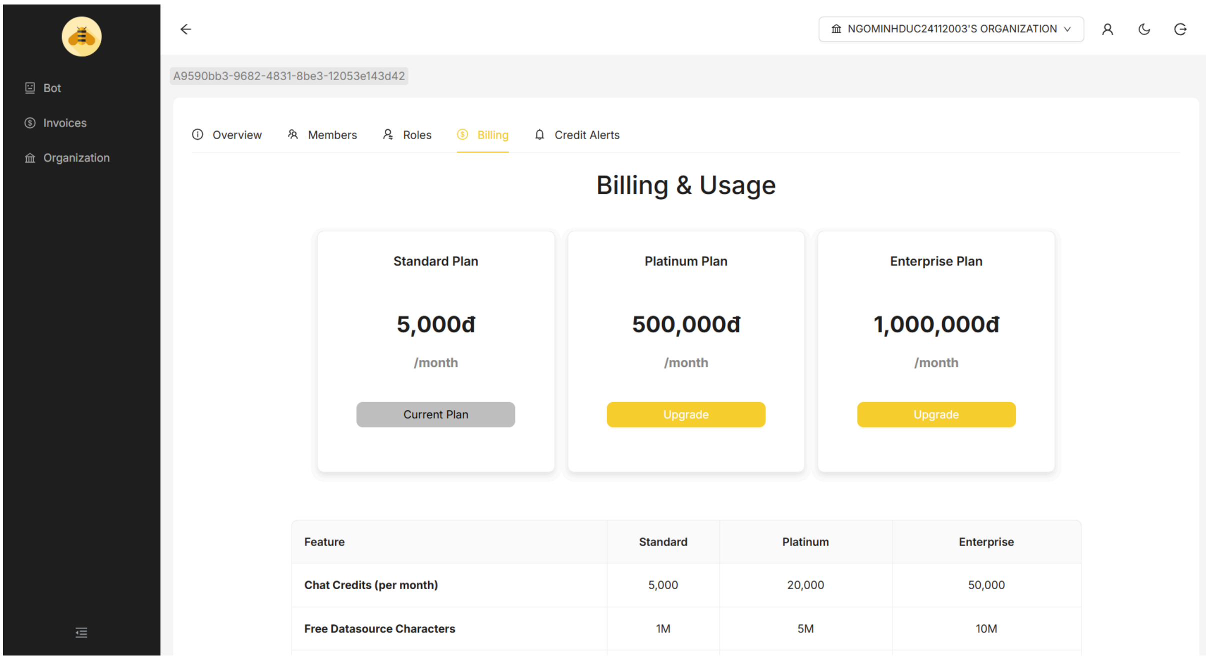
Description: User wants to upgrade their organization's plan.

- In the Billing section, user views available plans.
- User clicks the “Upgrade” button on the desired plan.
- The system redirects the user to the VietQR payment page.
- User scans the displayed QR code using their banking app to complete the payment.
- After successful payment, the system payment displays a "Payment Successful" confirmation page.
- Back to the Billing page, the new plan is now shown as the active plan for the organization.


NGOMINHDUC24112003'S ORGANIZATION
Bot
Invoices
Organization
A9590bb3-9682-4831-8be3-12053e143d42
Overview
Members
Roles
Billing
Credit Alerts
Billing & Usage
Standard Plan
5,000đ
/month
Upgrade
Platinum Plan
500,000đ
/month
Upgrade
Enterprise Plan
1,000,000đ
/month
Upgrade
Feature
Standard
Platinum
Enterprise
Chat Credits (per month)
5,000
20,000
50,000
Free Datasource Characters
1M
5M
10M


HWB
payOS
Chi tiết đơn hàng
Xem
Mở App Ngân hàng bất kỳ để quét mã VietQR hoặc chuyển khoản chính xác số tiền, nội dung bên dưới
VIETQR PRO
Ngân hàng
Ngân hàng TMCP Đầu tư và Phát triển Việt Nam
Chủ tài khoản:
HOANG MAI DUNG
Số tài khoản:
V3CAS2111708492
Sao chép
Số tiền:
5,000 vnd
Sao chép
Nội dung:
CSEON6YHMR1 Payment Standard
Sao chép
napas 247
BIDV
Huỷ
Lưu ý : Nhập chính xác số tiền 5,000, nội dung CSEON6YHMR1 Payment Standard khi chuyển khoản



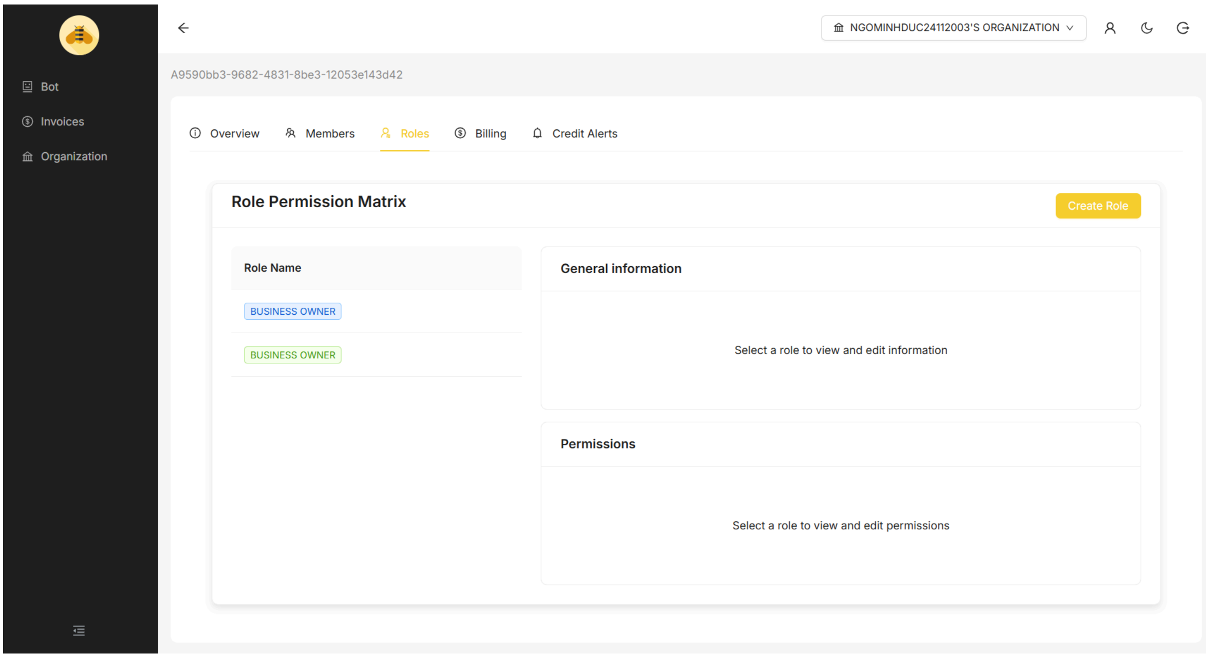


3.3.2 Add member for organization

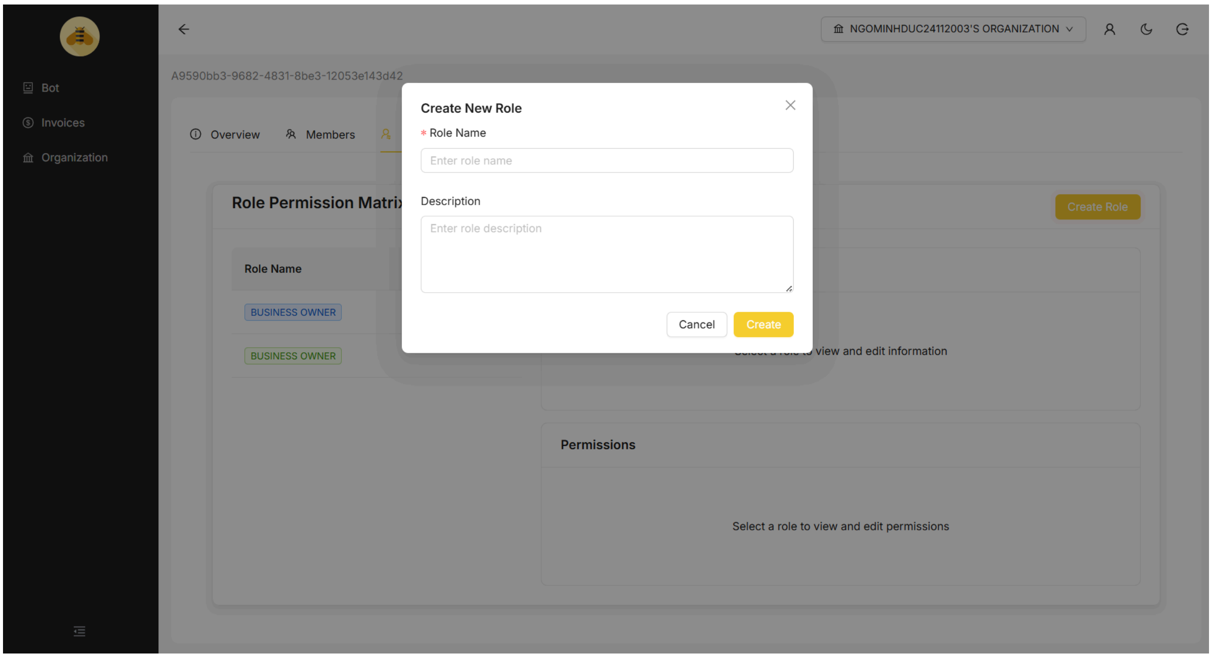
3.3.2.1 Create Role and Assign Permissions

Description: User wants to create a new role and assign permissions to that role.

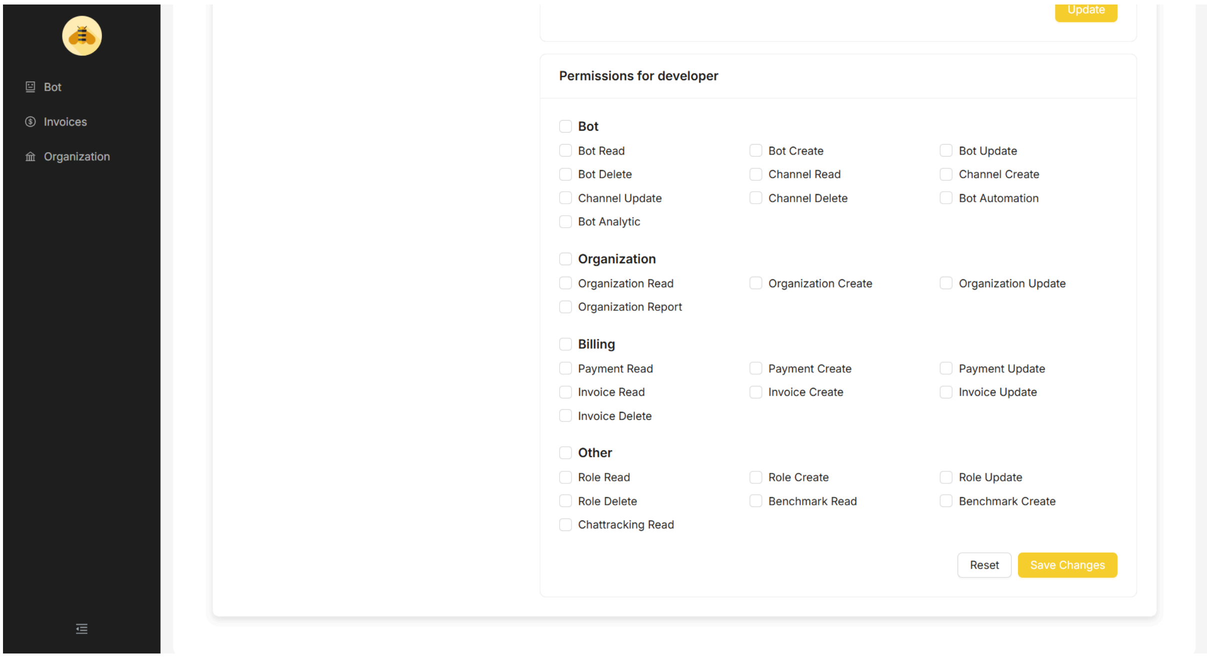
- After logging into the system and selecting an Organization, user navigates to the Roles section.



- User clicks “Create Role”.
- User fills in the Role Name and optional Role Description.
- User clicks “Save” to create the new role.



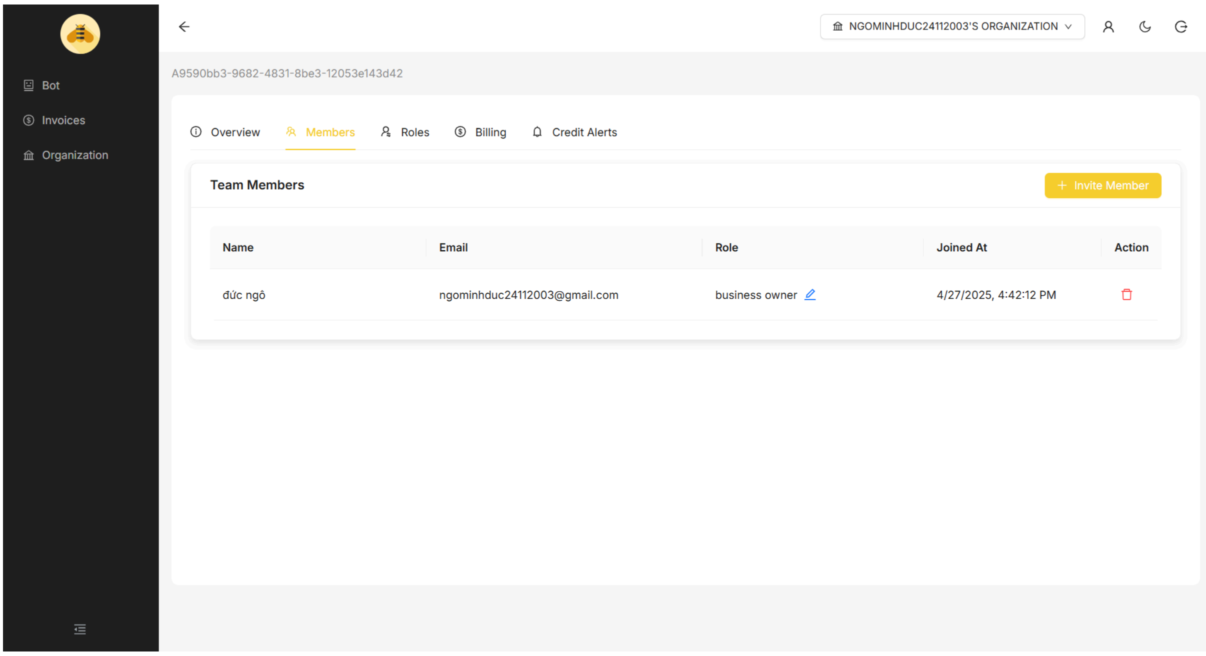
- After successful creation, the system redirects back to the List Roles page.
- User finds and clicks on the newly created Role from the list.
- On the Role Details page, user selects or configures the Permissions associated with this Role.
- User clicks “Save” to confirm the assigned permissions.

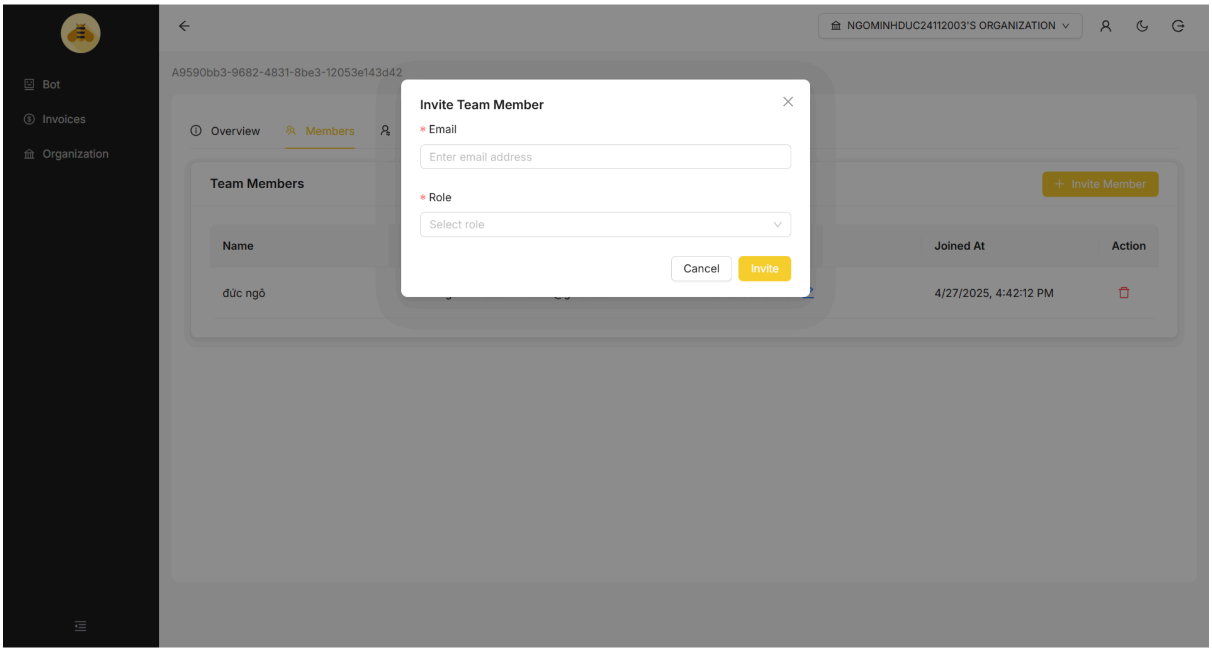


3.3.2.2 Invite Member to Organization

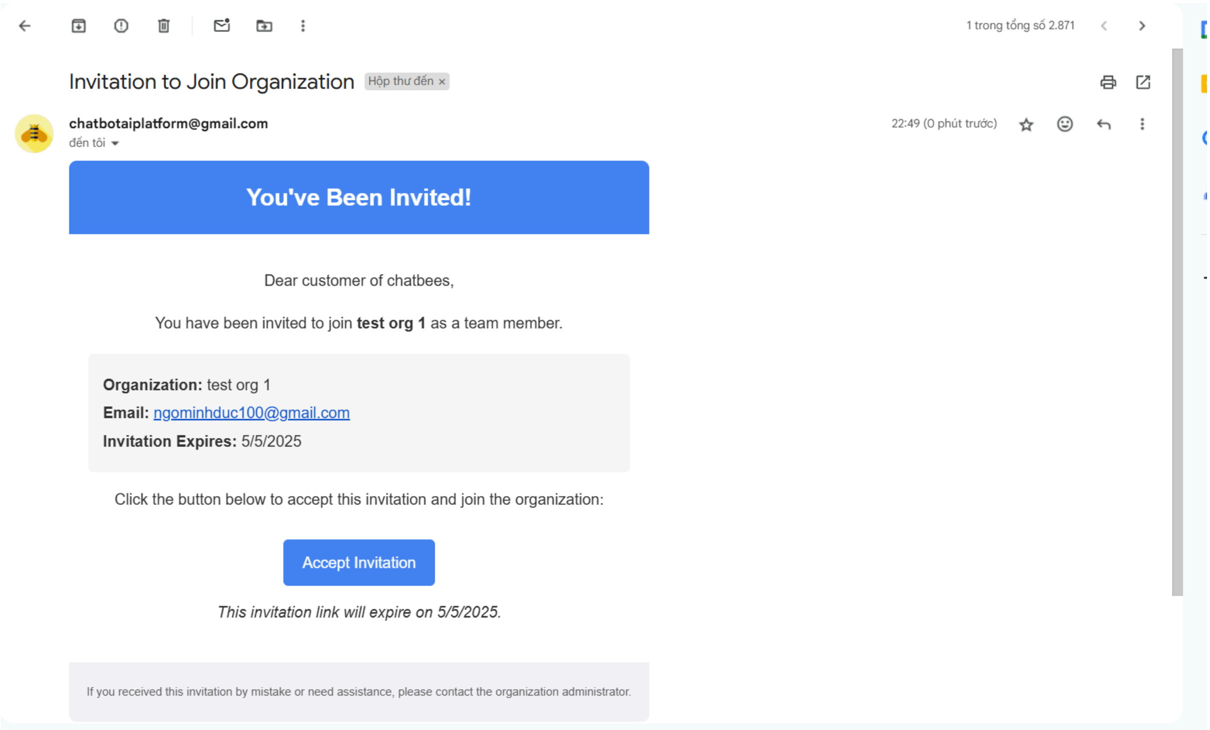
Description: User wants to invite new members into the organization and assign roles to them.

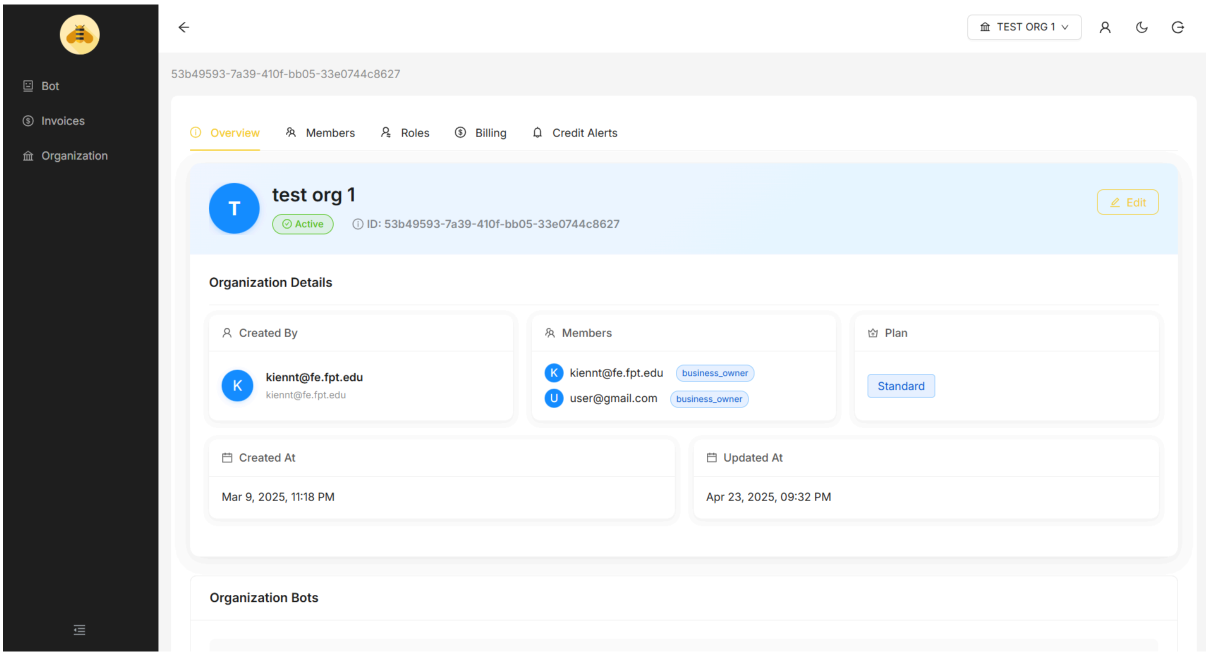
- After logging into the system and selecting an Organization, user navigates to the Members section.
- User clicks "Invite Member".
- User enters the email address of the member they want to invite and selects the Role for this member
- User clicks the "Invite" button.
- The system sends an invitation email to the provided email address with a link to join the organization.





