UniFeast

CAMPUS FOOD RECOMMENDATION APP

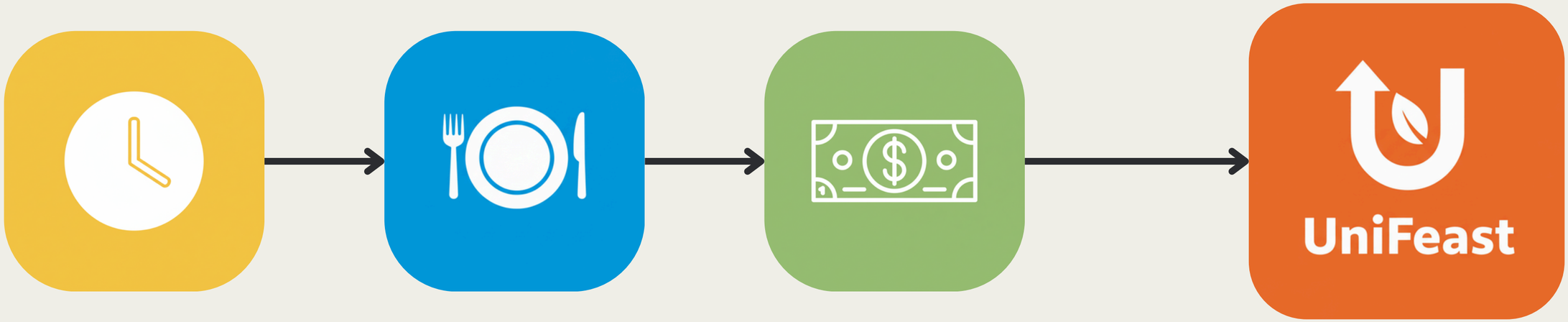
Addressing Campus Dining Challenges with AI-Powered Solutions

Natalie Chain, SDE Track
Qinxuan Li, AI/ML Track
Charlotte, PM Track
Selina, PM Track