- Once the member click “accept invitation”, their status is updated, and they are added to the list of members in the organization.
- The system navigates users to the new organization.



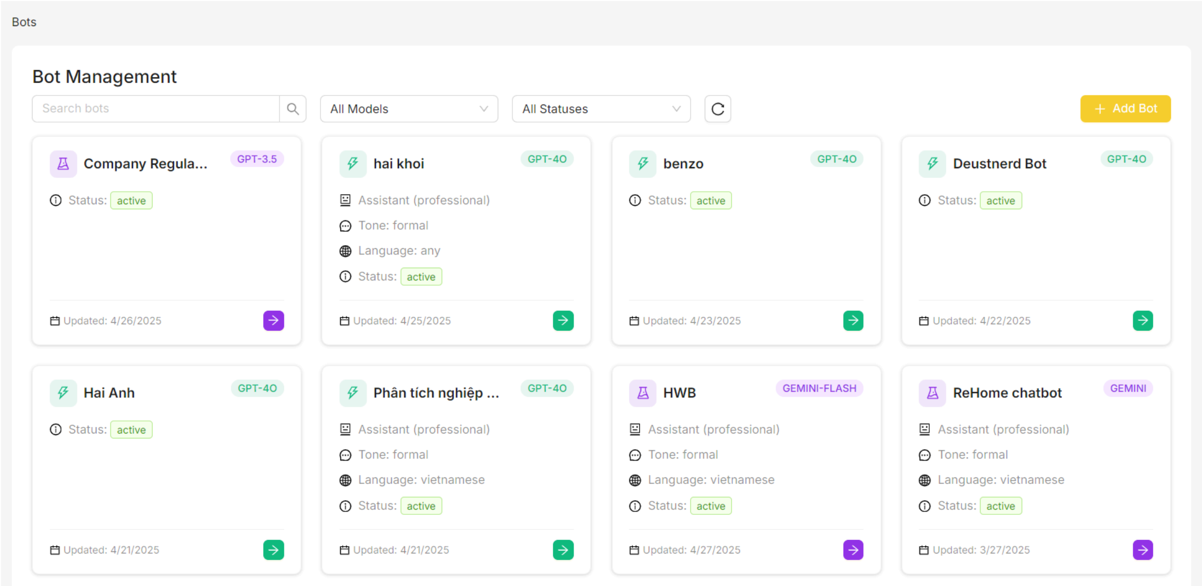


3.4. Config Bot

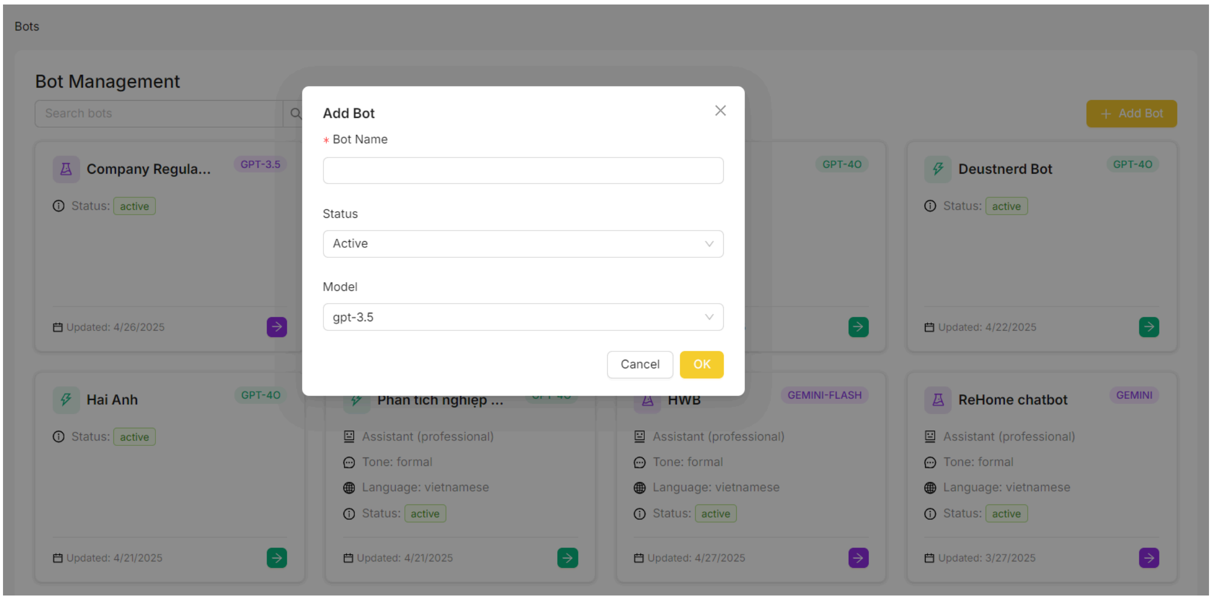
3.4.1 Create a new bot

Description: Create a new bot

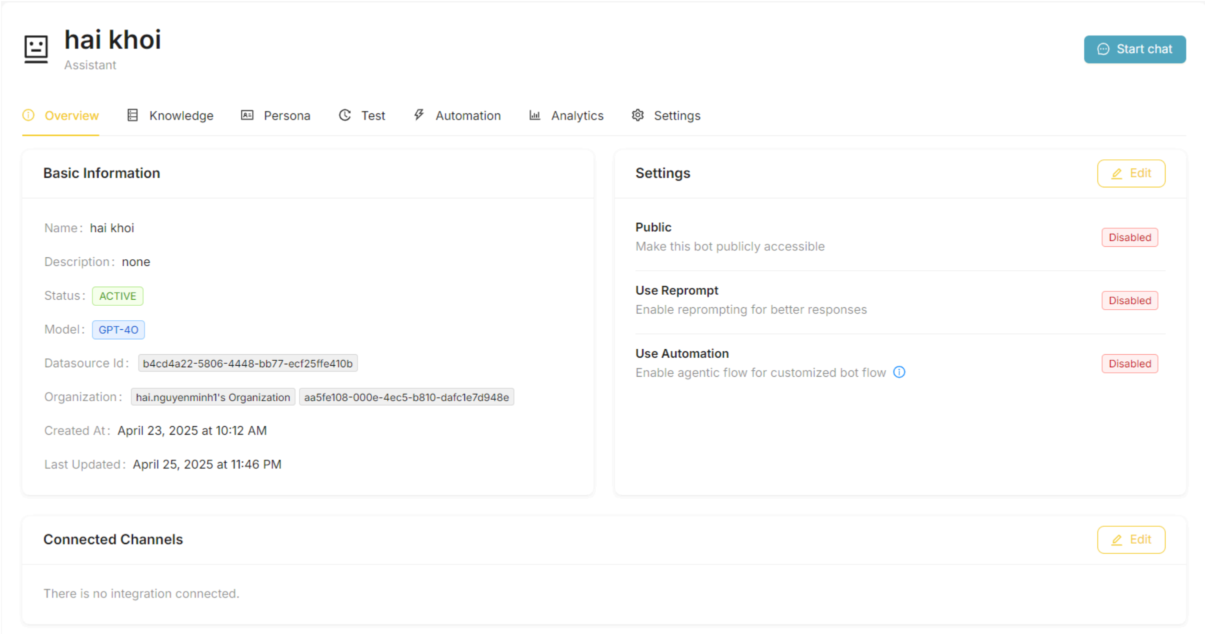
- Navigate to the Bot Management dashboard to view all your bots. Click the "Add Bot" yellow button in the top-right corner



- **Enter Bot Name**: Type your desired bot name in the required field
- **Set Status**: Choose "Active" from the dropdown to make your bot available immediately
- **Select Model**: Choose AI model (GPT-3.5, GPT-4.0, GEMINI, etc.) based on your needs



- After creation, the system take you to your bot overview page



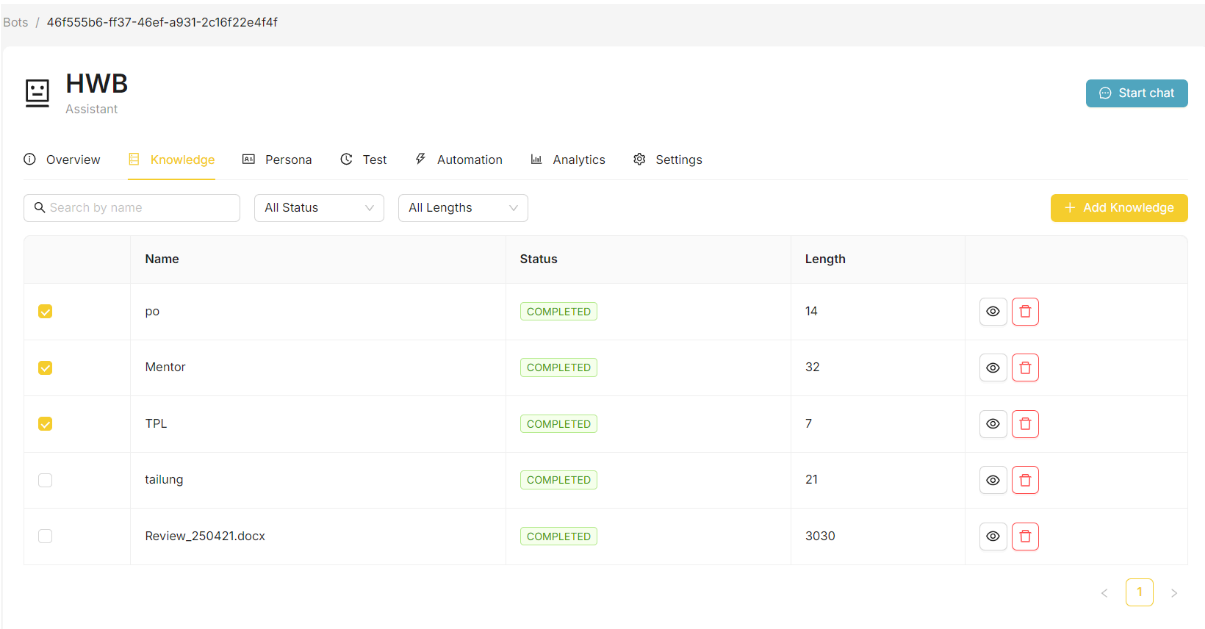
3.4.2 Config bot settings

3.4.3 Config prompt

3.4.4 Config bot knowledge

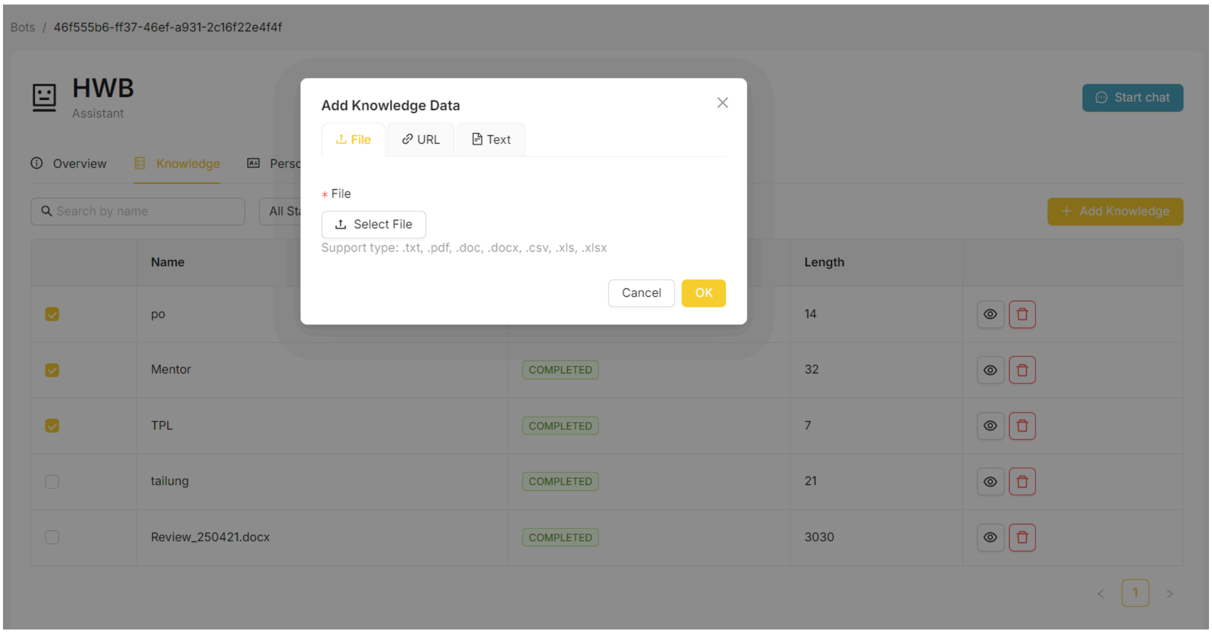
Description: Adding and managing knowledge sources for your bot. Properly configured knowledge lets your bot provide accurate information based on your specific content.

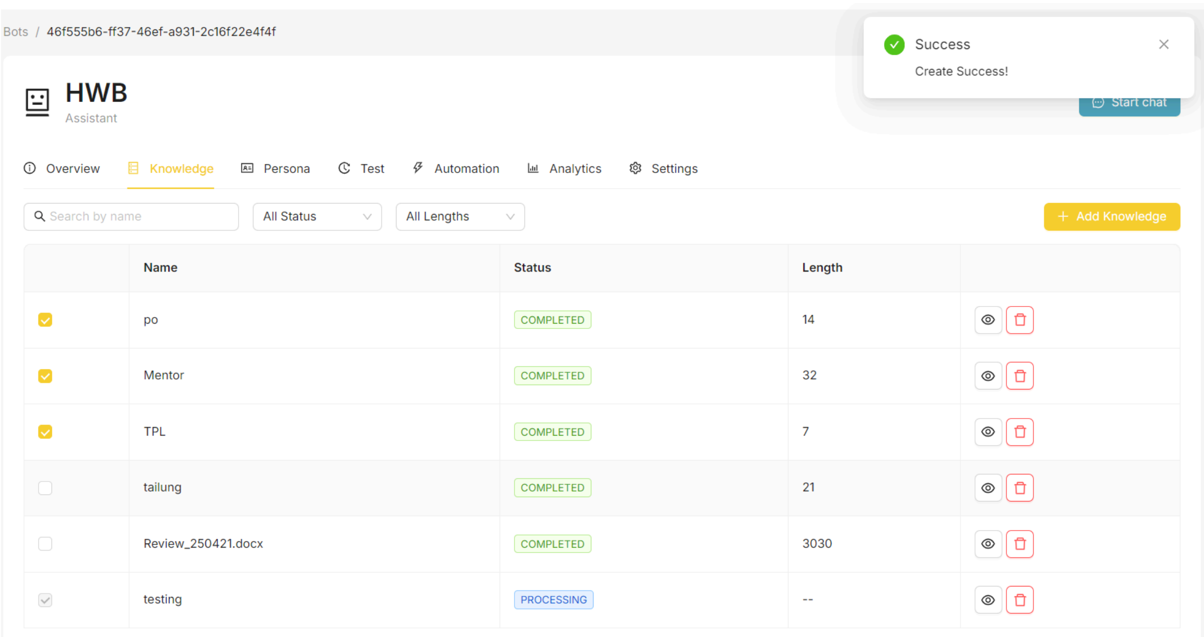
- Navigate to the Knowledge tab to view list knowledge page



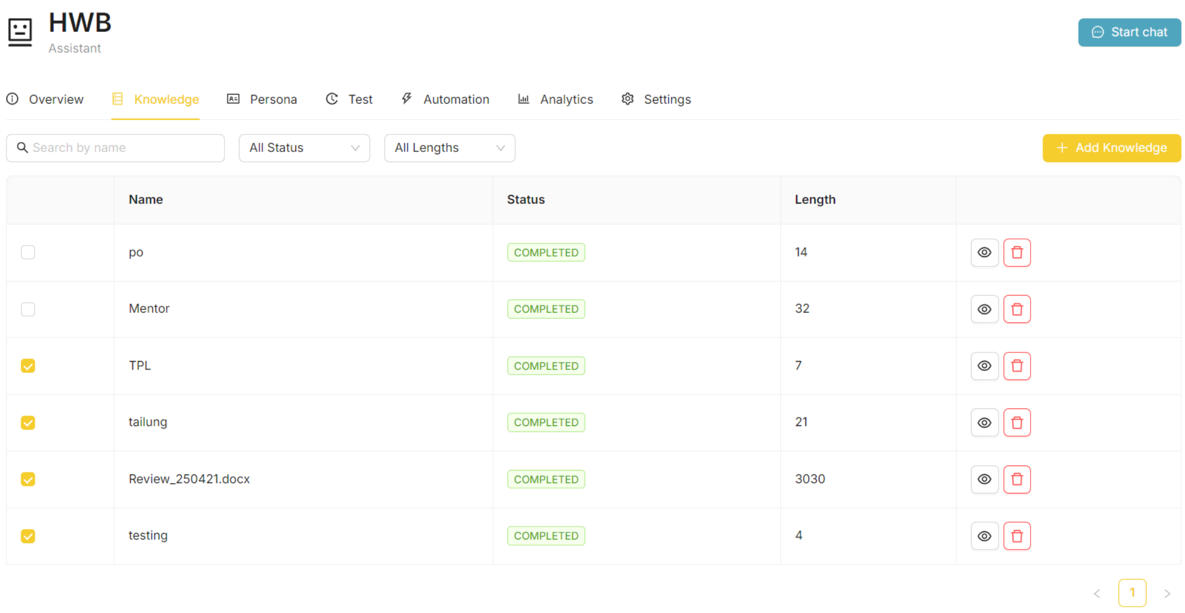
- The user can add new knowledge to the bot by clicking the "Add Knowledge" button in the top-right corner
- Choose your knowledge source type:
    - File: Upload documents (.txt, .pdf, .doc, .docx, .csv, .xls, .xlsx)

- URL: Add website content
- Text: Enter text directly





- Click the checkbox in the leftmost column next to any knowledge source you want to activate



3.4.5 Config automation

3.4.6 Config analytic

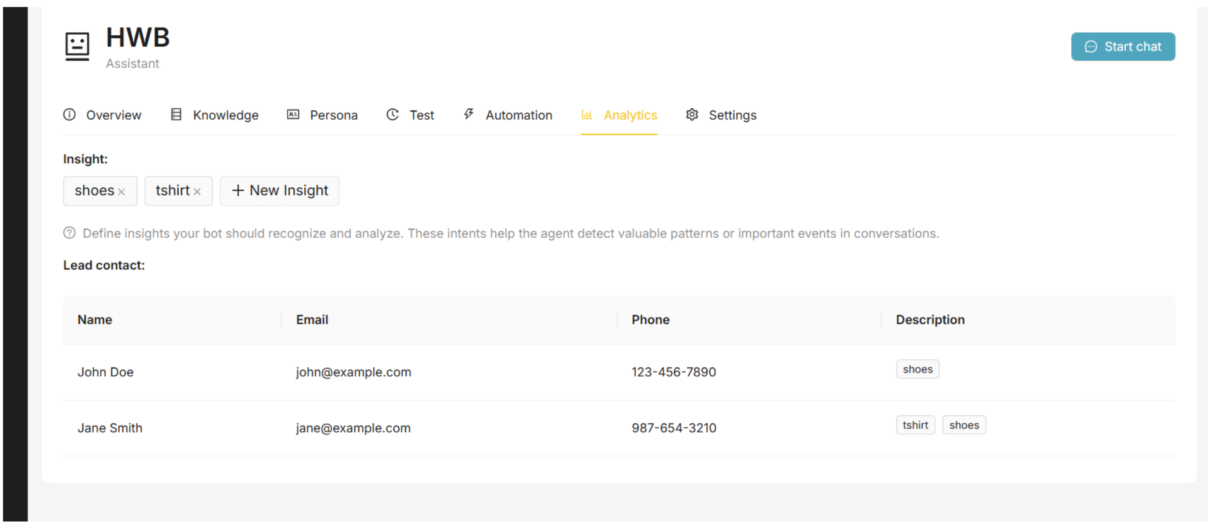
Description: The User wants to view previous bot testing conversations

- In the bot detail page, click on the test tab.
- The user will then have a list of previous bot testing conversations

Description: The user wants to define insights that the bot should recognize and analyze. These insights help the agent detect valuable patterns or important events in conversations.

In the bot detail page, under the "Insight" section, click on the "+ New Insight" button.

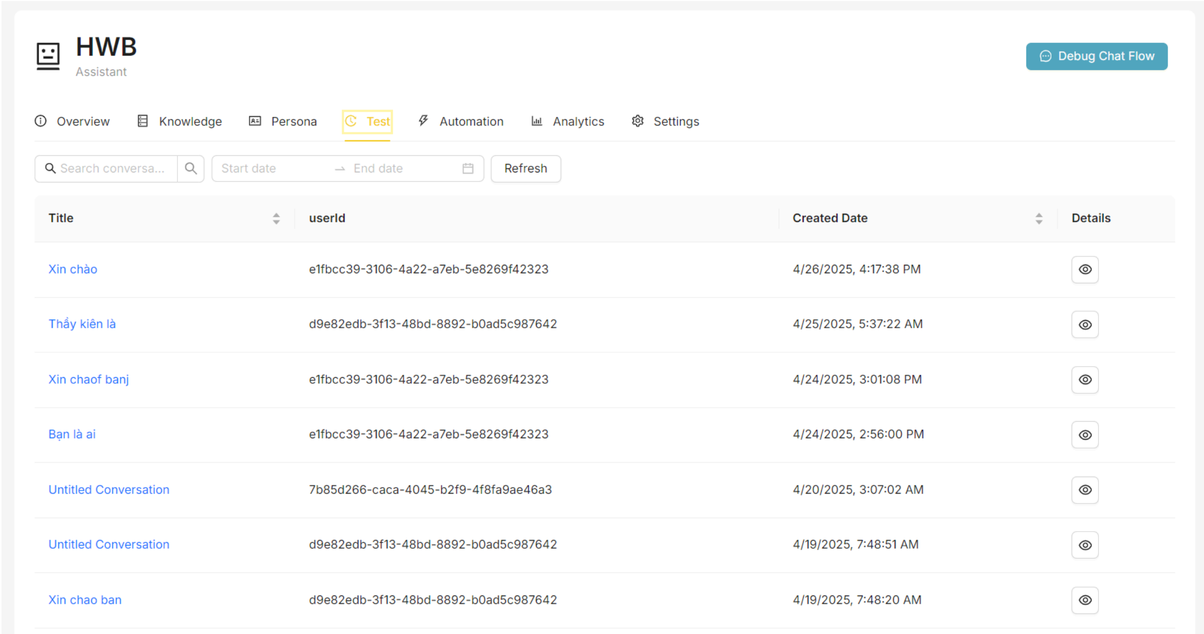
The user can then define a new insight by entering its name and relevant keywords.



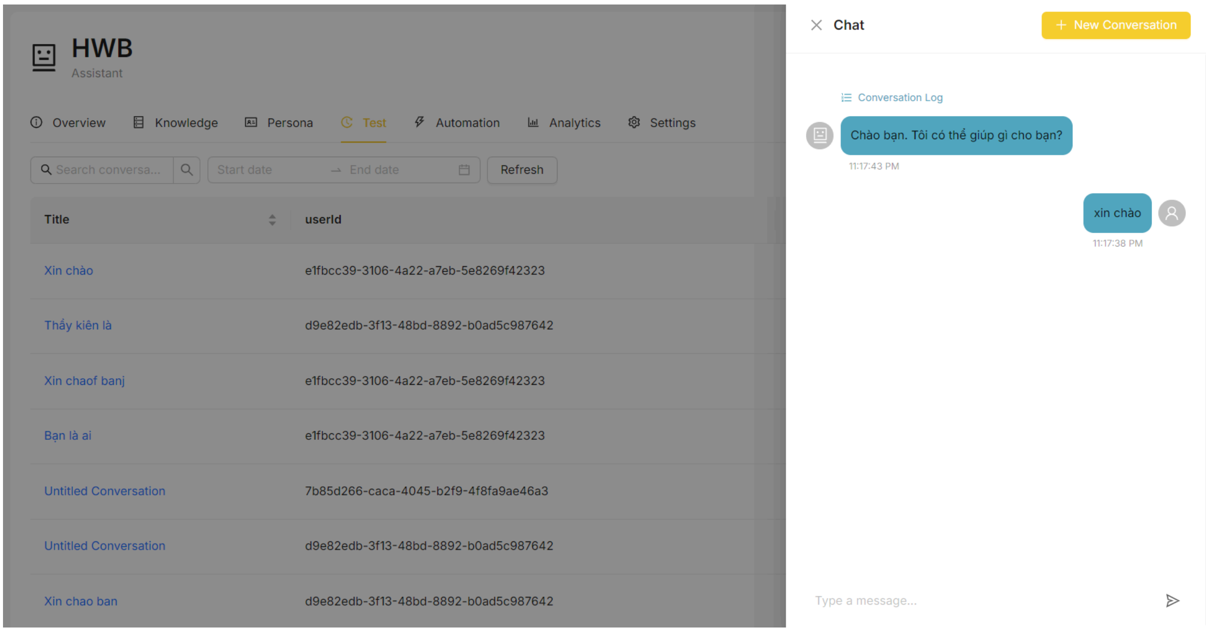
3.4.7 View test bot history

Description: The User wants to view previous bot testing conversations

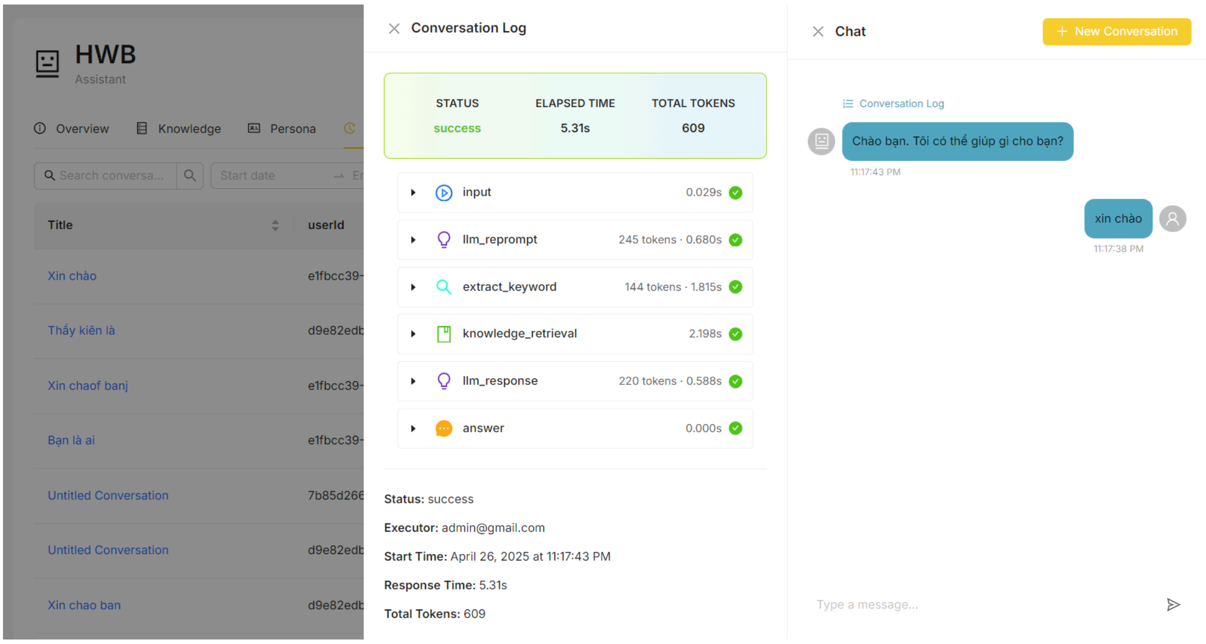
- In the bot detail page, click on the test tab.
- The user will then have a list of previous bot testing conversations



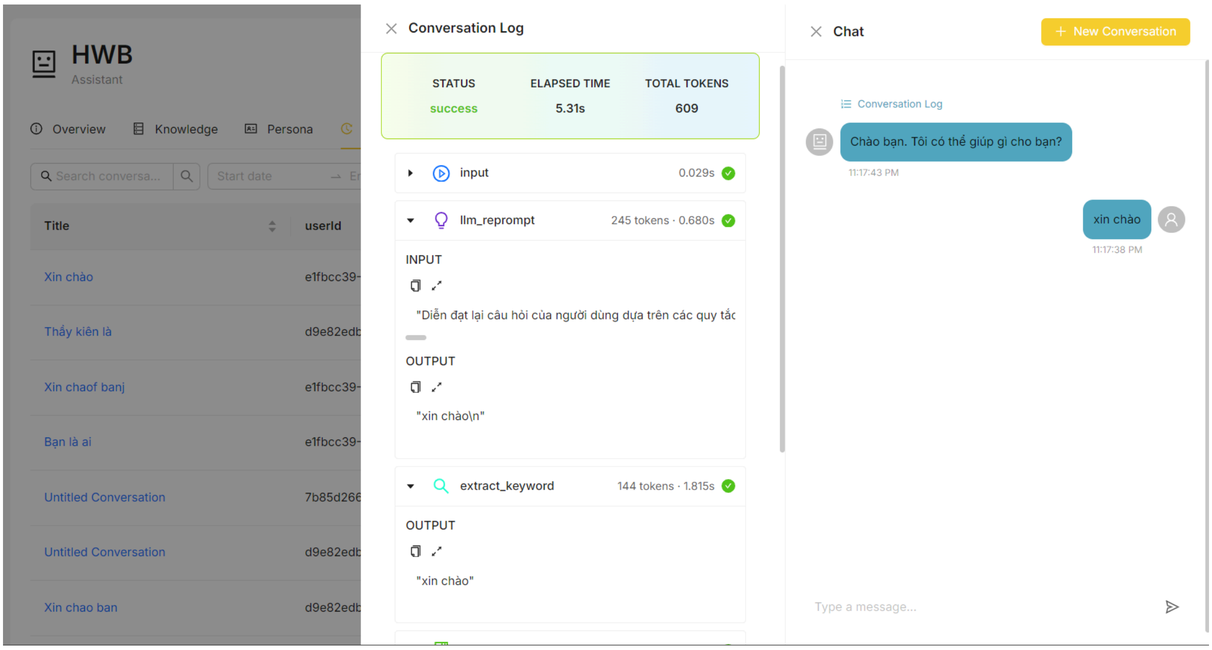
- Click on one of the conversations to preview its message content



- To view insight on how the response was made, click on the “conversation log” button, and it will display information such as response time, token usages, and in in-depth workflow of the bot



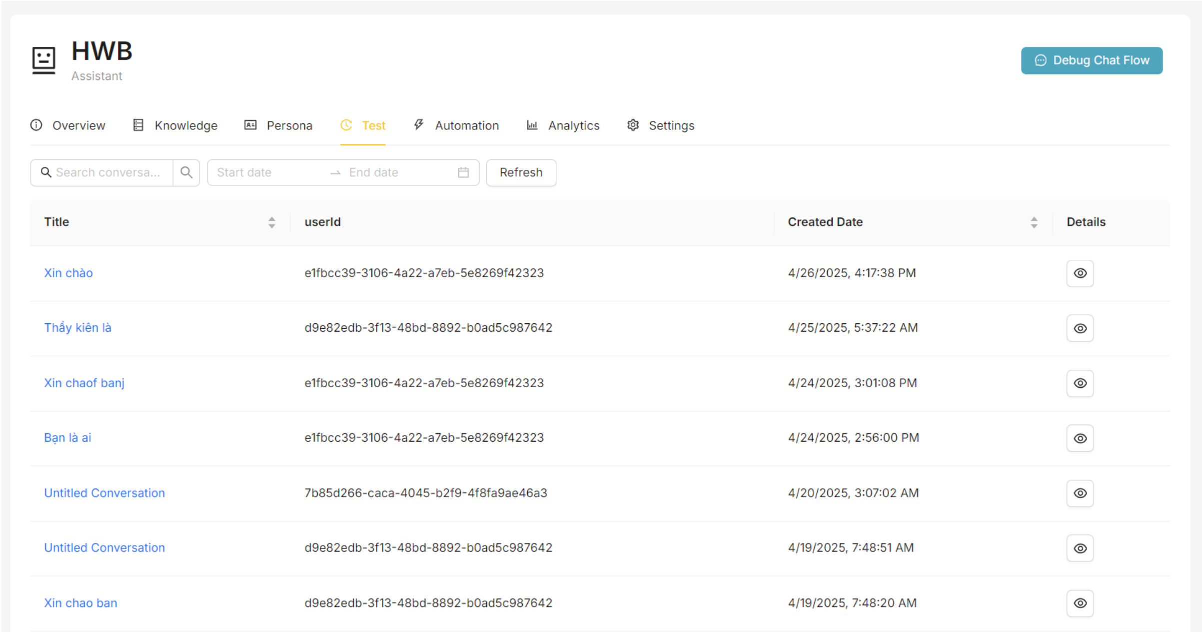
- From here, you can see the input and output of each step, you can check your prompt to see how the bot responds to the prompt, how the bot extracts keywords from previous messages, or which document the bot retrieves to answer the question.



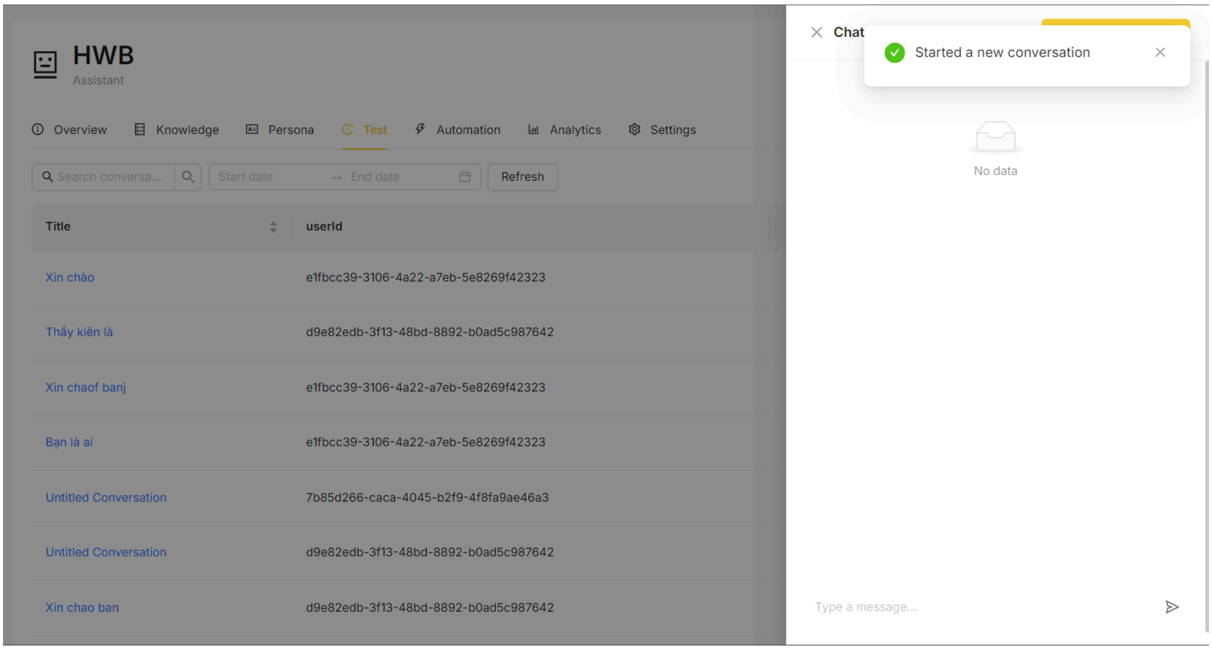
3.4.8 Create a test for bot conversation

Description: The User wants to create a new testing conversation

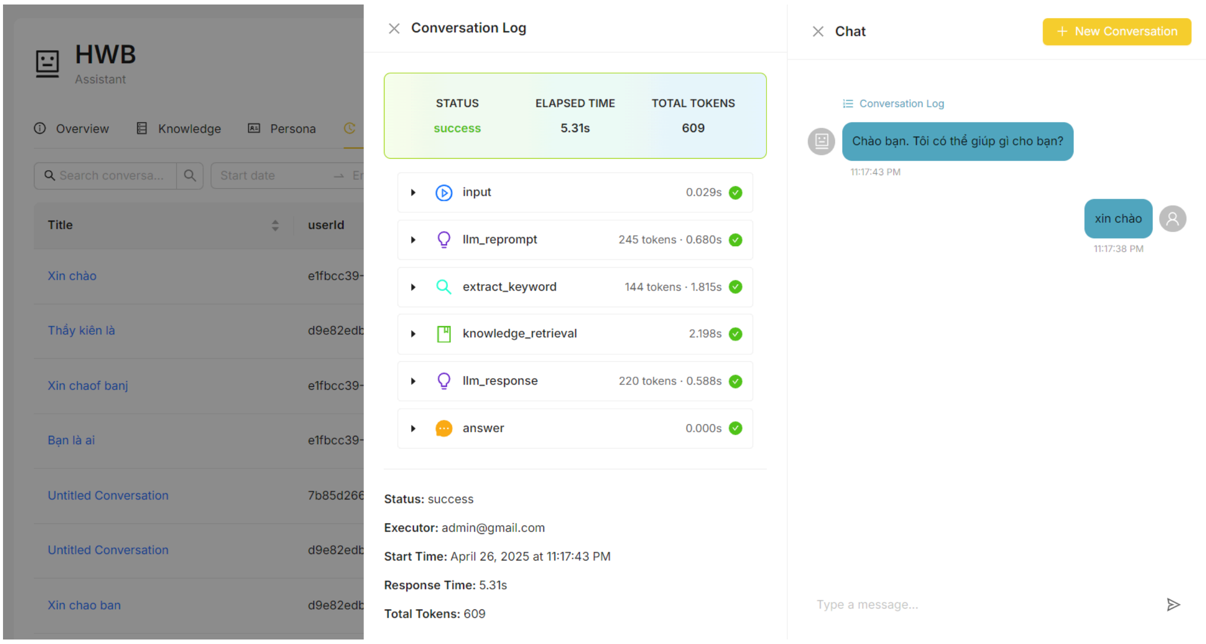
- In the bot detail page, click on the test tab.
- Click on Debug chat flow.



- If you have previously opened a conversation, then click on the "New Conversation" button. Otherwise, a new conversation will be automatically opened.



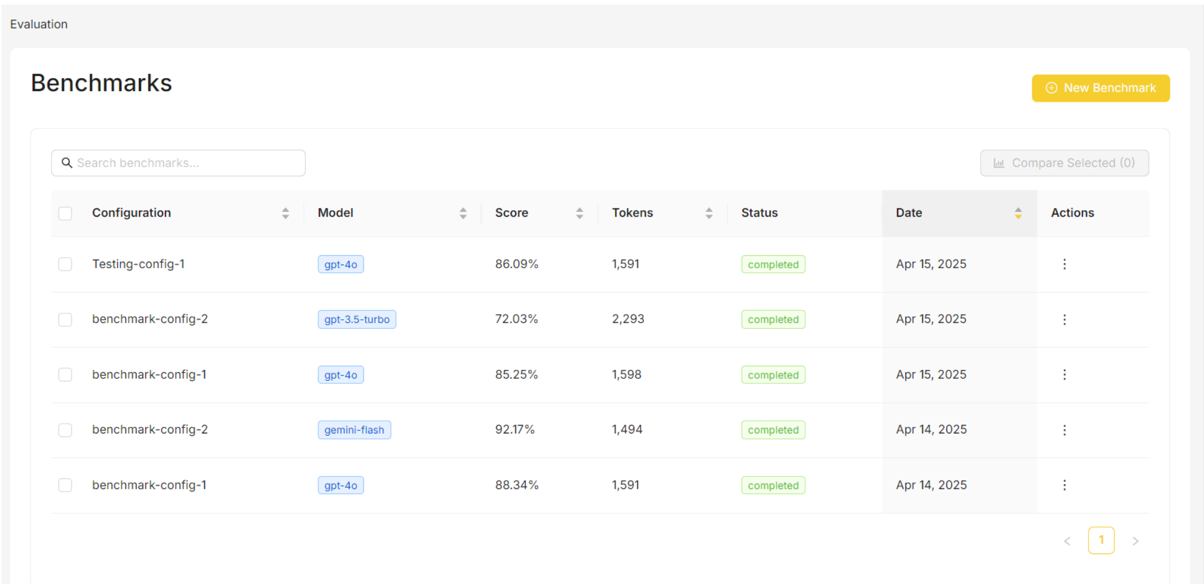
- From here, chat normally to view conversation logs.