amazon aws

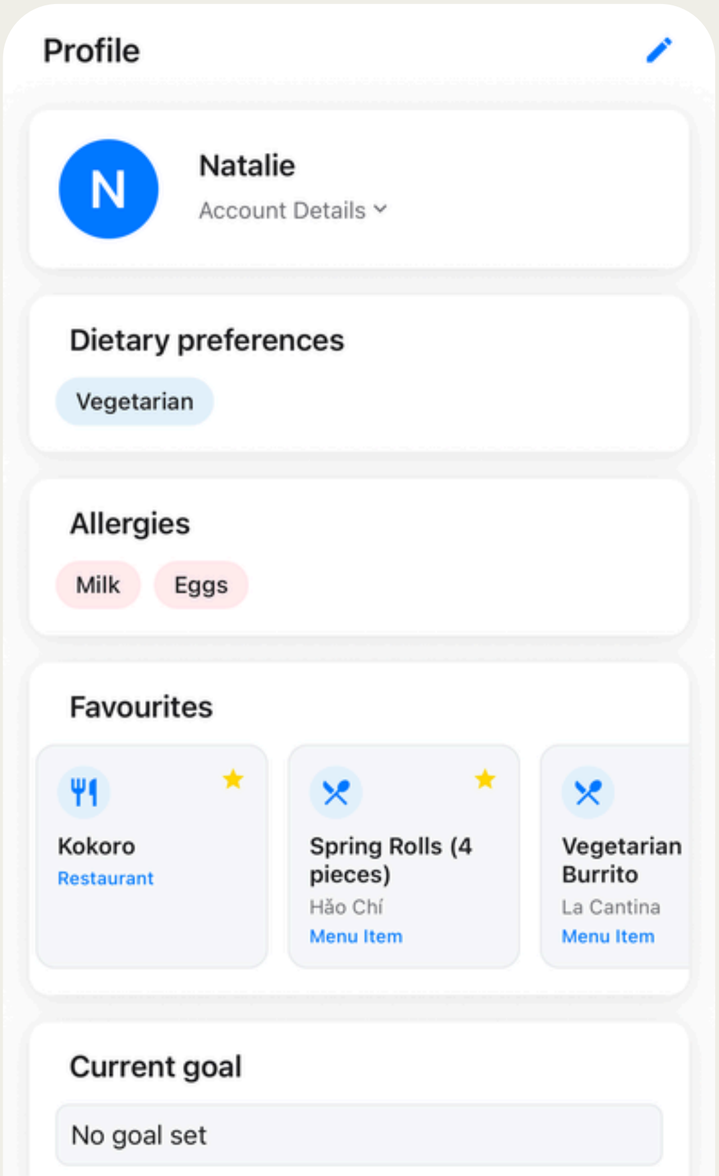
IMPERIAL

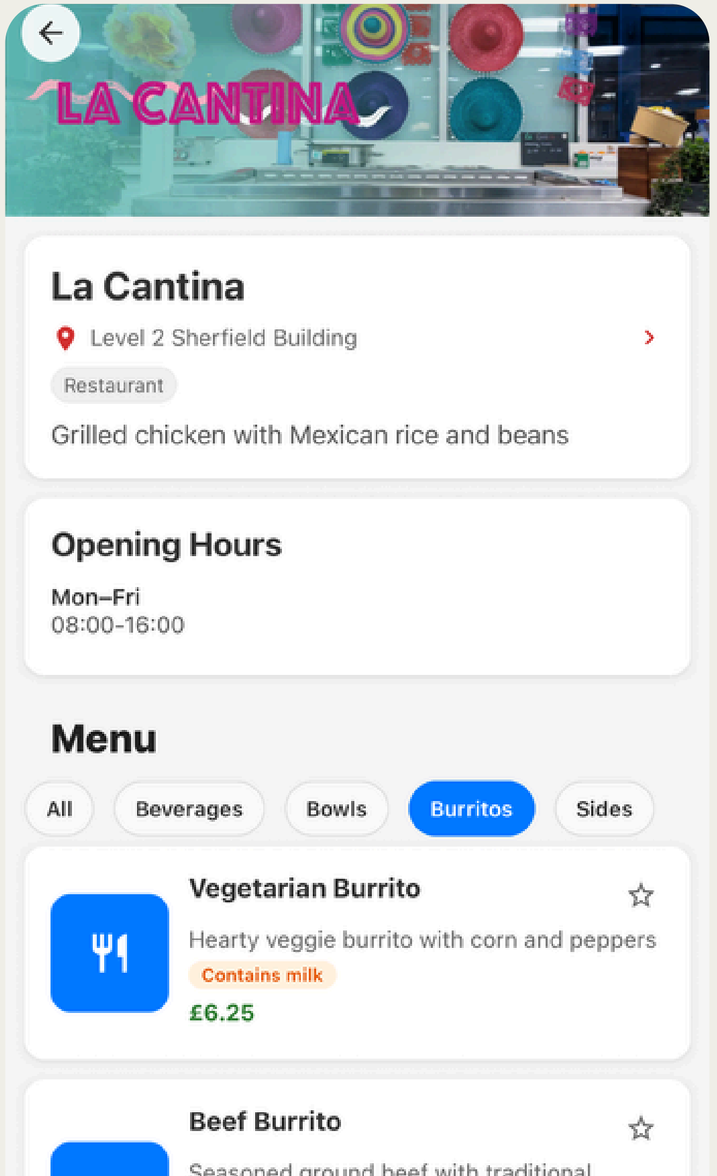
PROBLEM + SOLUTION



aws
IMPERIAL

KEY FEATURES





09:17

Imperial College London

South Kensington Campus

Café | Restaurant | Bar | Open



Kokoro

South Kensington

Closed | Café | Sushi



La Cantina

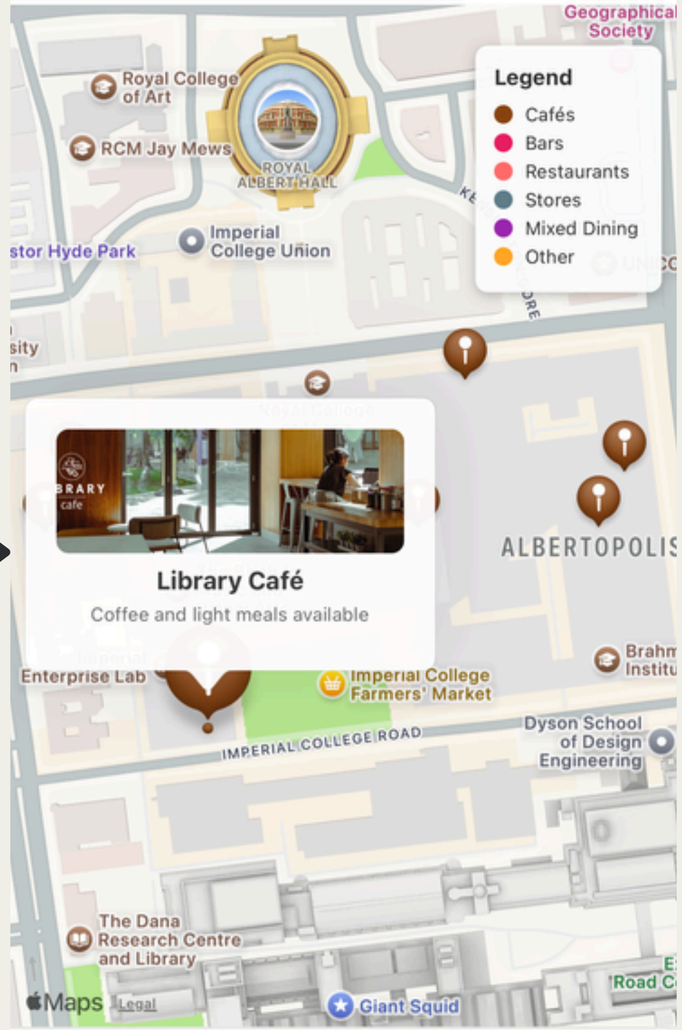
Level 2 Sherfield Building

Open | Restaurant | Mexican



09:19

Imperial College London



Fly to Campus | Show Closed

UniFeast Chatbot

Campus Dining Helper

I want to have Asian food for lunch

09:21

Here are some Asian food options for lunch that are safe for you, considering your vegetarian preference and allergies to milk and eggs:

09:22



Vegetable Sushi Rolls

Kokoro at H-Bar

Fresh sushi rolls filled with assorted vegetables

H-Bar

Lunch

Sushi · Japanese

Dietary:

Vegetarian

Allergens:

gluten

Ask me about food...