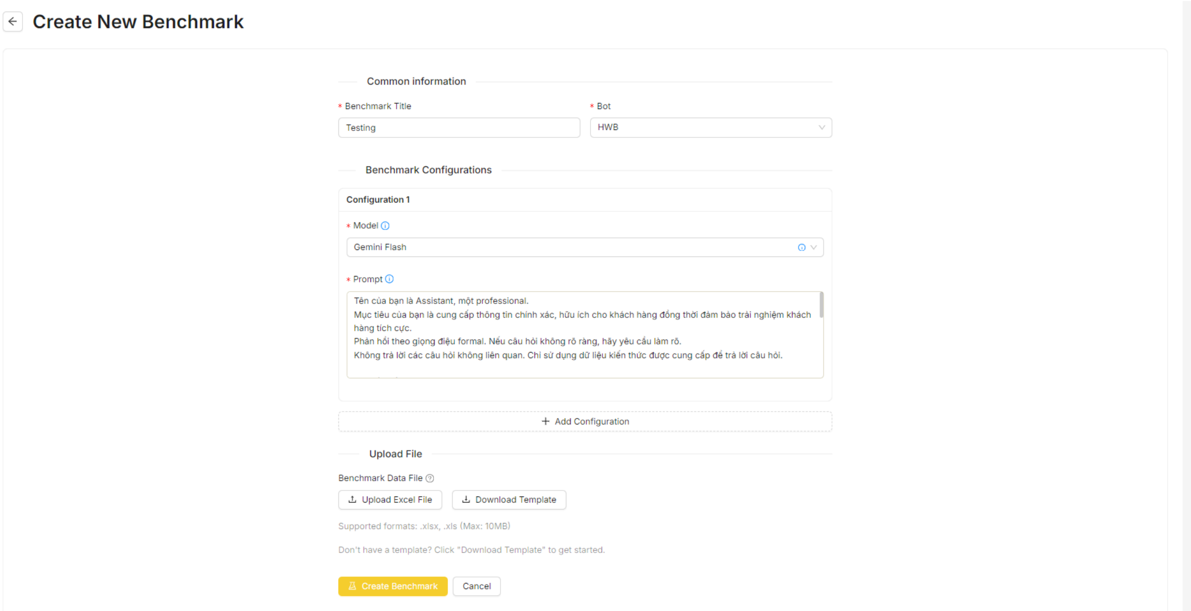
3.4.9 Create a test for bot evaluation

Description: The User wants to create a new bot evaluation test

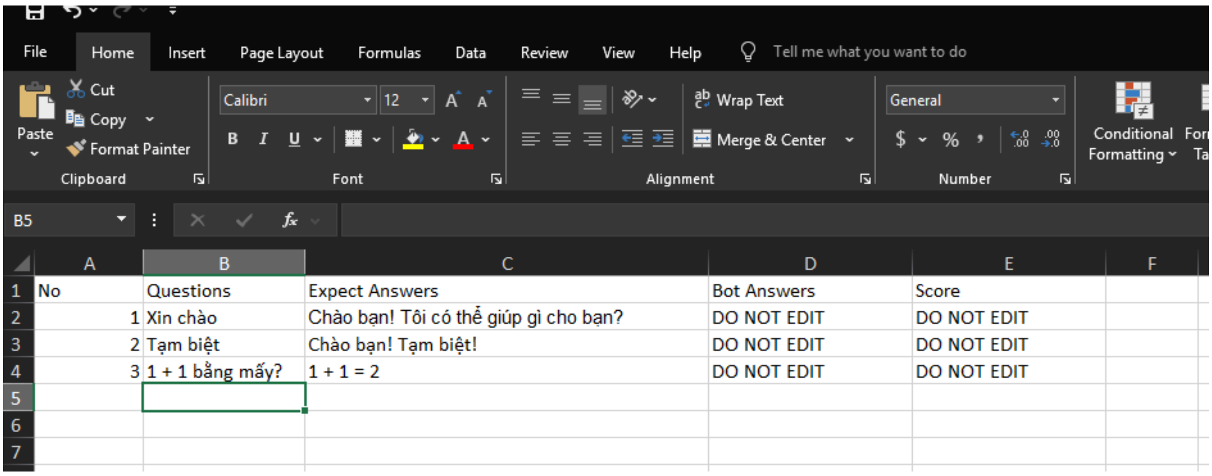
- Open the evaluation page, the user will see a list of previous evaluation tests. Click on the “new benchmark” button to start testing



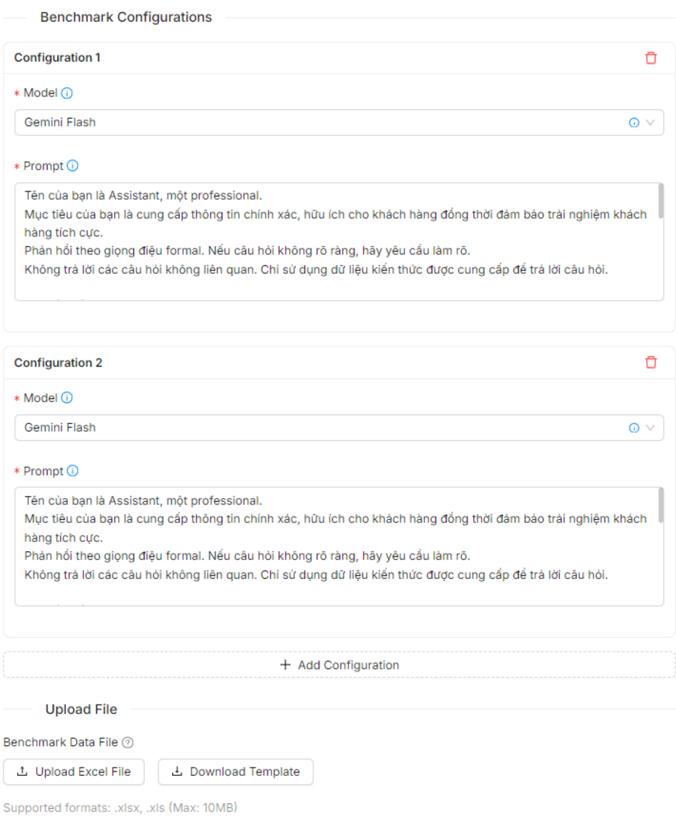
- Inside the new benchmark page, the user will input a title for the evaluation test.
- Then the user will select which bot they want to test. The bot list will include all the bots of the organization the user is in.



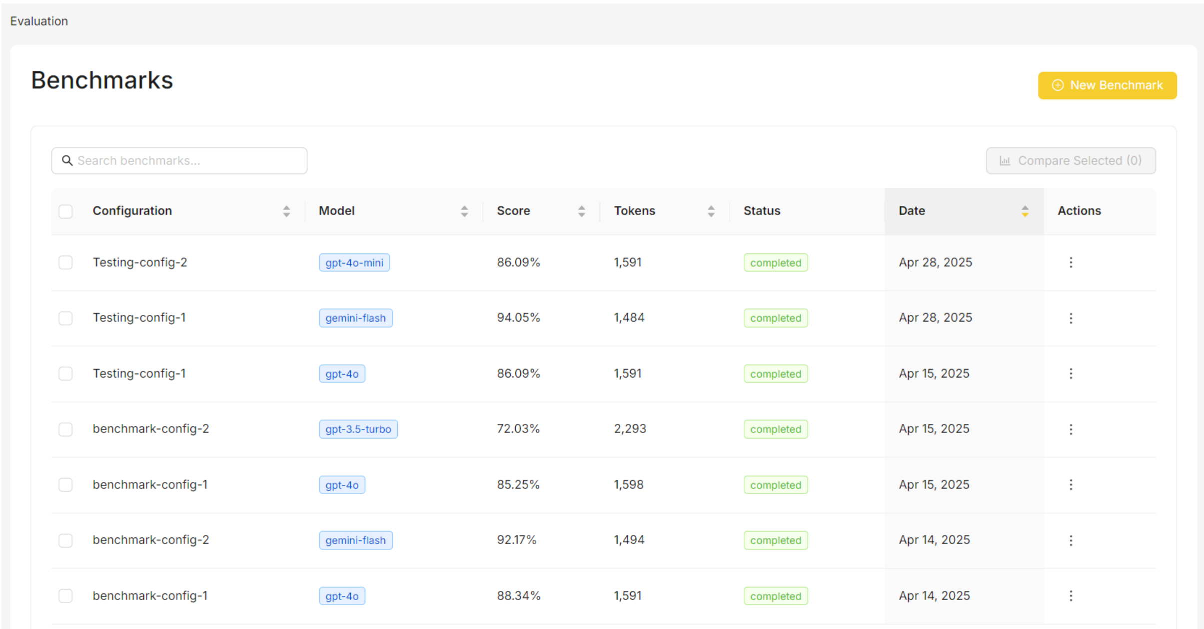
- To properly test bot evaluation, users need to input a question file. The user can download the template file and update it with their own questions and expect answers



- The user then chooses to configure the benchmark to evaluate their bot, including choosing models and choosing their prompt.
- If they want to do more than one config, they can click on "new configuration" button and a new config form will display. Input this and the system will run more than one configuration.



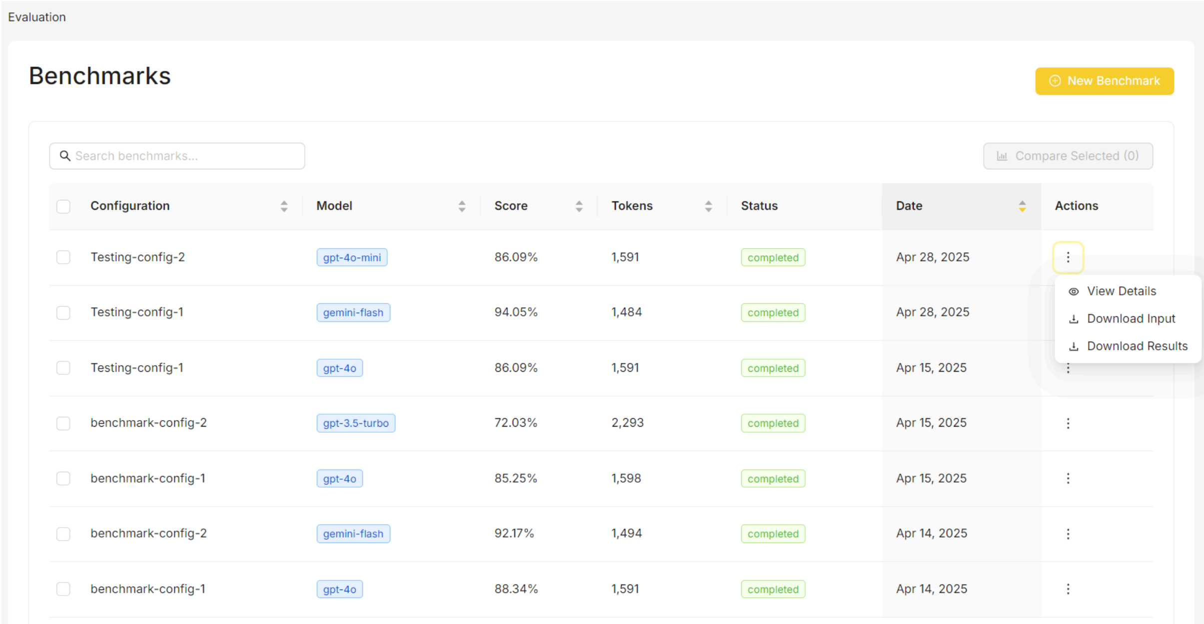
- Once ready, click the “create benchmark” button and let the system run the configs.



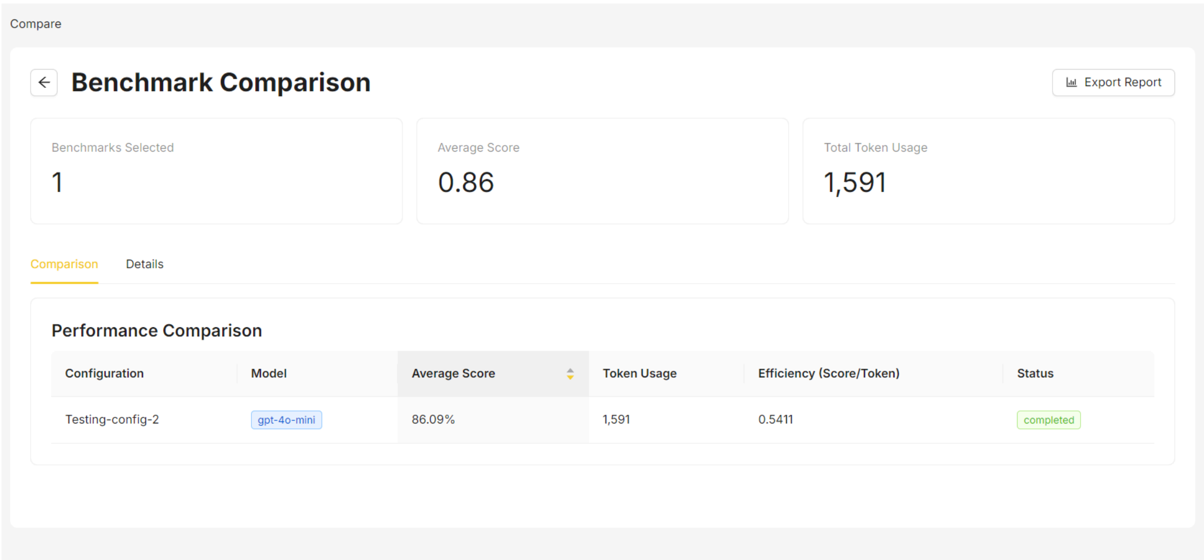
3.4.10 View bot evaluation test detail

Description: The User wants to view details of the bot evaluation test

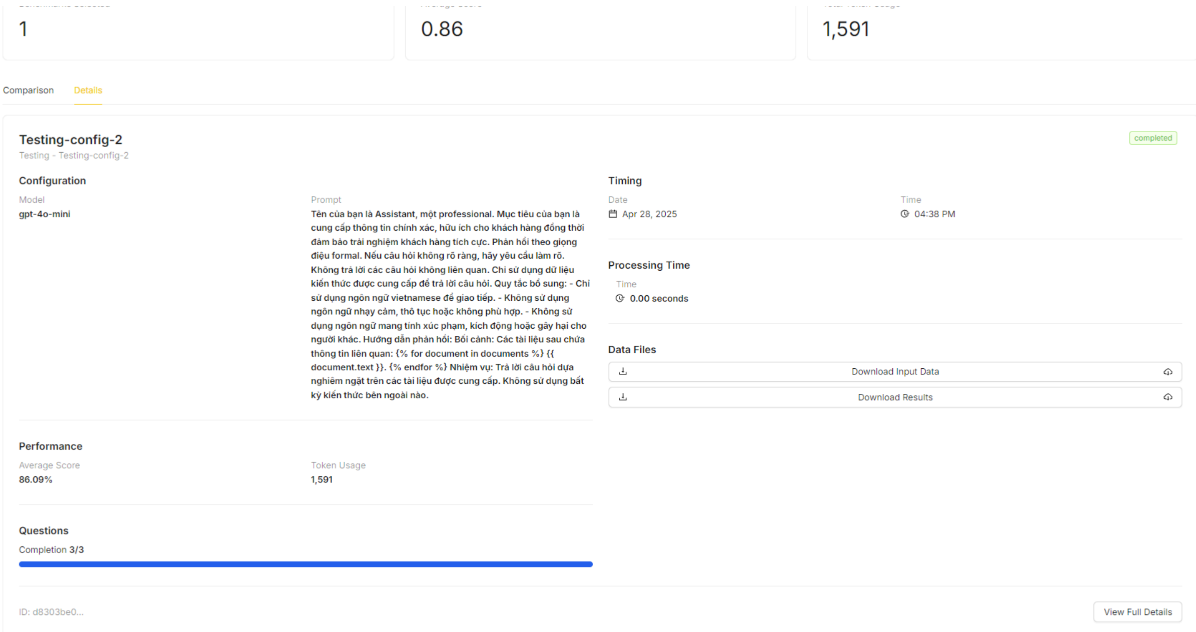
- In the list, click on View Detail in the action column



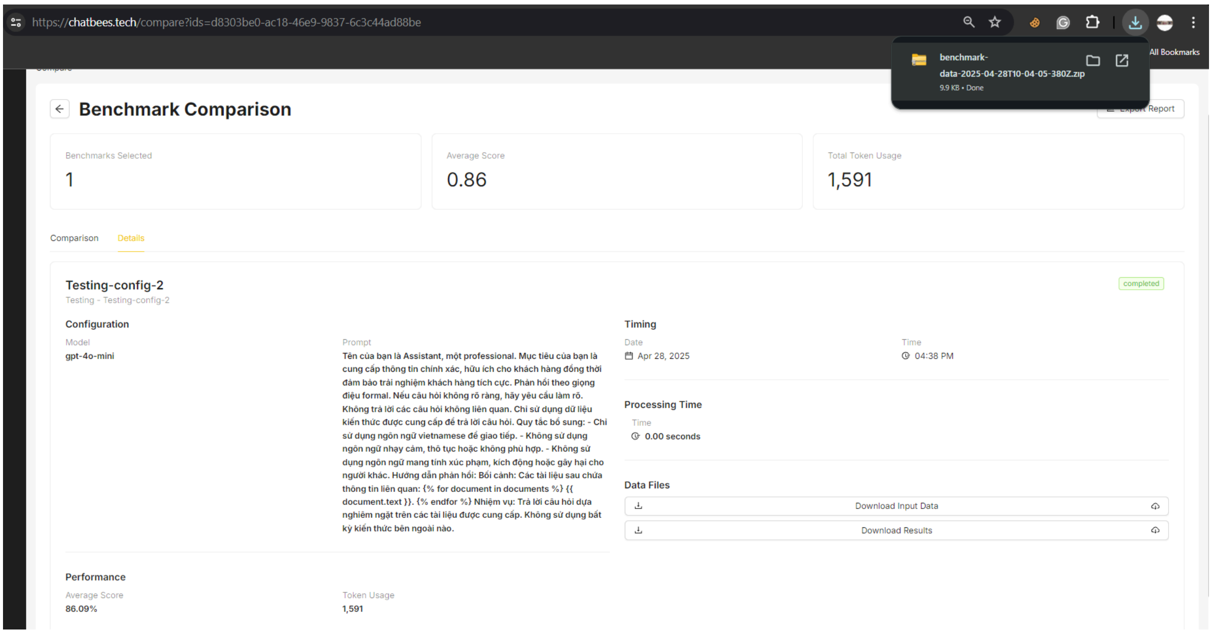
- In the detail page, the user can view details of the evaluation run, including response time, token usage, chosen bot configurations, and scores of the similarity to expected answers.
- Shows a side-by-side performance comparison of your selected AI models, displaying key metrics like scores, token usage, and efficiency in an easy-to-read table format. This gives you a quick overview of how different configurations perform against each other.



- Provides in-depth information about each individual benchmark run, including the specific prompts used, timing information, processing speeds, and completion rates. Here you can examine the full context of each test and download the associated data files for further analysis.



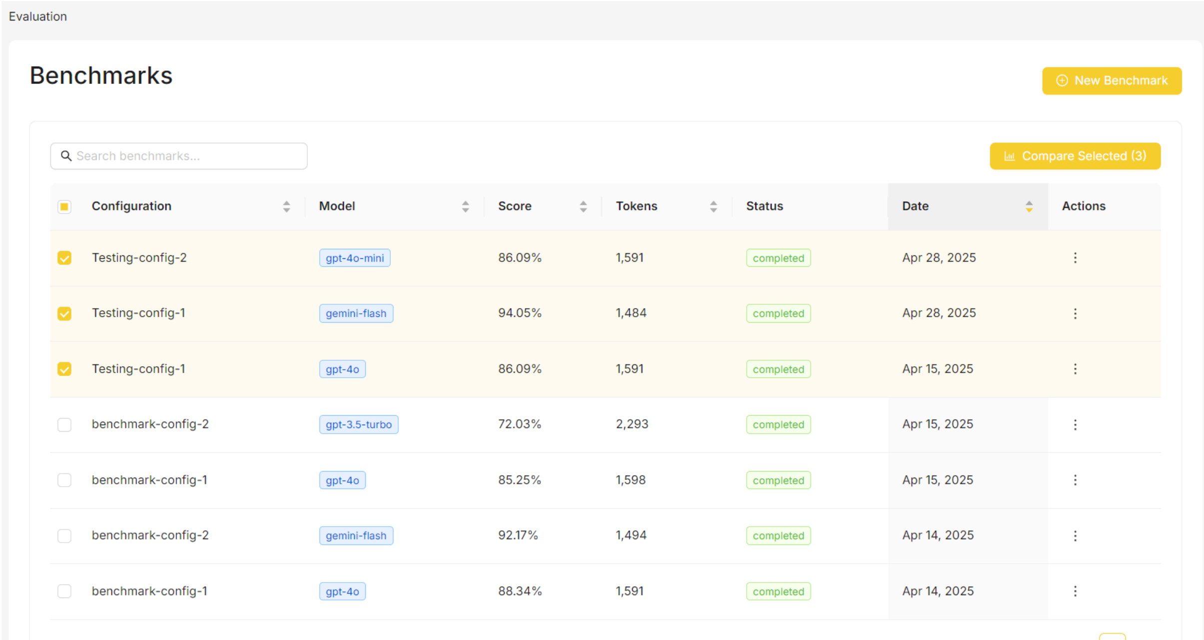
- Users can also download input data files, output data files for further inspection.



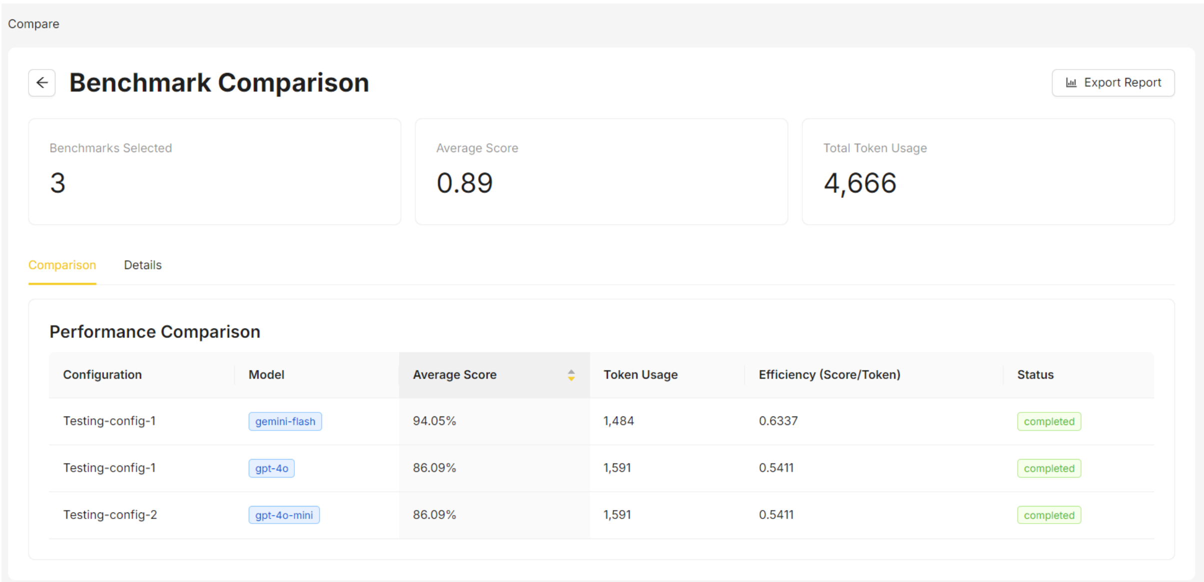
3.4.11 Compare different bot evaluation tests

Description: The User wants to view and compare bot evaluation tests

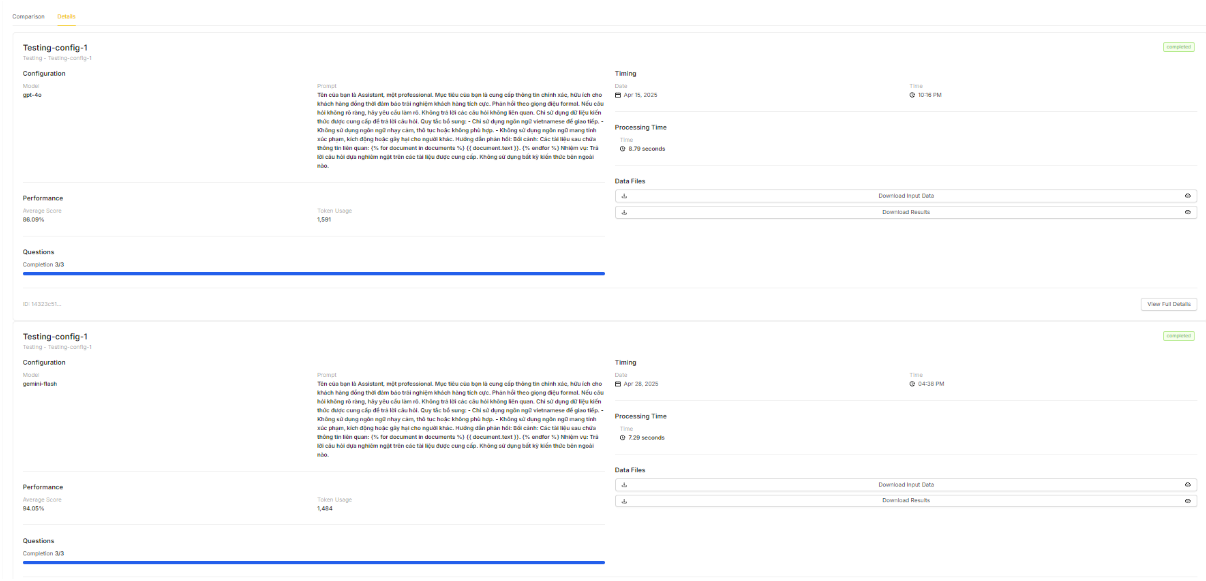
- In the list, click on the check box in the first column to select multiple evaluation tests. Click the “compare selected” button to be directed to the detail page.



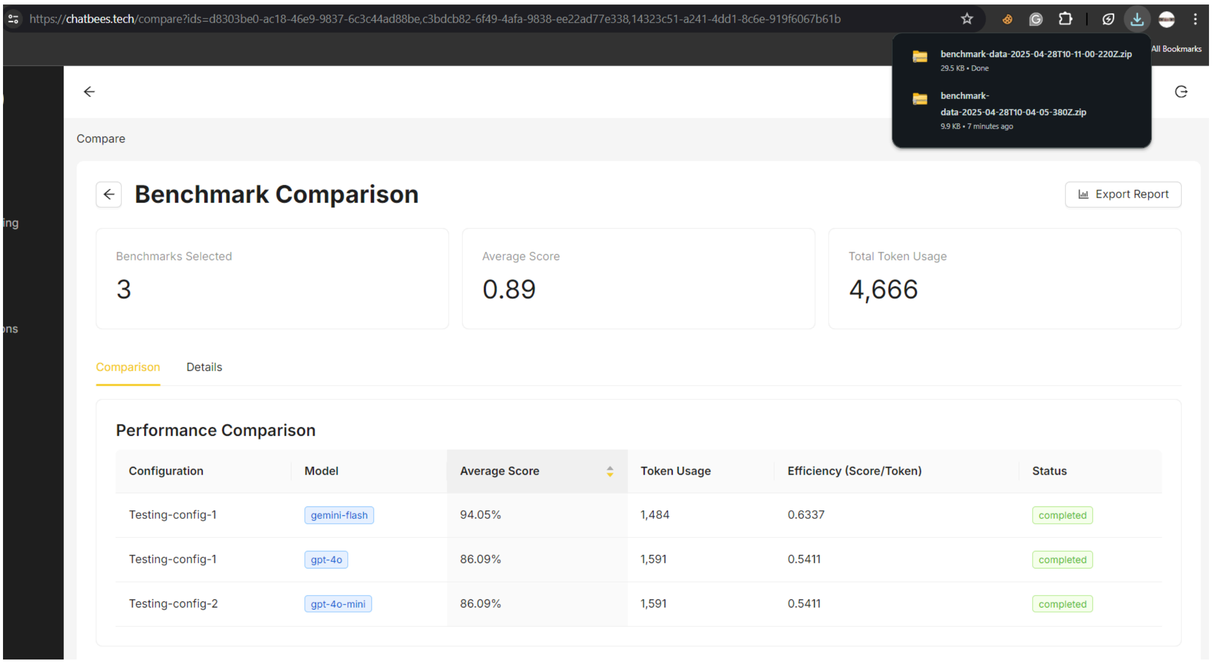
- Shows a side-by-side performance comparison of your selected AI models, displaying key metrics like scores, token usage, and efficiency in an easy-to-read table format. This gives you a quick overview of how different configurations perform against each other.



- Provides in-depth information about each individual benchmark run, including the specific prompts used, timing information, processing speeds, and completion rates. Here you can examine the full context of each test and download the associated data files for further analysis.



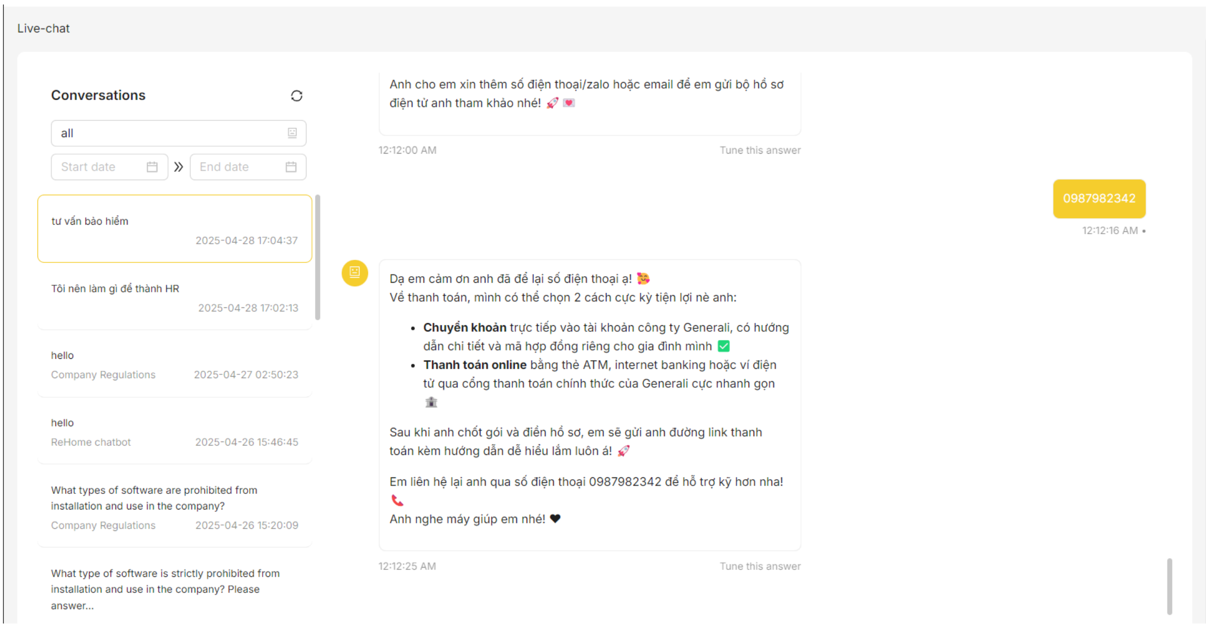
- Users can also download input data files, output data files for further inspection.



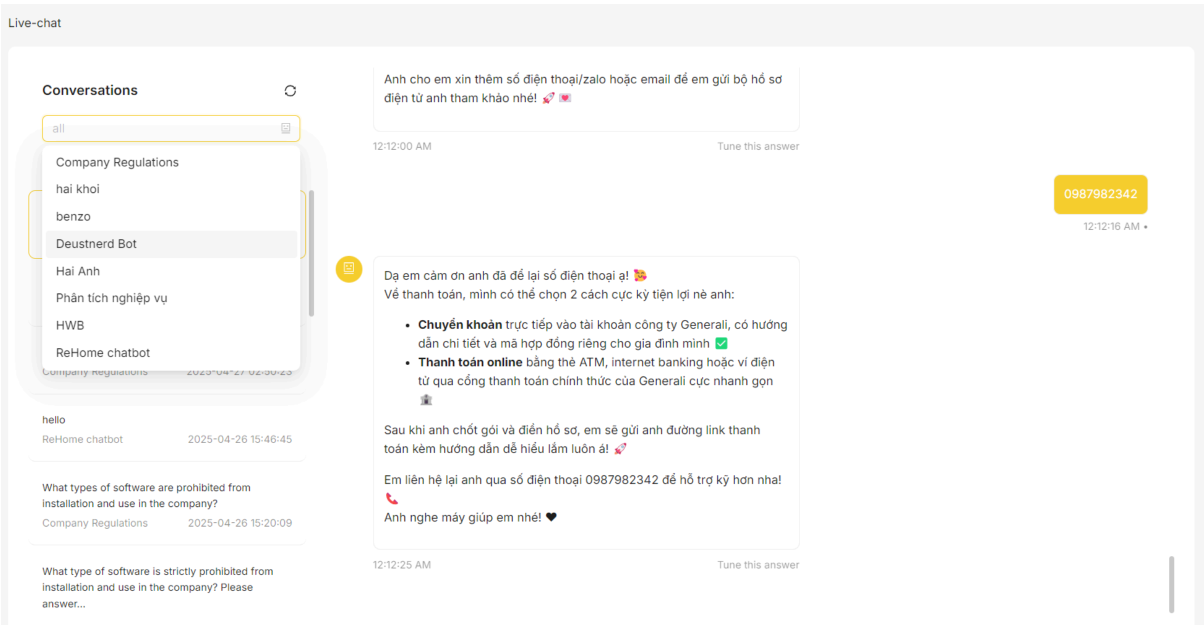
3.4.12 View live chat

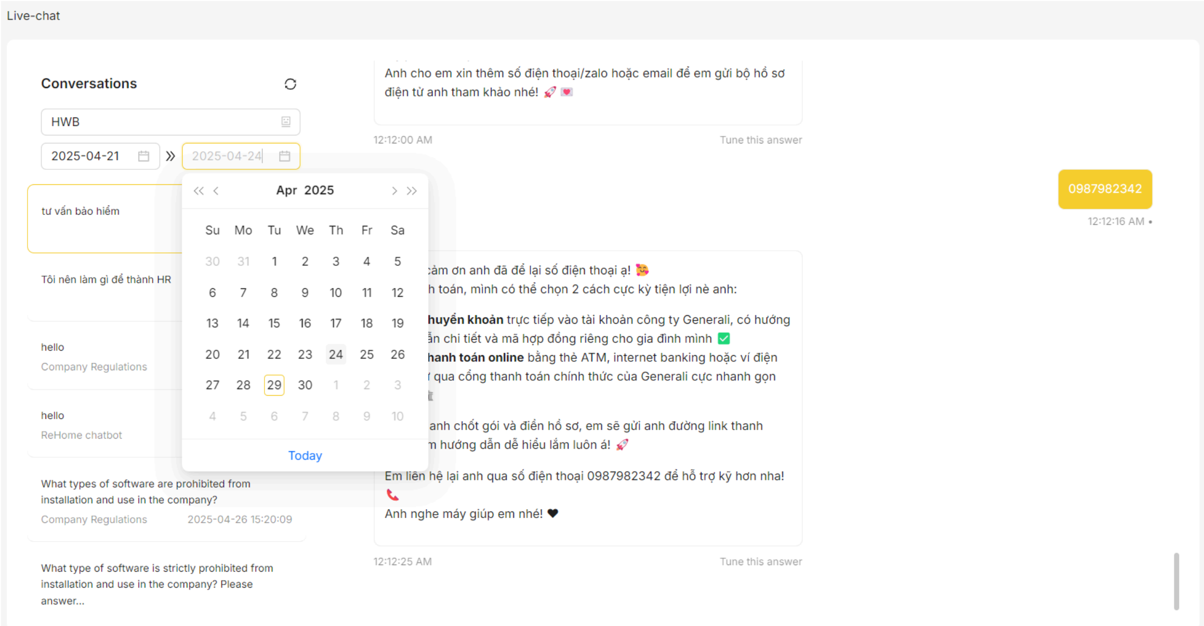
Description: The User wants to view customer conversation

- Click on tab "Chat tracking" on the sidebar, the system will take the user to a live chat page where user can monitor user conversations.



- The user can select which bot they want to specifically monitor, or the date range from which the conversation happens.



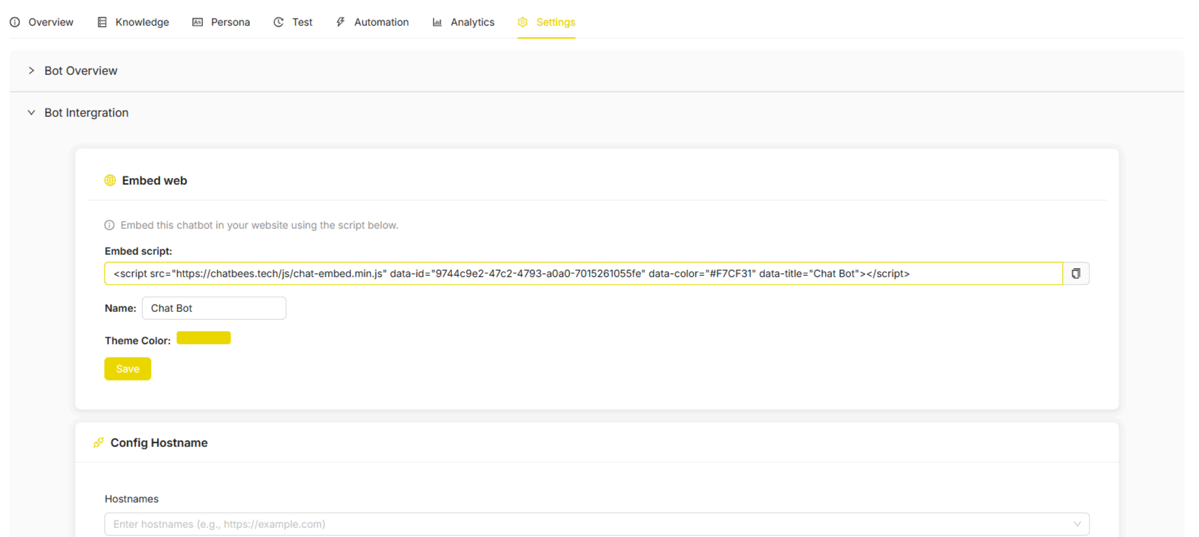


3.5. Bot Integration

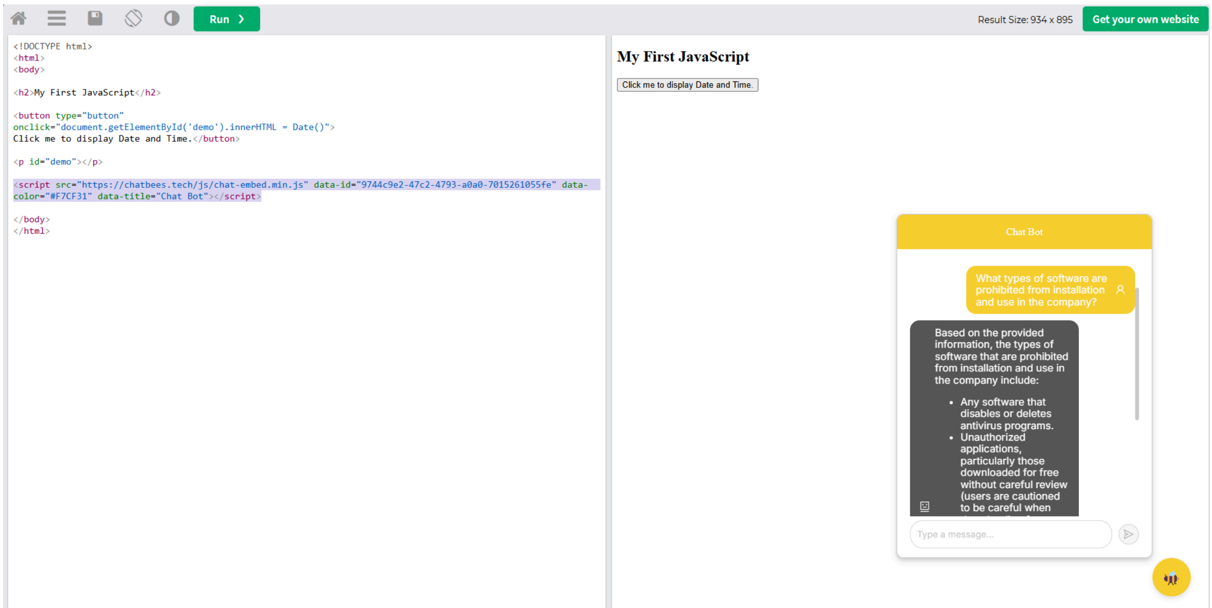
3.5.1 Embed Web

Description: Embed the bot on a website using a script tag to respond based on parsed data

- Copy “Embed script” from the "Embed Web" section in the Settings tab.



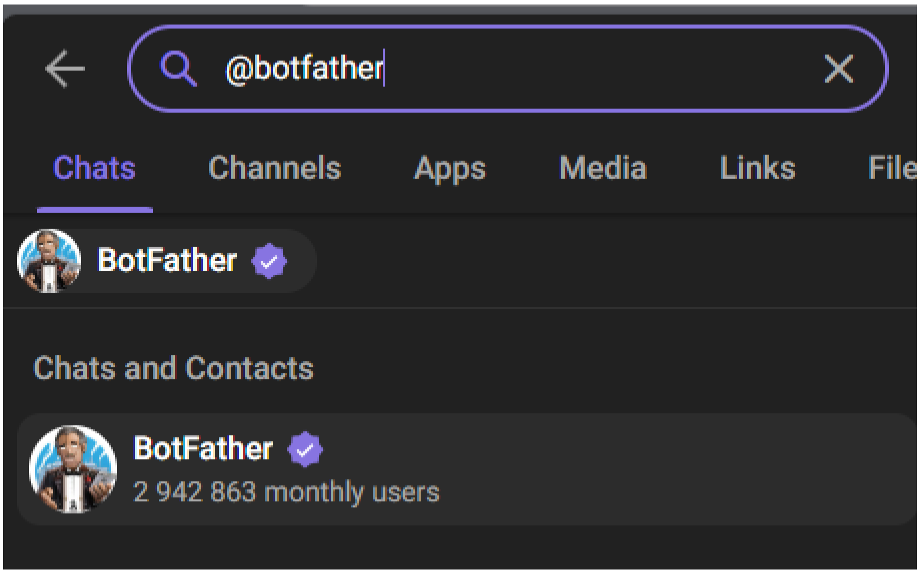
- Paste it into your website's source code to enable chat functionality.



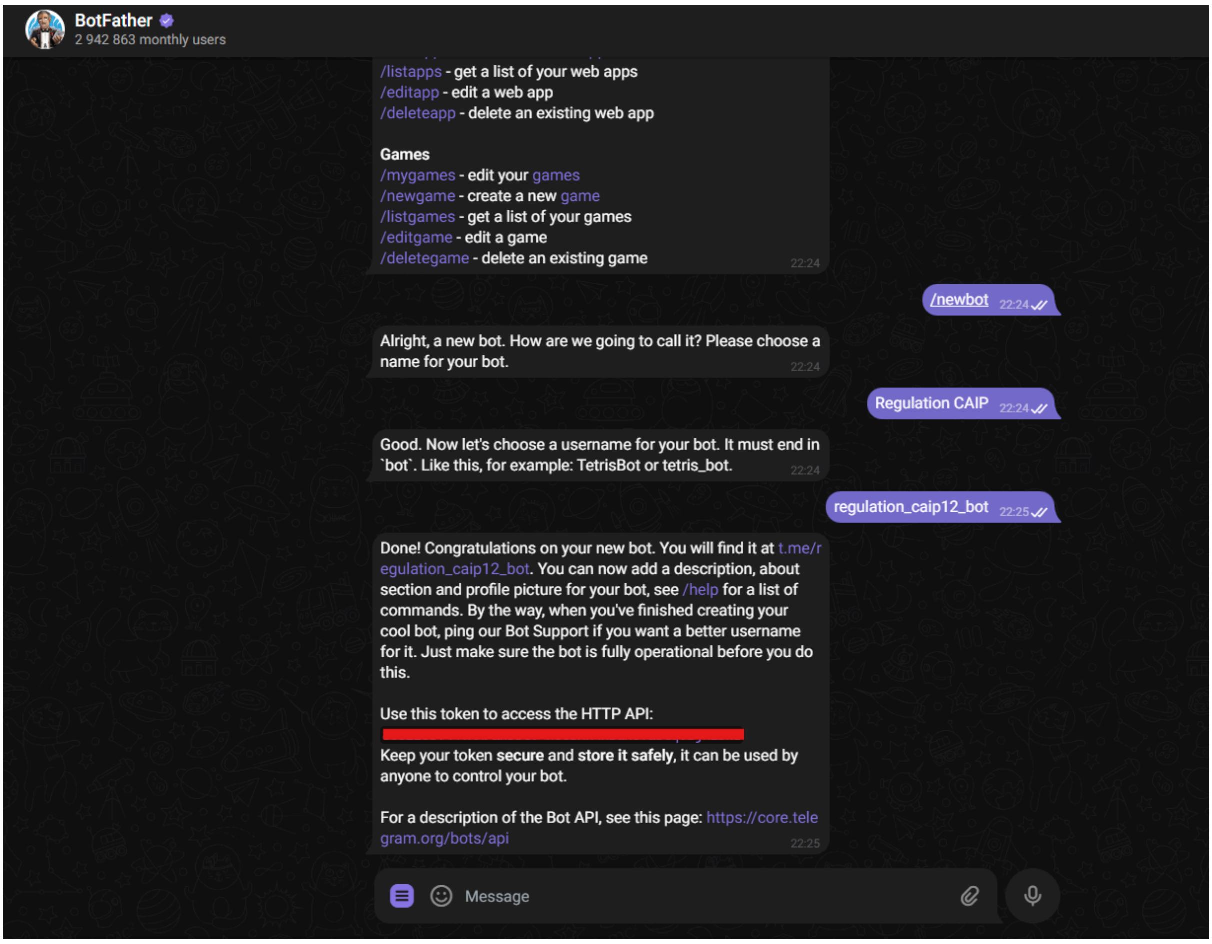
3,5.2 Telegram Integration

Description: Integrate the bot with Telegram for automated responses using parsed data.