Kokoro

South Kensington

Café

Classic sushi roll

Menu

Main Dishes | Small Dishes | Soft Drinks | Sushi

Vegetable Roll

Fresh vegetable sushi

£4.75

California Roll

Classic sushi roll

Contains eggs

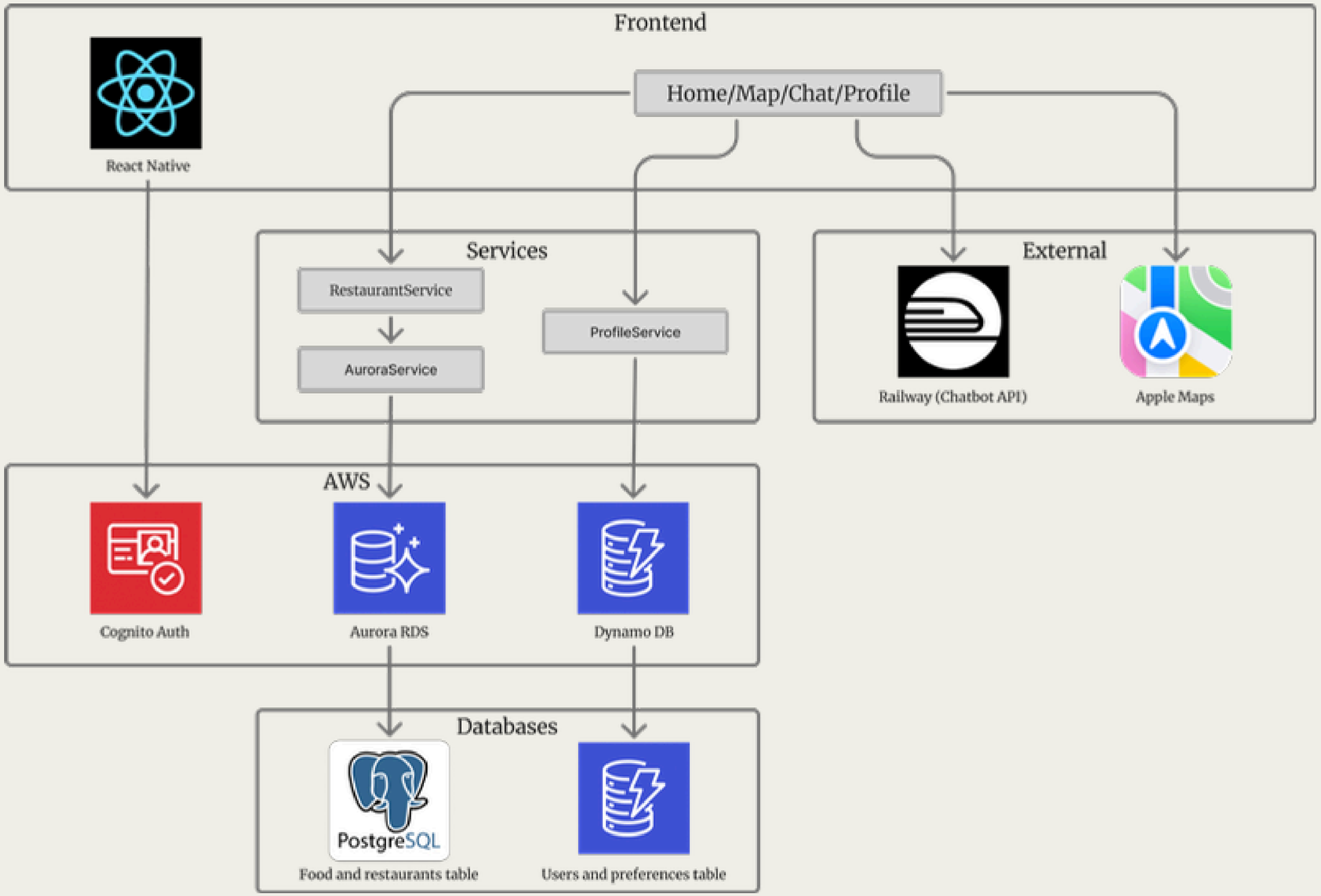
£4.95

Chicken Roll

Cooked chicken option

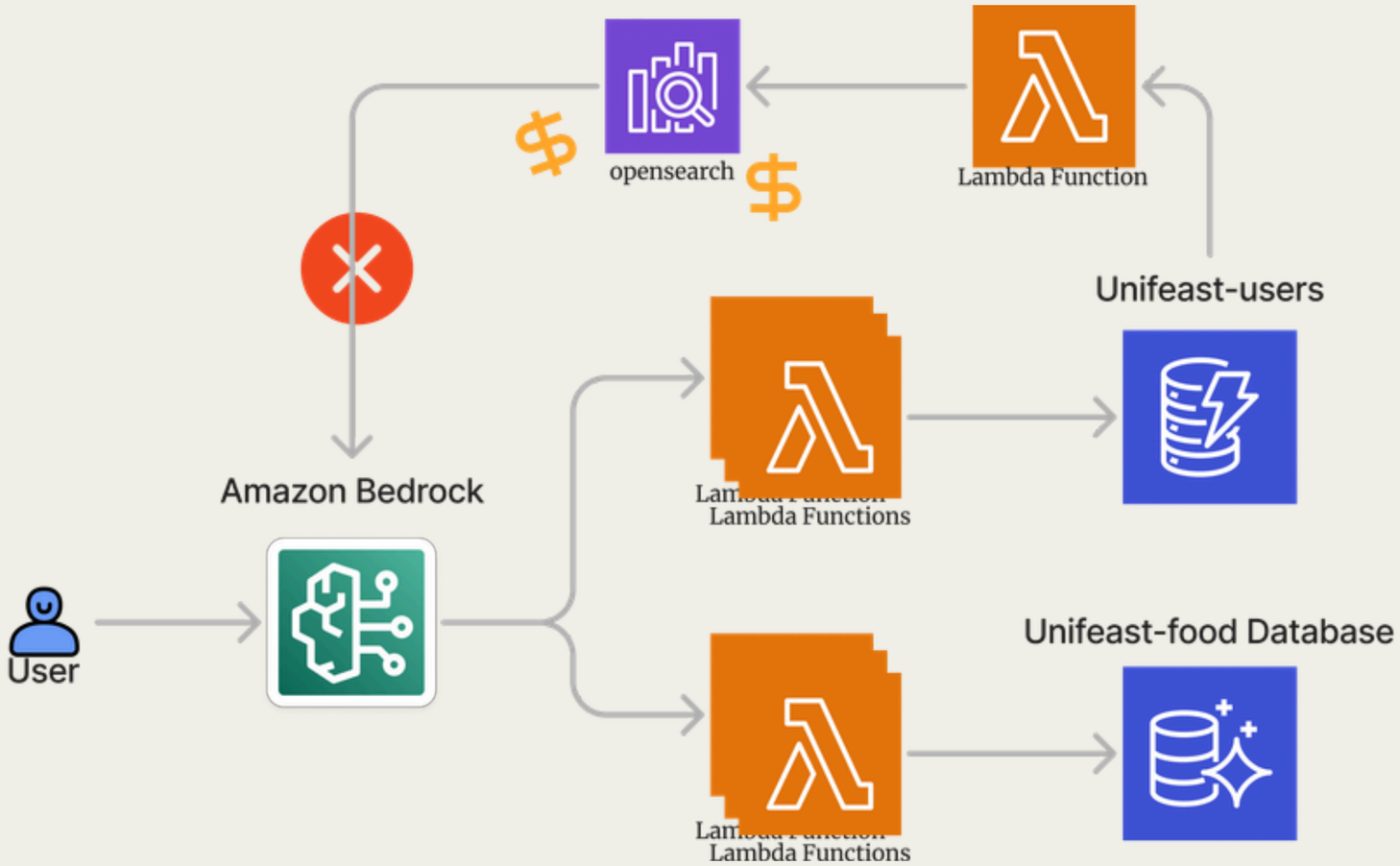
£5.25

APP DESIGN


Frontend
React Native
Home/Map/Chat/Profile
Services
RestaurantService
AuroraService
ProfileService
External
Railway (Chatbot API)
Apple Maps
AWS
Cognito Auth
Aurora RDS
Dynamo DB
Databases
PostgreSQL
Food and restaurants table
Users and preferences table