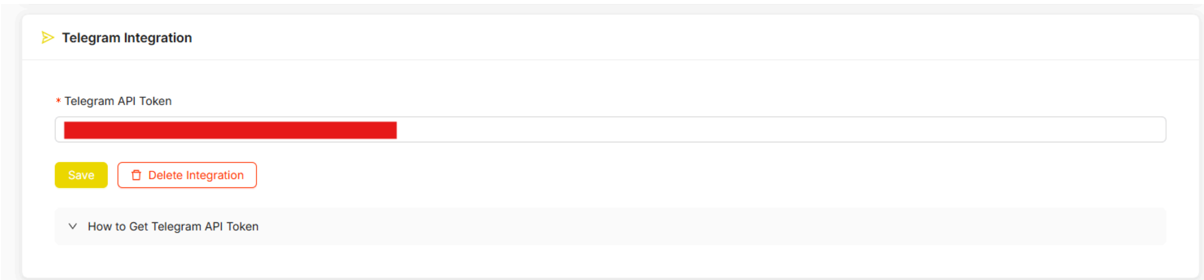
- Open Telegram and search for @BotFather.



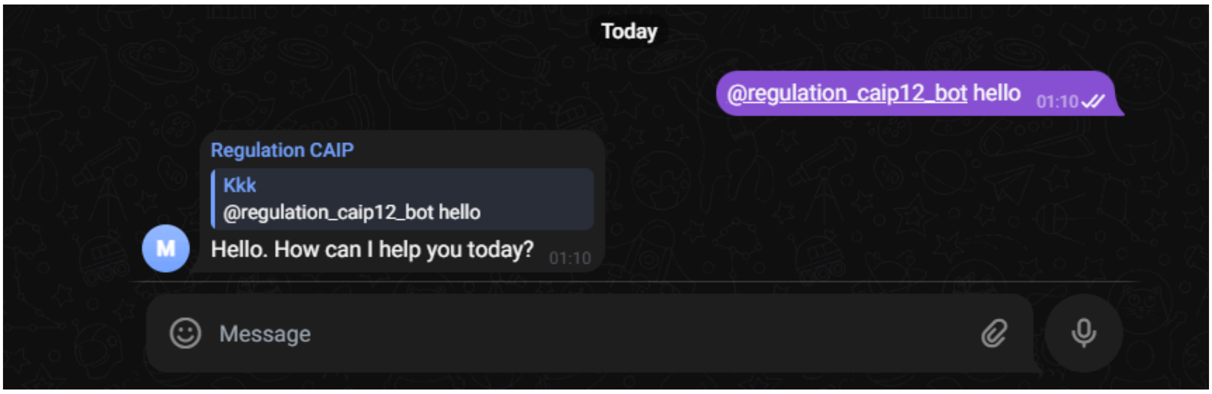
- Start a chat with BotFather and send the command /start.
- Send the command /newbot to create a new bot.
- Follow the instructions to set a name and username for your bot.
- Once created, BotFather will provide you with an API token.



- Copy API token and paste it into the field “Telegram API Token”.



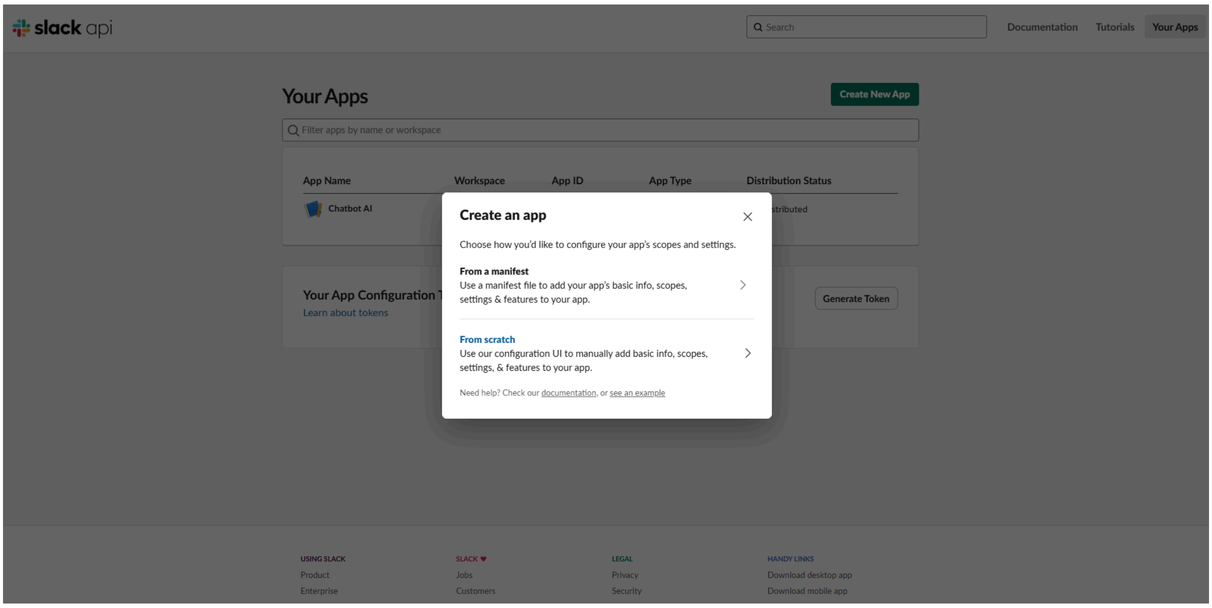
- Start chatting with bot.

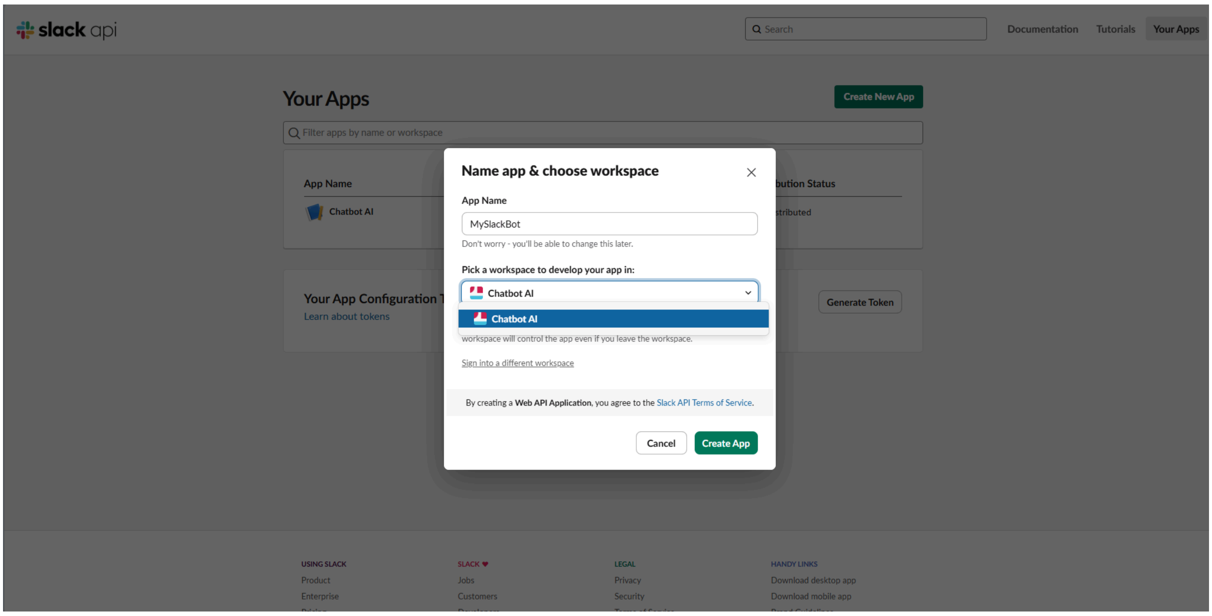


3.5.3 Slack Integration

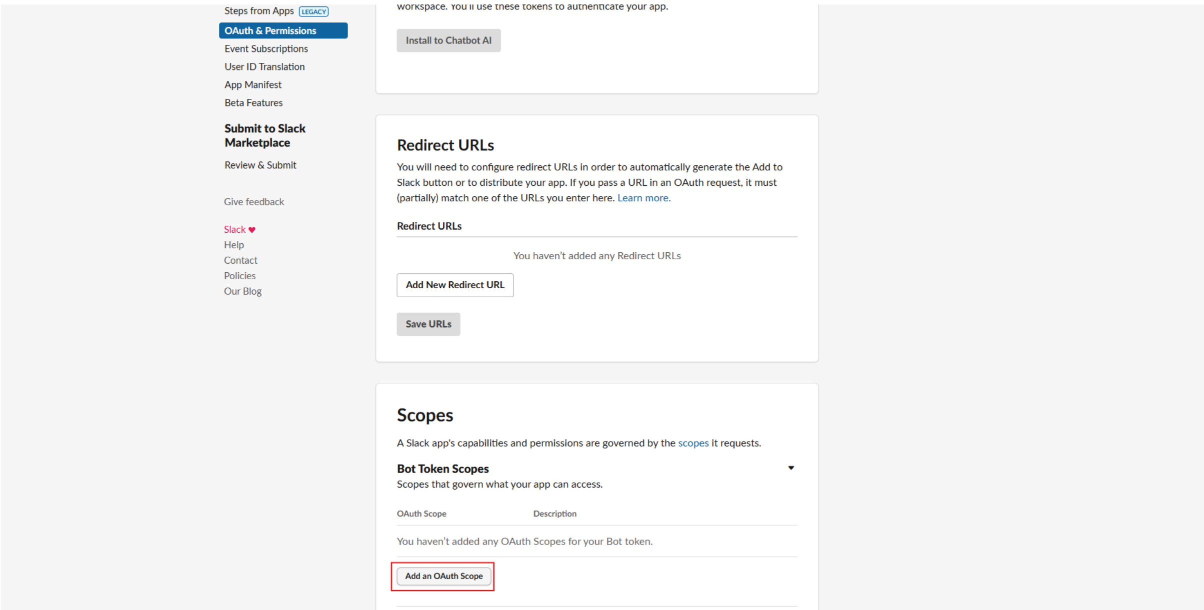
Description: Integrate the bot with Slack for automated responses in channels using parsed data.

- Go to https://api.slack.com and sign in with your Slack account.
- Click "Create New App" → Select "From scratch".
- Name your app (e.g., MySlackBot) and choose a workspace.

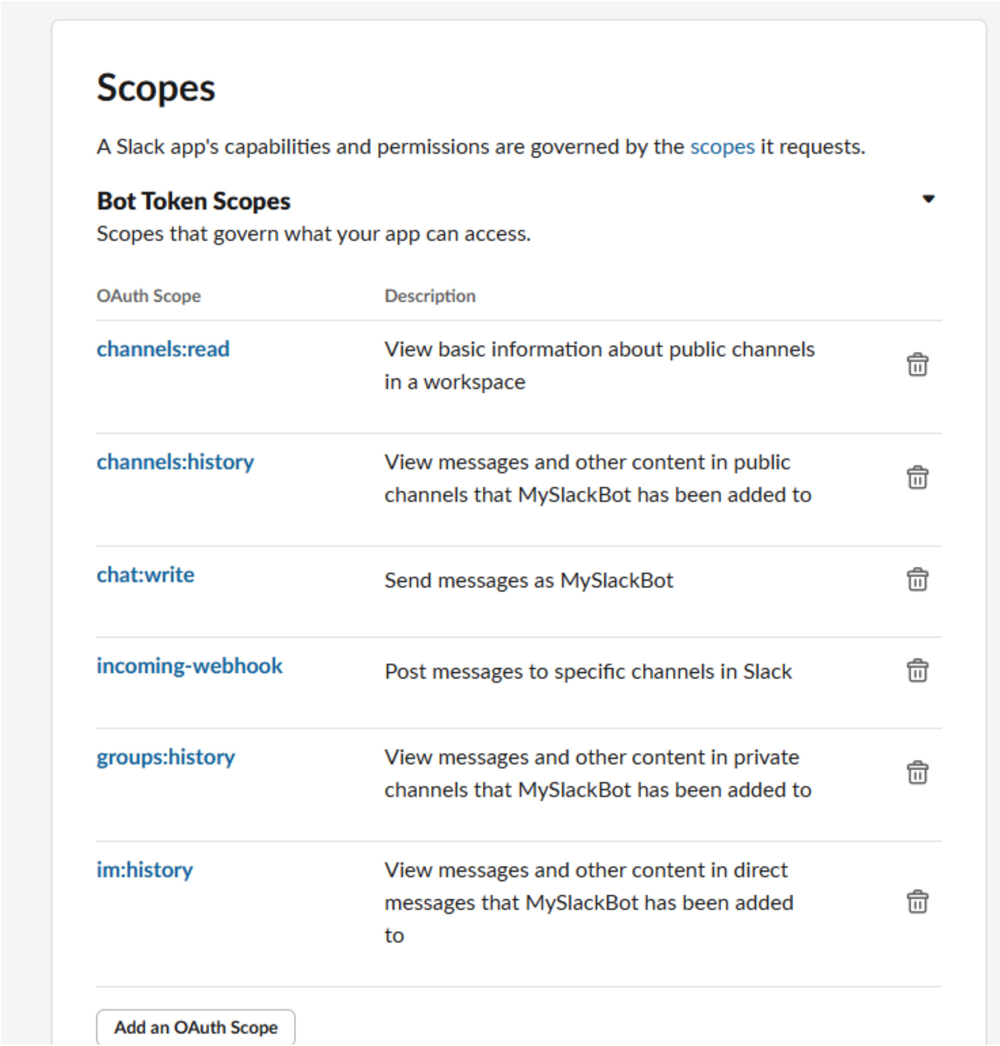




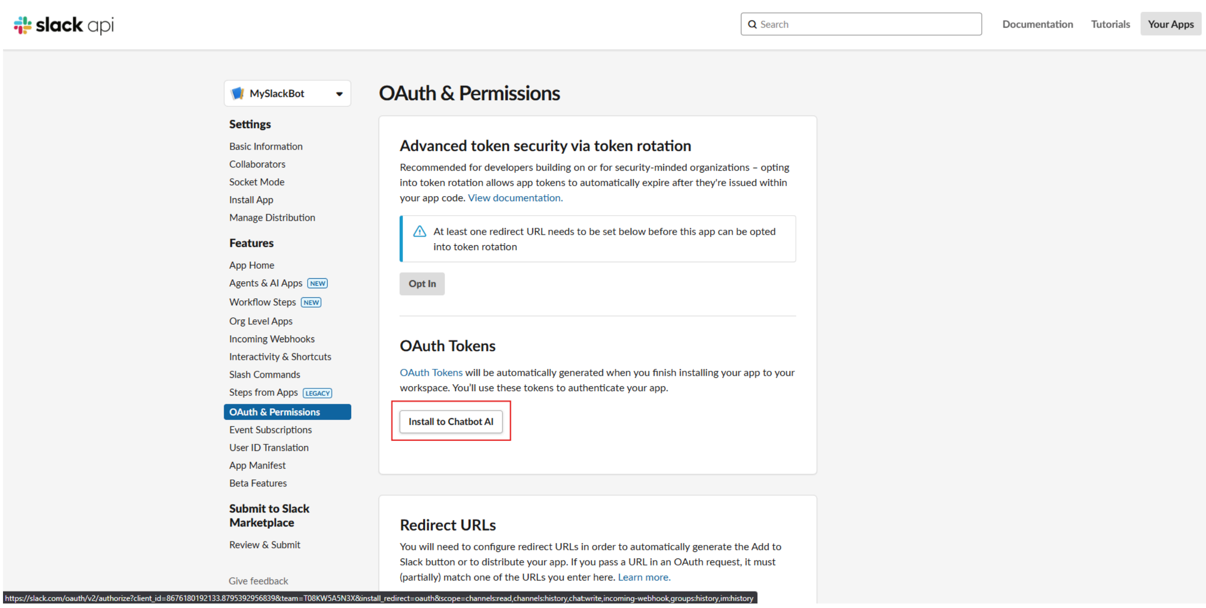
- Navigate to the "OAuth & Permissions" section in the sidebar.

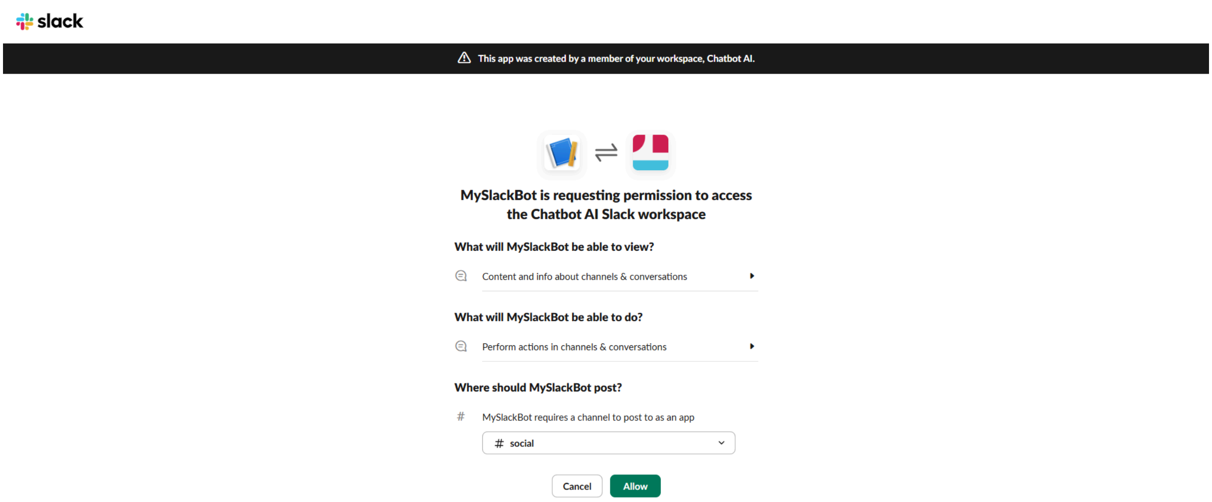


- Add the following Bot Token Scopes: channels:read, channels:history, chat:write, incoming-webhook, groups:history, im:history.

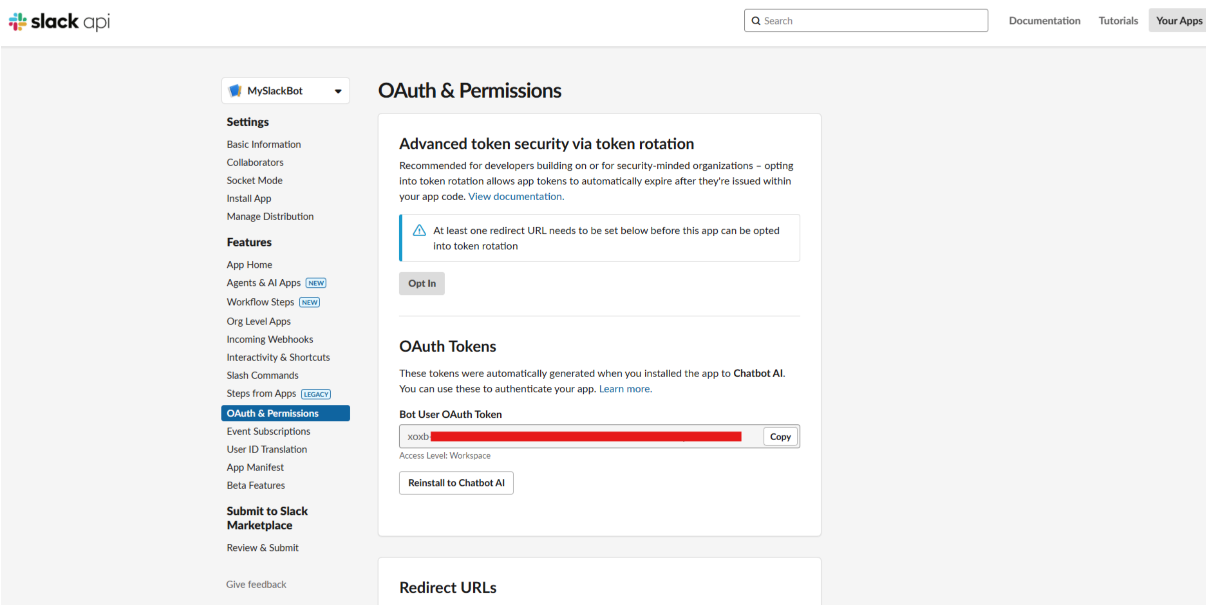


- Click "Install to Workspace" to install the app to your workspace, then authorize it.

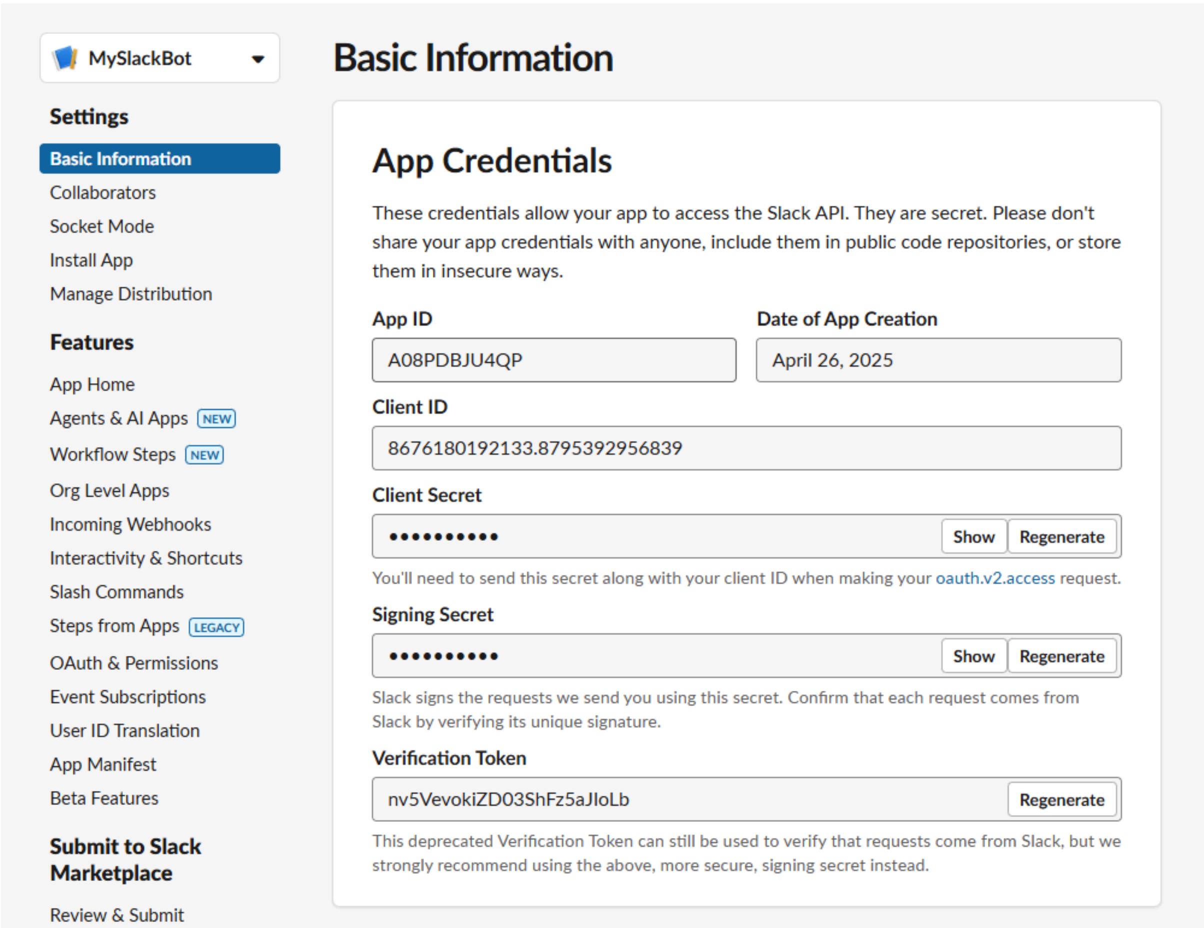


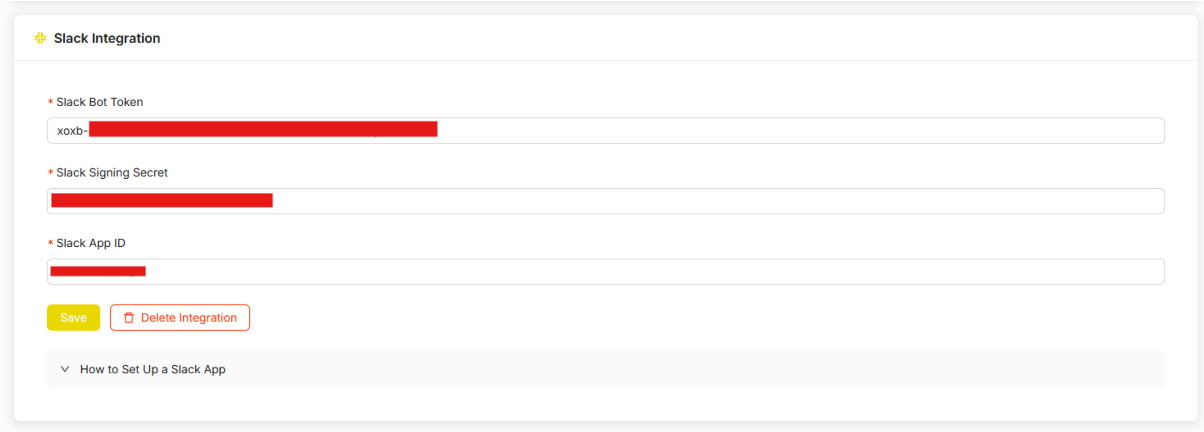


- Copy the Bot User OAuth Token (it starts with xoxb-) and paste it into the "Slack Bot Token" field.

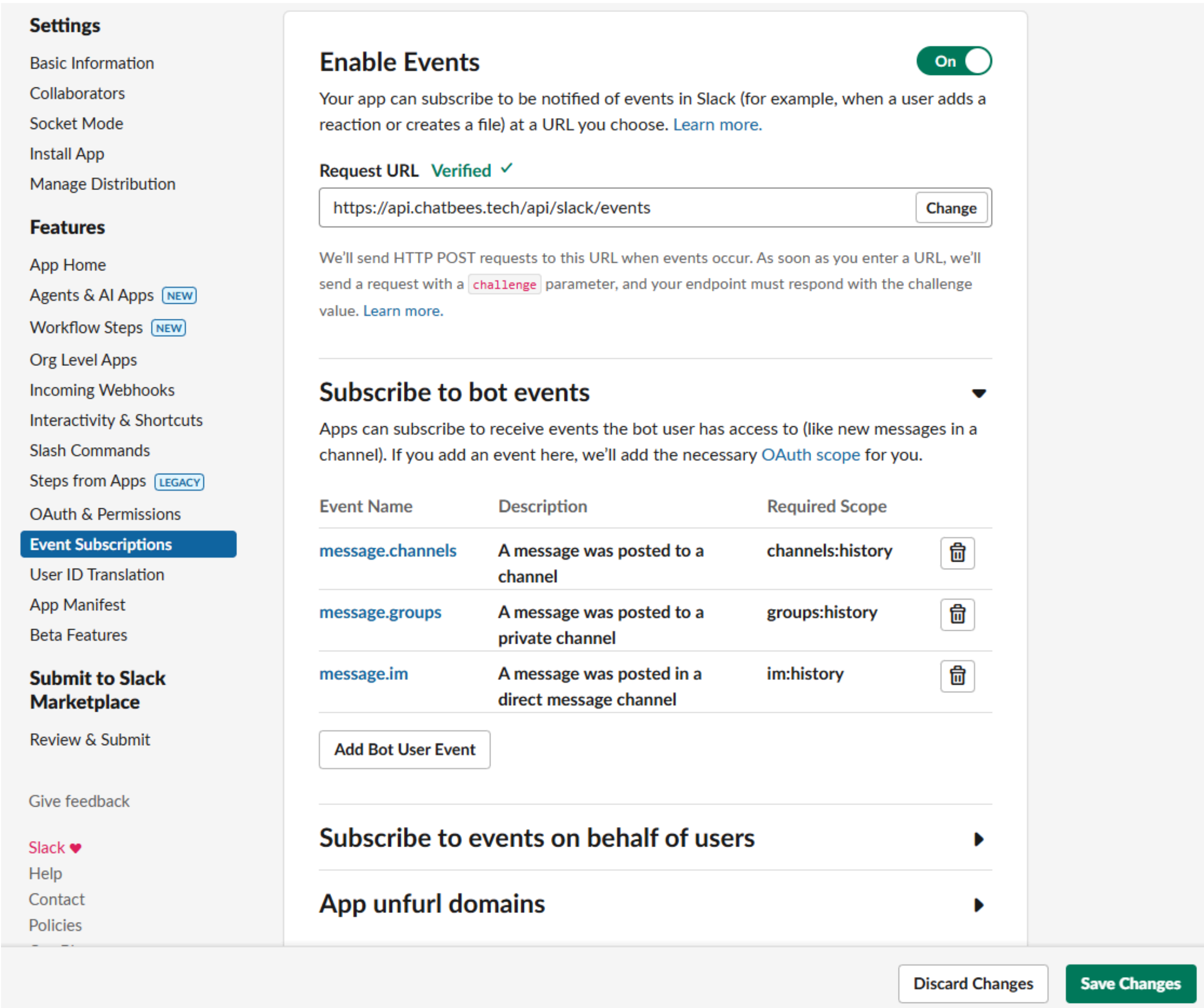


- In the "Basic Information" section, find the App ID.
- Copy the App ID and paste it into the "Slack App ID" field

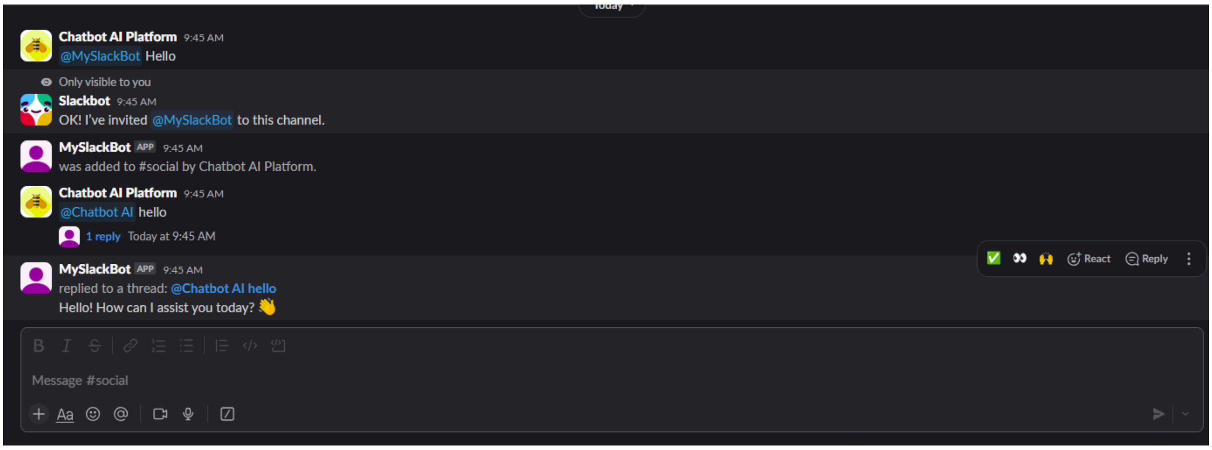




- Navigate to the "Event Subscriptions" section in the sidebar.
- Enable Events by toggling the switch to "On".
- Add a Request URL: https://api.chatbees.tech/api/slack/events
- Subscribe to the following bot events: message.channels, message.groups, message.im.
- Click "Save Changes" to apply the settings.



- Start chatting with bot in Channel.



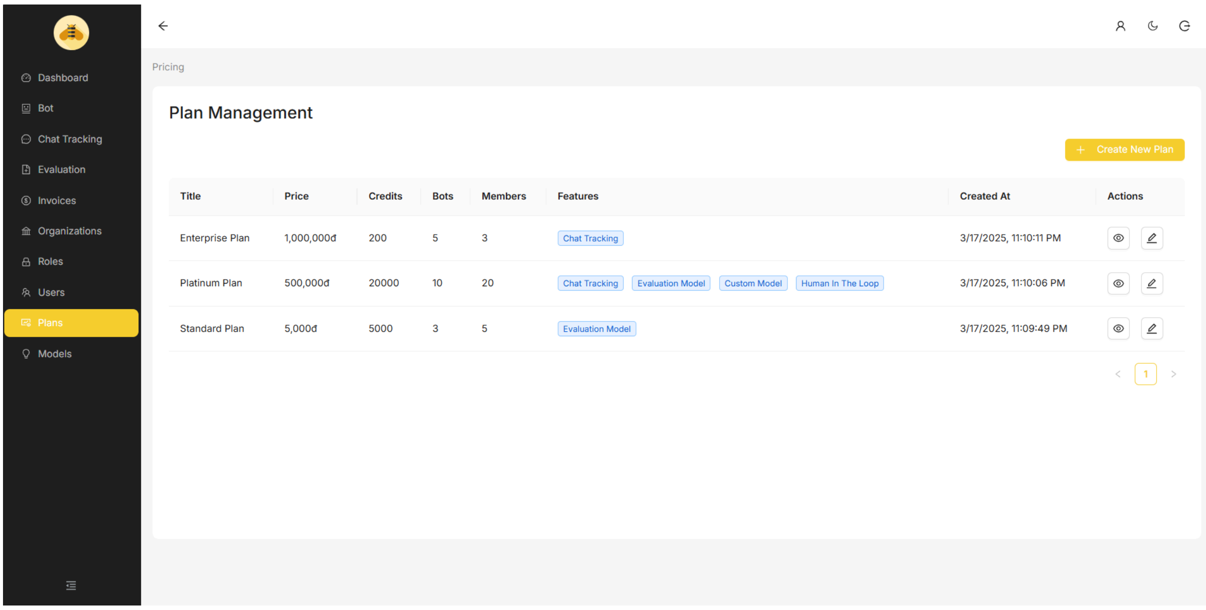
3.6. Admin Management

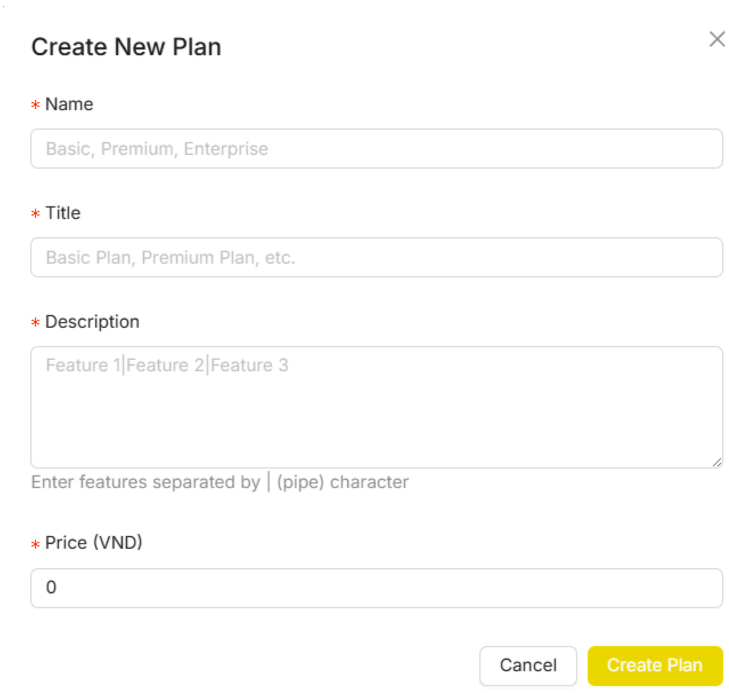
3.6.1 Plan Management

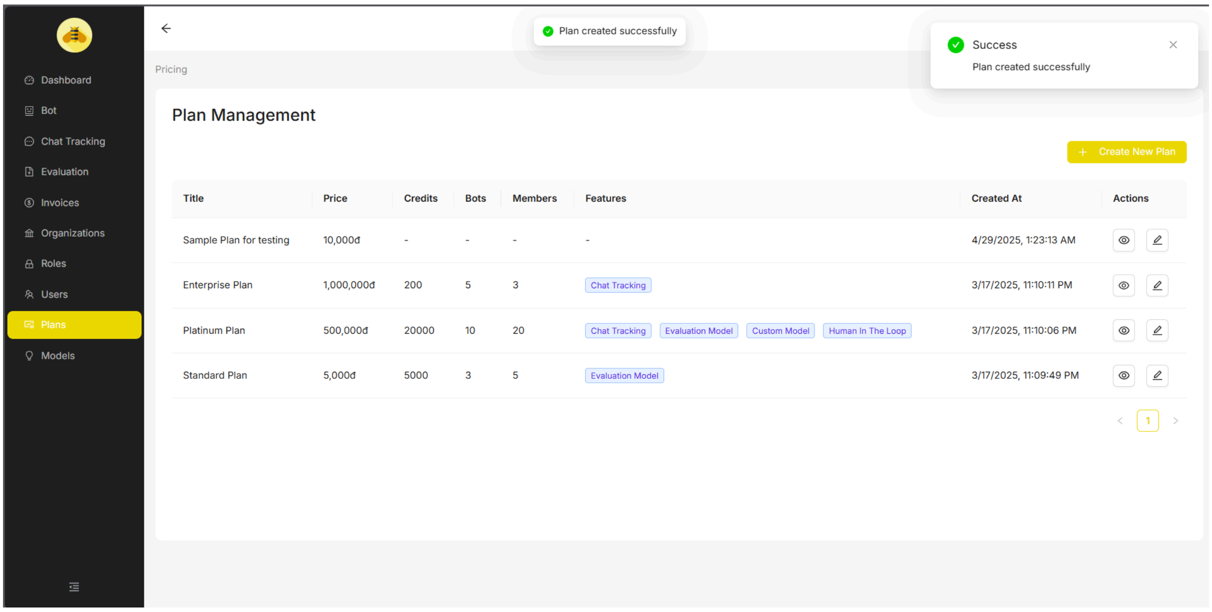
3.6.1.1 Create Plan

Description: Admin wants to create new plan for system

- After logging into the system, Admin navigates to the Plans section.
- Admin clicks “Create New Plan”.
- Admin fills in Name, Title, Description and Price for new plan.
- User clicks the “Create Plab” button.



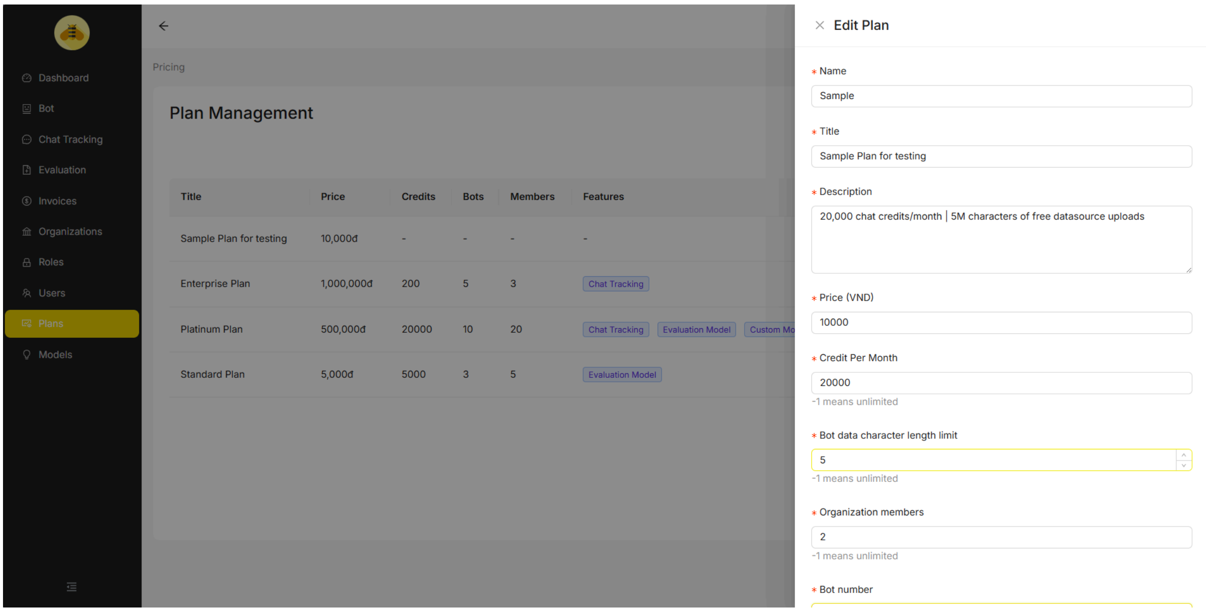


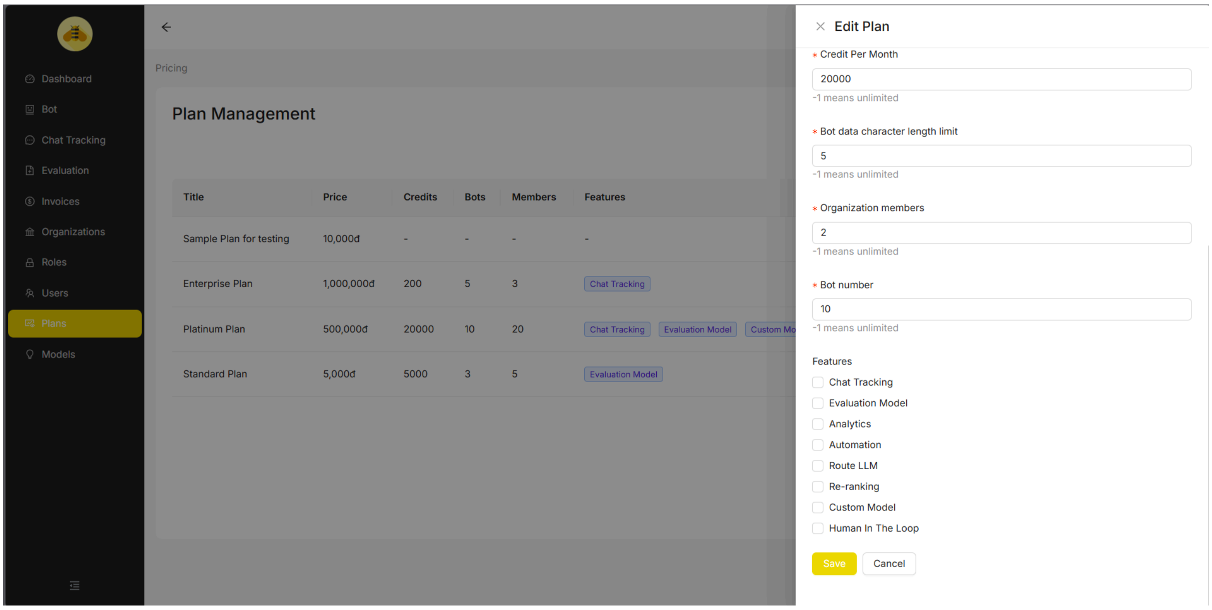


3.6.1.2 Update Plan

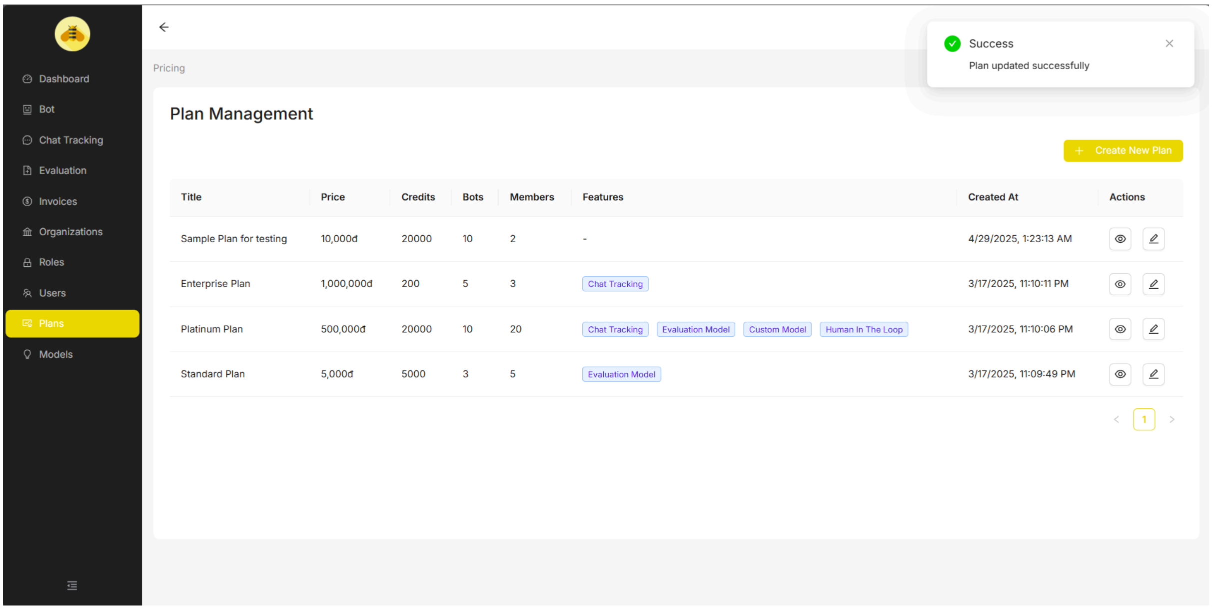
Description: Admin wants to update plan

- Admin clicks icon edit in Column Actions of the plan Admin wants to change.
- Admin enters the fields that need to be changed to suit the plan.