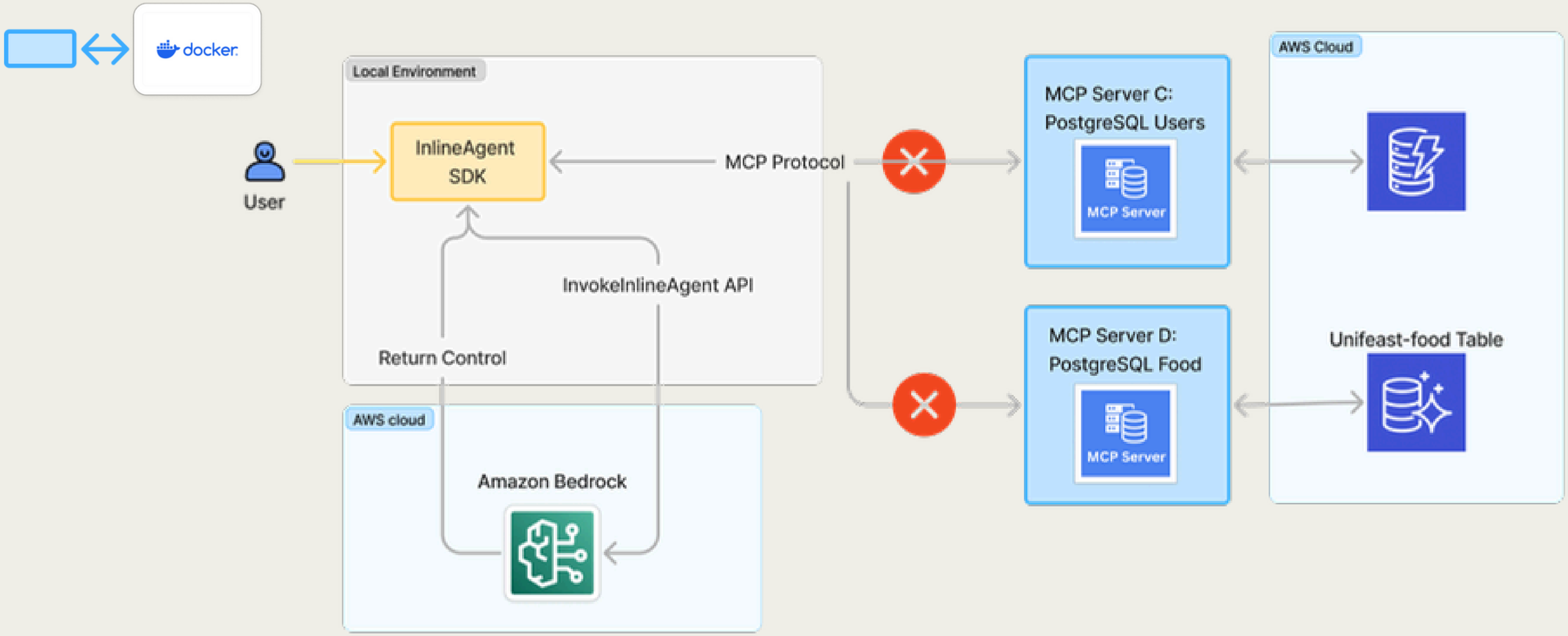
AI CHATBOT DESIGN

Version 1: AWS Bedrock Agent + Lambda + Databases + OpenSearch



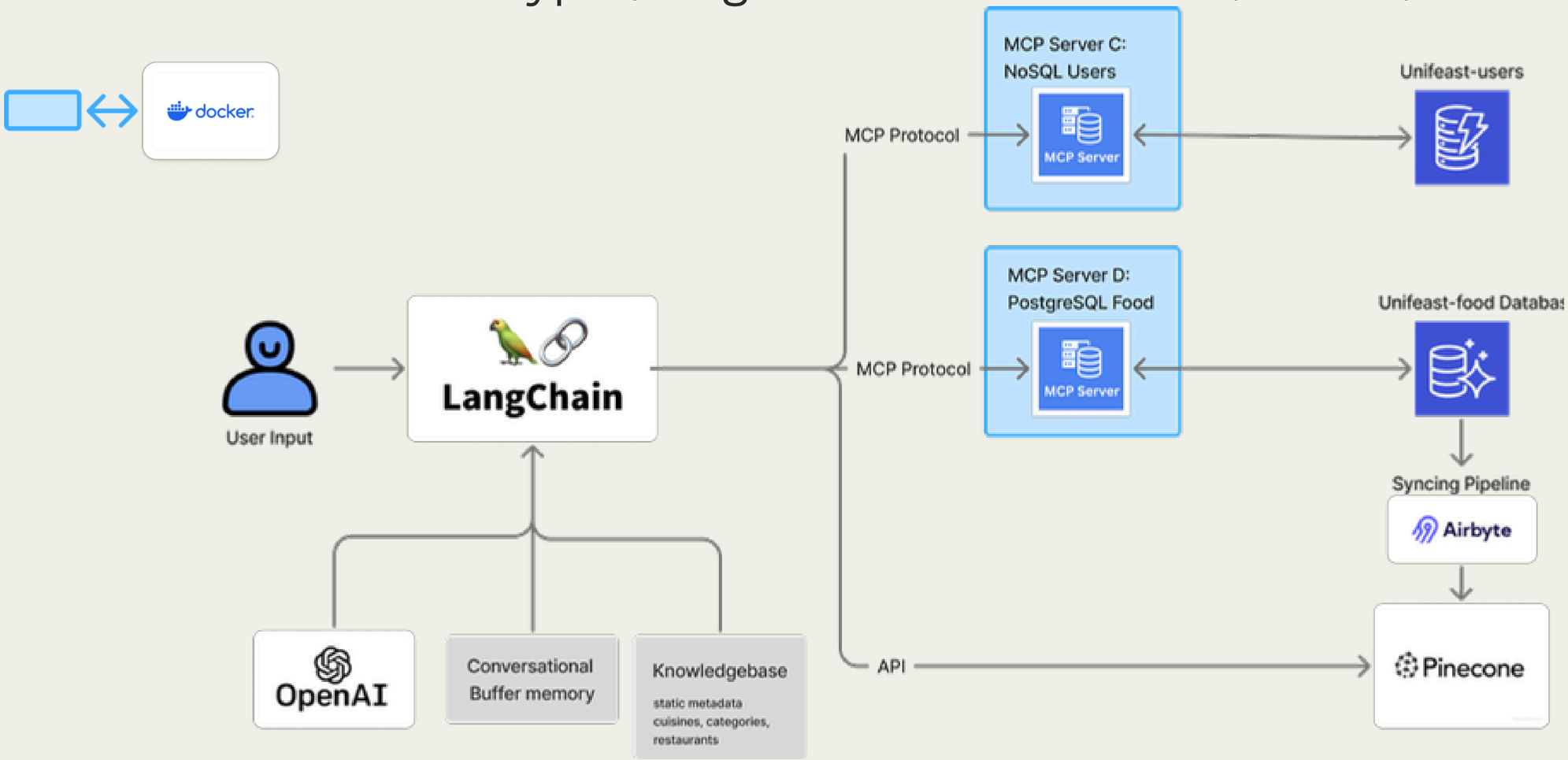
AI CHATBOT DESIGN

Version 2: In Agent SDK + AWS MCP Servers



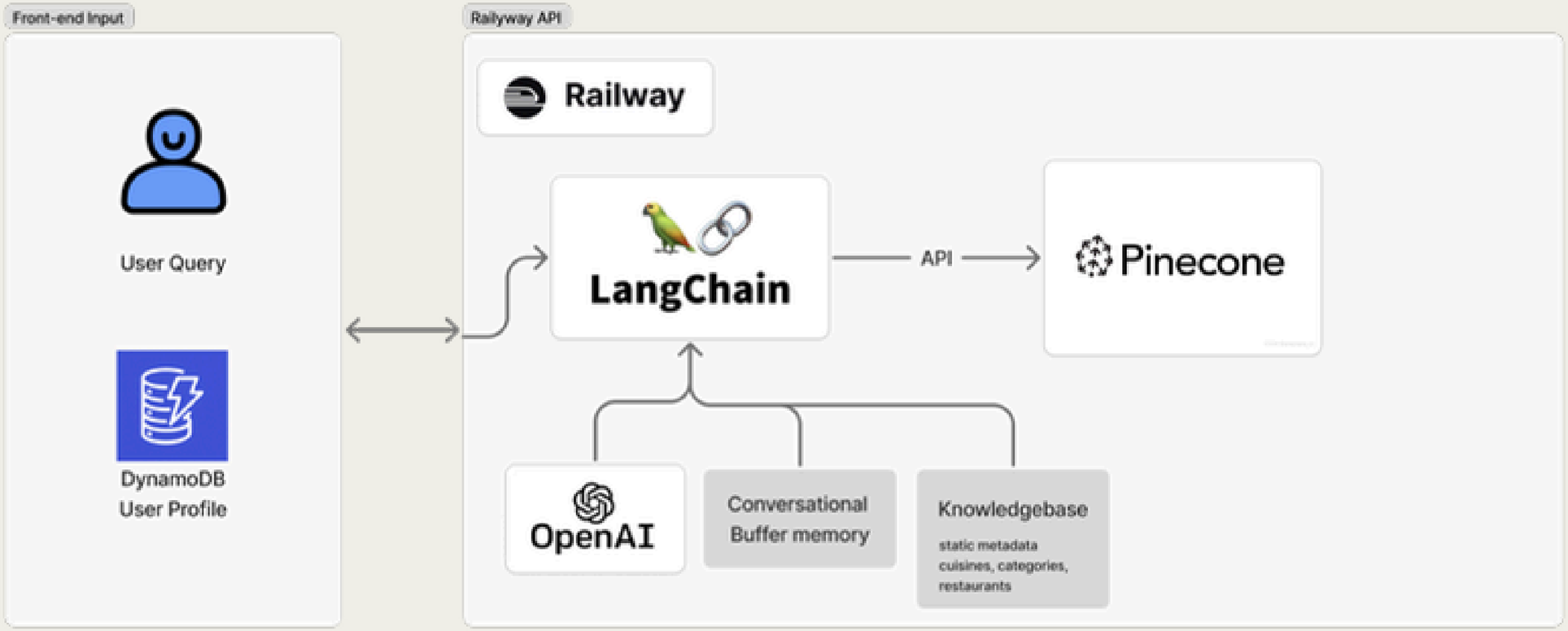
AI CHATBOT DESIGN

Version 3: Local Prototype (LangChain + MCP Servers(Docker) + Pinecone



AI CHATBOT DESIGN

Version 4: Deployed API (Derived from Version 3, MCP Servers Removed)



BACKEND DATABASE DESIGN

Purpose-built schemas for scalability, fast allergy matching, and personalized recommendations