- User clicks the "Save" button.

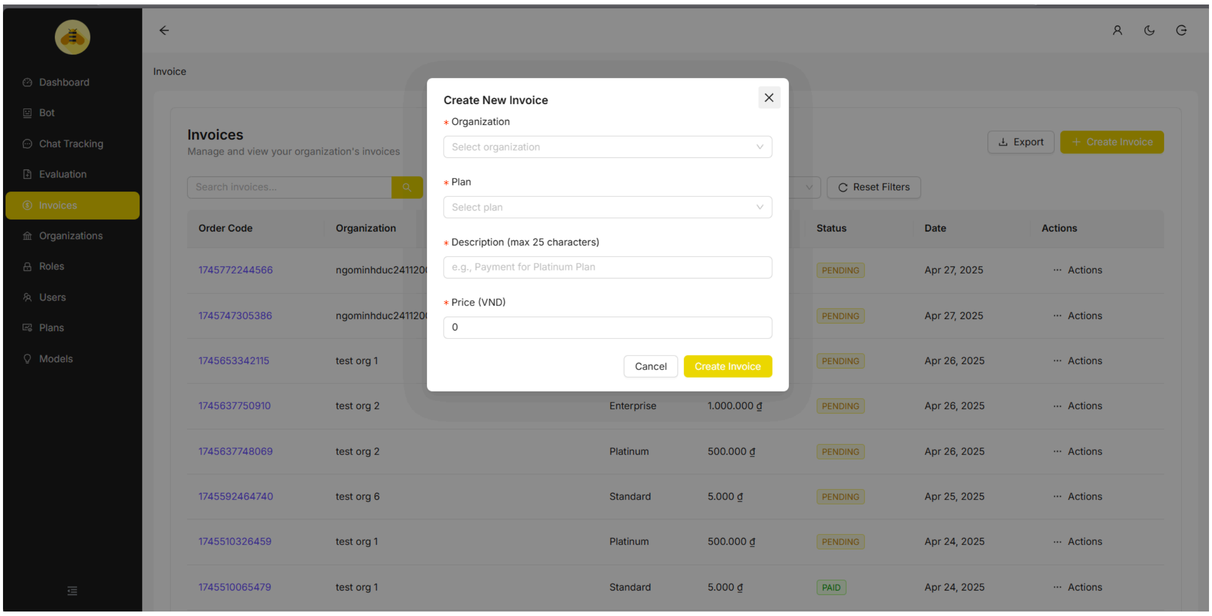


3.6.2 Invoice Management

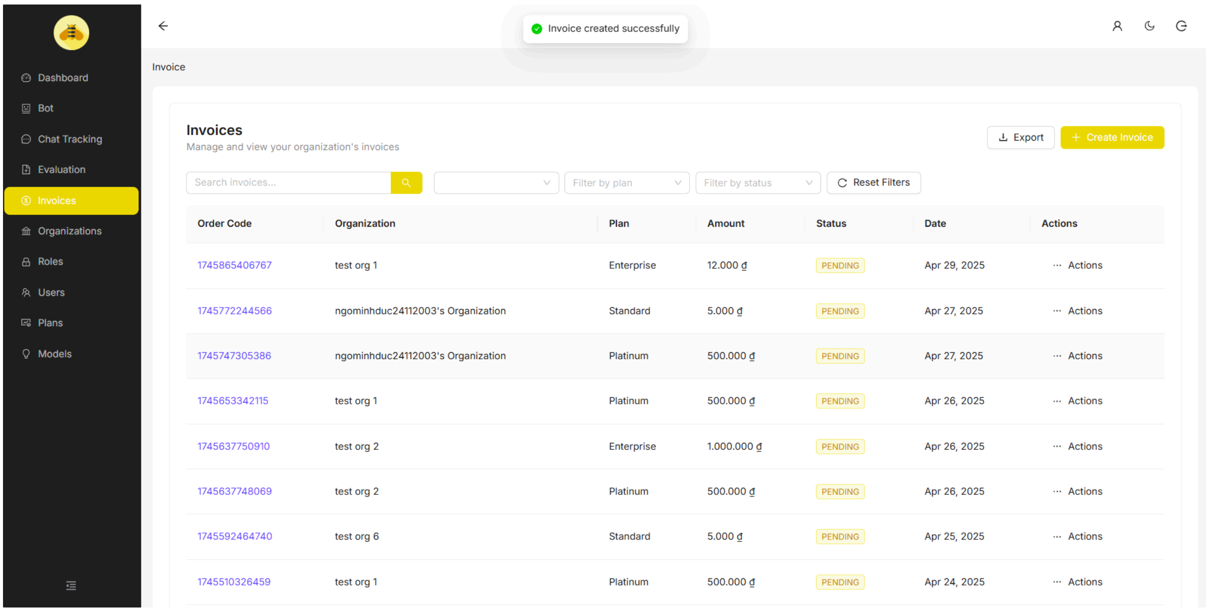
3.6.2.1 Create Invoice

Description: Admin wants to create a new invoice for Organization.

- After logging into the system, Admin navigates to the Invoices section.
- Admin click "Create Invoice" button on Table
- Admin choose Organization, Plan, and fill in Description, Price.



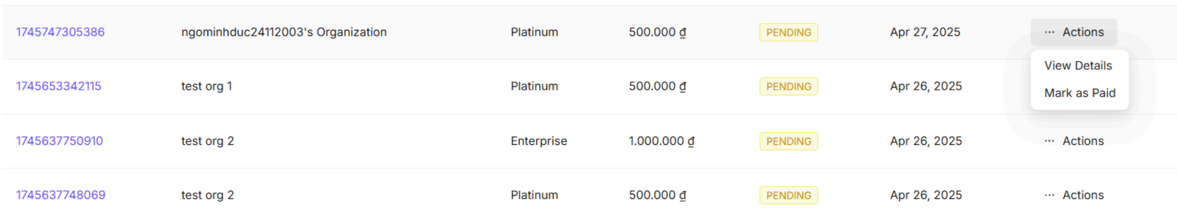
- Admin click "Create Invoice" button on "Create New Invoice" Dialog



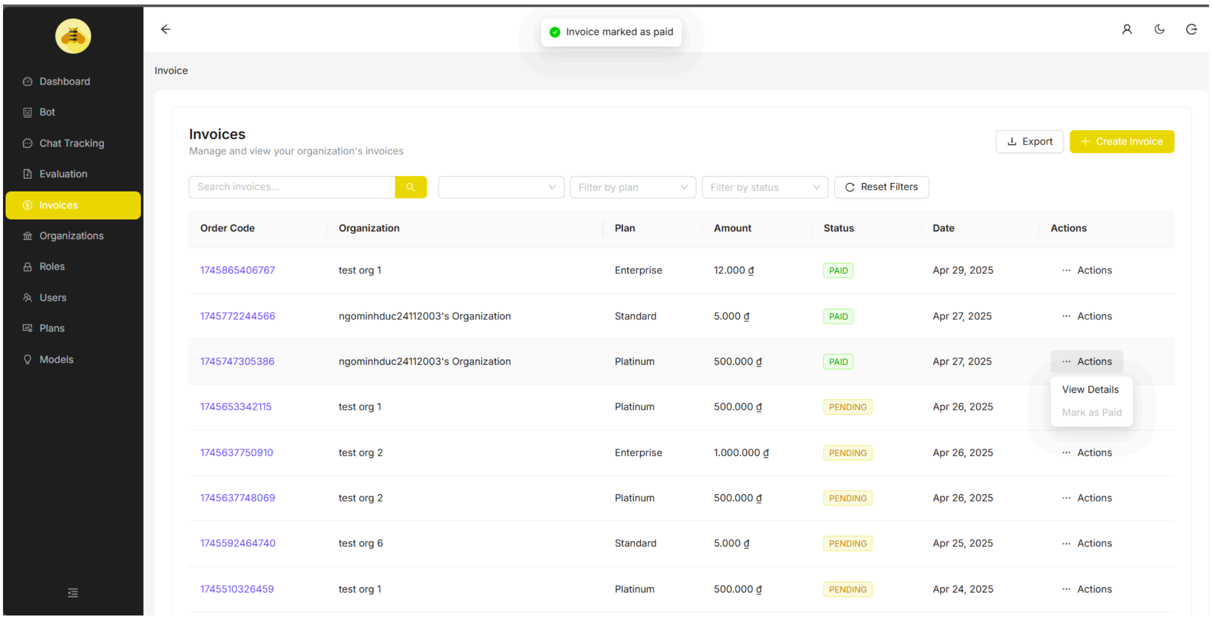
3.6.2.2 Update Invoice

Description: Admin wants to change Status “PENDING” to “PAID” of an Invoice.

- Admin hover to field Actions of an pending-status Invoice that need to be updated.



- Admin click “Mark as Paid”

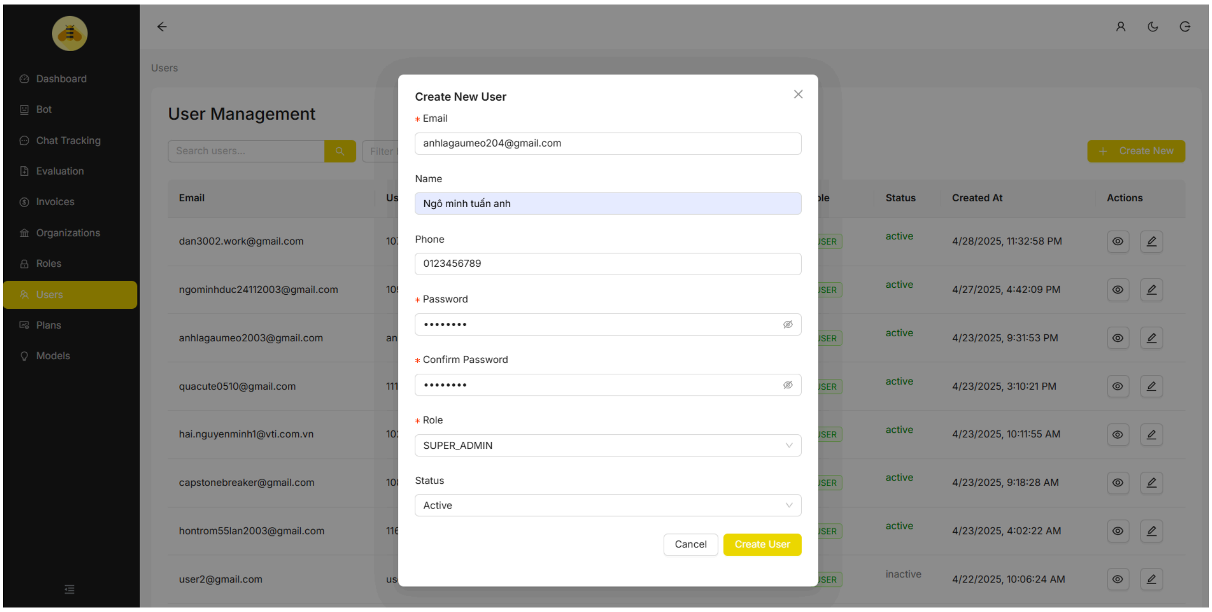


3.6.3 User Management

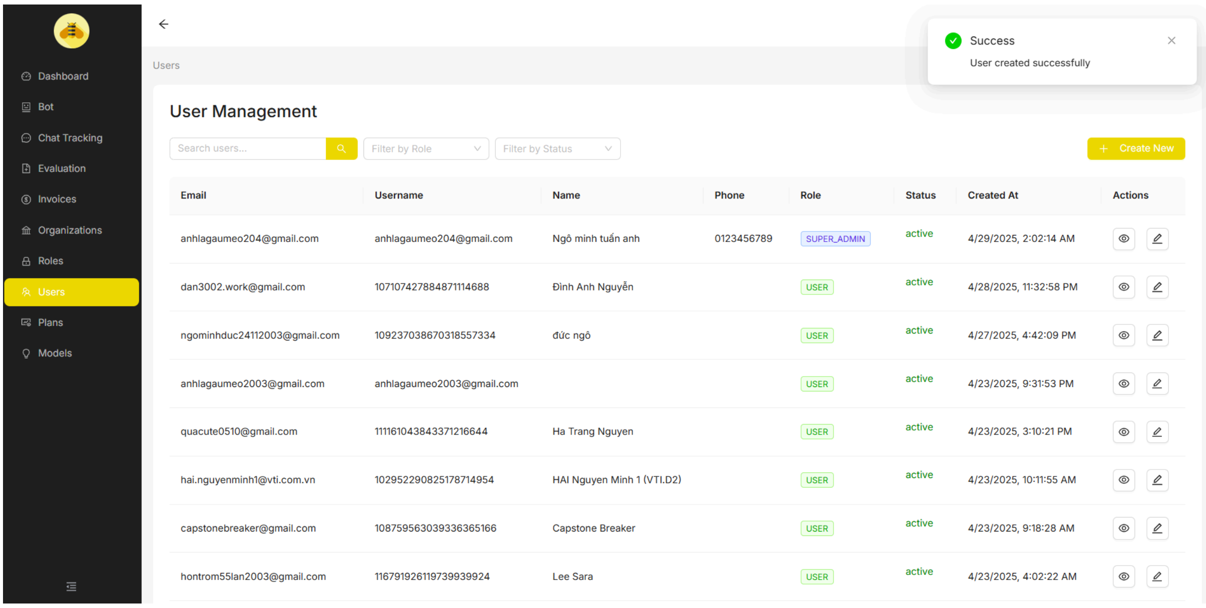
3.6.3.1 Create User

Description: Admin wants to create new account

- After logging into the system, Admin navigates to the Users section.
- Admin click “Create Invoice” button on Table
- Admin fill in new Account information.



- Admin click “Create User” on “Create New User” Dialog

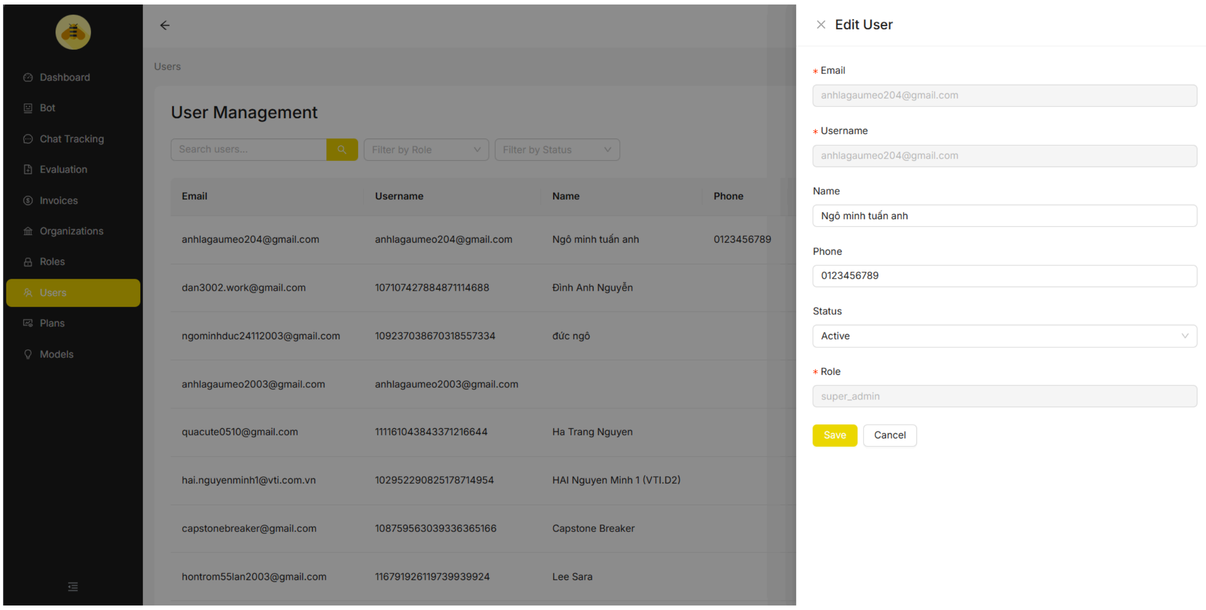


3.6.3.2 Update User

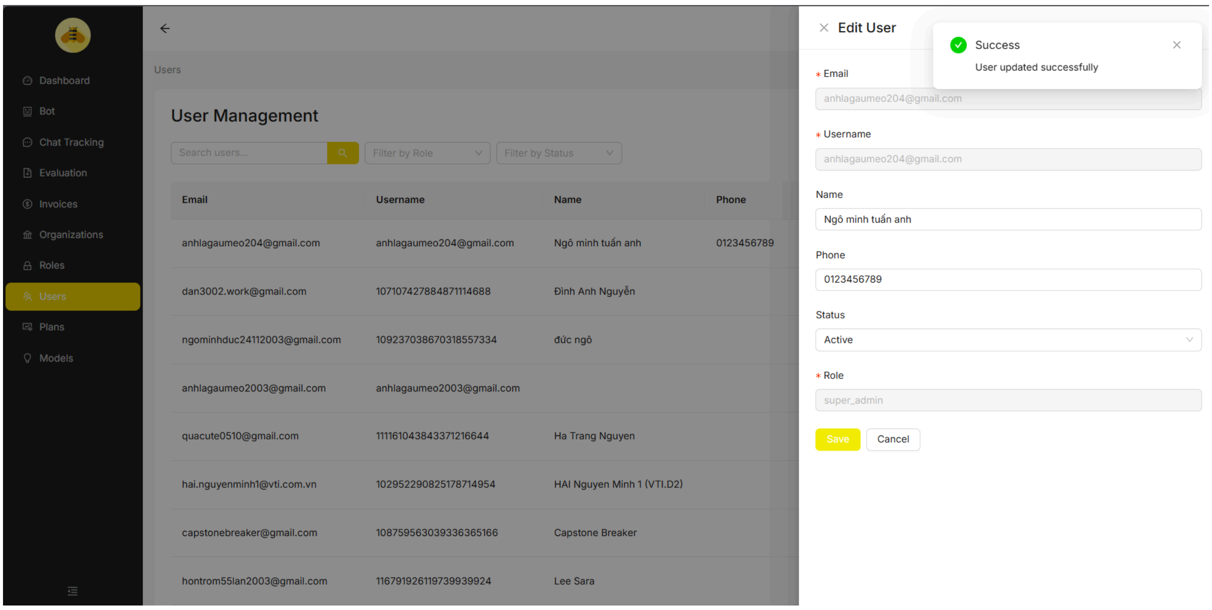
Description: Admin wants to update user

- Admin clicks the editing icon on the user that admin wants to update.

- Admin update user information.



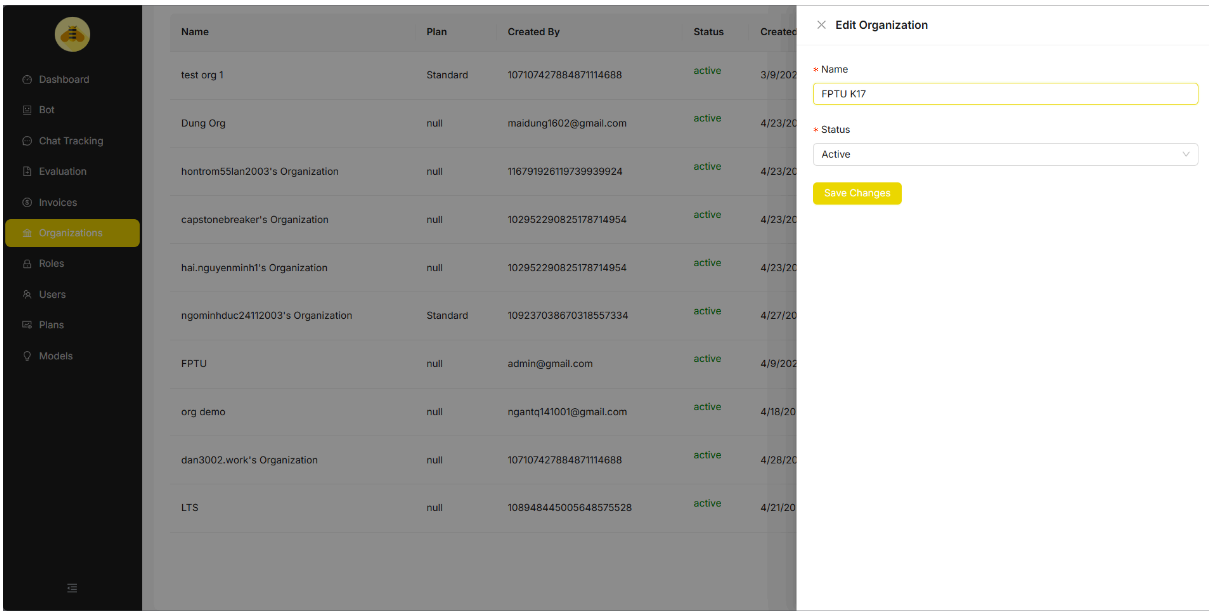
- Admin click "Save" button to update user information



3.6.4 Organization Management

Description: Admin can view and update organization information

- After logging into the system, Admin navigates to the Organization section.
- Admin clicks the editing icon on the Organization that admin wants to update
- Admin change Organization information.



- Admin click “Save Changes” to update Organization.