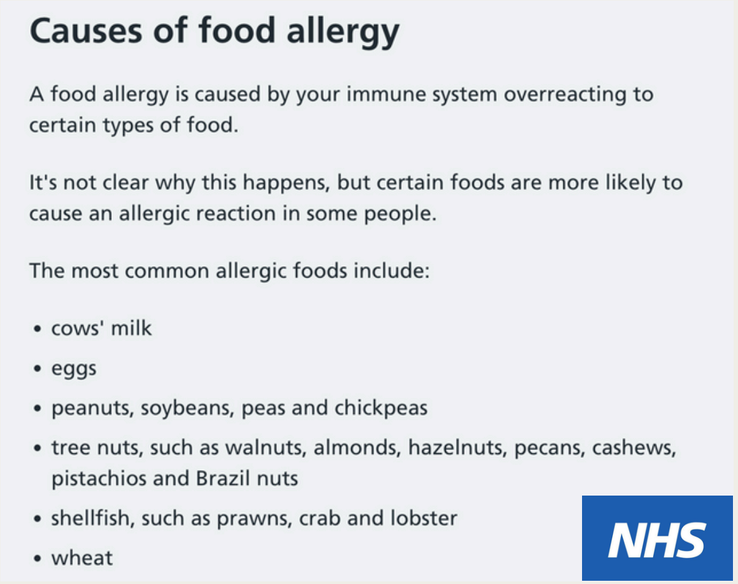
Database Type	AWS Service	Core Use Case	Why This Choice?
User Profiles	**DynamoDB (NoSQL)** 	Store user preferences/allergies	Flexible schema for user data; fast lookups via user_id (Cognito-linked).
Food + Restaurants	**Aurora PostgreSQL (SQL)** 	Manage relational food/restaurant data	Supports complex joins (food ↔ restaurant) and structured filtering (e.g., serve time, cuisine).
Semantic Search Index	**Pinecone (Vector DB)** 	Store vector embeddings of food/restaurant data for "magic search".	Enables semantic similarity (e.g., "creamy pasta without milk" → finds dishes matching meaning, not just keywords).

BACKEND DATABASE DESIGN

User Profile DB (DynamoDB: unifeast-users)

Group	Field Name	Type	Example Value
Keys	user_id	String	"cognito_789012"
Basic Info	user_name	String	"Alex Chen"
	user_identity	String	"Student"
	email	String	"alex.chen@uni.edu"
Preferences	favourites	Array	["Spicy Tofu Stir-Fry", "Vegan Burger"]
	budget	Number	15
	period_plan	String	"Weekly meal prep"
	dietary_preferences	Array	["vegan", "gluten-free", "low-sodium"]
Allergies	eggs_allergy	Boolean	TRUE
	milk_allergy	Boolean	FALSE
	peanuts_allergy	Boolean	TRUE
	shellfish_allergy	Boolean	FALSE
	tree_nuts_allergy	Boolean	FALSE
	other_allergies	Array	["soy", "sesame"]
Metadata	session_data	String	'{"last_login": "2024-10-22", "device": "mobile"}'
	created_at	Timestamp	2024-09-15 10:30:00
	updated_at	Timestamp	2024-10-20 16:45:00

- **Mass-scalable**: Boolean fields eliminate list-scanning bottlenecks for large user bases.
- **NHS-aligned**: Top 5 allergens cover most common cases.



Causes of food allergy

A food allergy is caused by your immune system overreacting to certain types of food.

It's not clear why this happens, but certain foods are more likely to cause an allergic reaction in some people.

The most common allergic foods include:

- cows' milk
- eggs
- peanuts, soybeans, peas and chickpeas
- tree nuts, such as walnuts, almonds, hazelnuts, pecans, cashews, pistachios and Brazil nuts
- shellfish, such as prawns, crab and lobster
- wheat

NHS

BACKEND DATABASE DESIGN

Food DB (PostgreSQL: unifeast_food)

Group	Field Name	Type	Example Value
Keys	id	Integer	456
	restaurant_id	Varchar	"rest_987"
Core Details	dish_name	Varchar	"Spicy Tofu Stir-Fry"
	description	Text	"stir-fry with bell peppers, ginger, chili"
Classifications	category	Varchar	"Main Dish"
	food_type	Varchar	"Vegan"
	cuisine_type	Varchar	"Asian"
Allergens	ingredients	Array	["tofu", "bell peppers", "soy sauce"]
	dietary_tags	Array	["soy-based", "spicy"]
	other_allergens	Array	["sesame oil"]
	milk_allergy	Boolean	FALSE
	eggs_allergy	Boolean	FALSE
	peanuts_allergy	Boolean	FALSE
	tree_nuts_allergy	Boolean	FALSE
	shellfish_allergy	Boolean	FALSE

Group	Field Name	Type	Example Value
Pricing	student_price	Numeric	7.5
	staff_price	Numeric	9
Context	serve_time	Varchar	"Lunch, Dinner"
	location	Varchar	"Campus Dining Hall, Station 3"
	restaurant_name	Varchar	"Global Eats"
	opening_hours	Varchar	"11:00 AM – 7:00 PM"
	accessibility	Varchar	"Wheelchair accessible"
Metadata	notes	Text	"Spice level: medium"
	available	Boolean	TRUE
	embedding_id	Varchar	"embed_456_20241022"
	created_at/updated_at	Timestamp	2024-10-22 14:30:00

BACKEND DATABASE DESIGN

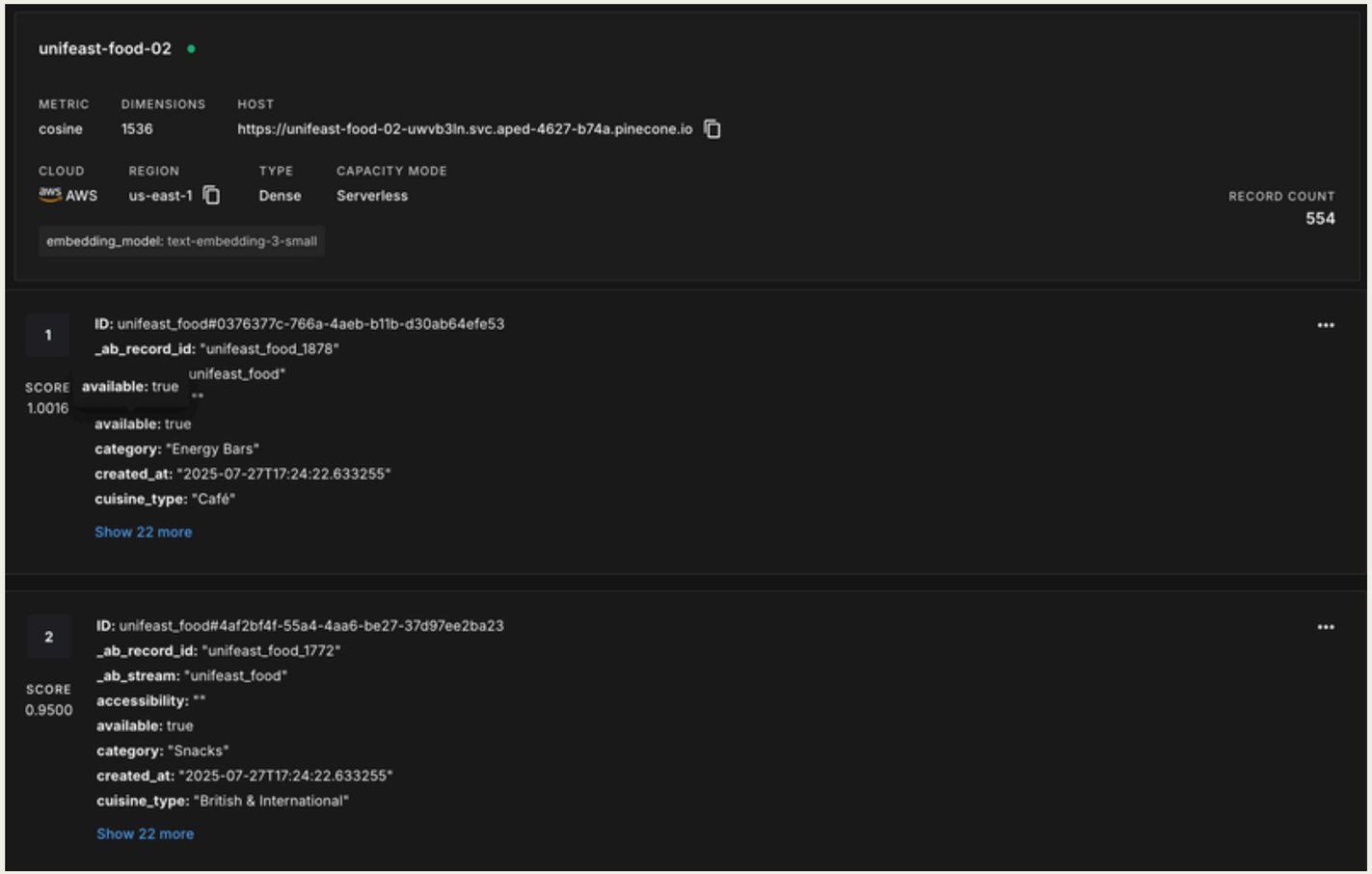
unifeast_restaurants (Venue-Level Data)

Group	Field Name	Type	Example Value
Key	restaurant_id	Varchar	"rest_987"
Core Info	record_type	Varchar	"On-Campus Dining"
	restaurant_name	Varchar	"Global Eats"
	description	Text	"International cuisine spot with vegan options"
Context	location	Varchar	"Student Union, 2nd Floor"
	opening_hours	Varchar	"Mon-Fri: 8AM-8PM; Sat-Sun: 10AM-6PM"
	cuisine_type	Varchar	"Multi-Cuisine"
	contact	Varchar	"dining.global@uni.edu"
	accessibility	Varchar	"Elevator access, wheelchair ramps"
Media	banner_url	Varchar	"https://uni-cdn.com/global-eats-banner.jpg"
	banner_position	Varchar	"left center" (default)
Metadata	notes	Text	"Accepts meal swipes"
	created_at/updated_at	Timestamp	2024-09-01 09:00:00
Group	**Field Name**	**Type**	**Example Value**

Group	Field Name	Type	Example Value
Key	restaurant_id	Varchar	"rest_987"
Core Info	record_type	Varchar	"On-Campus Dining"
	restaurant_name	Varchar	"Global Eats"
	description	Text	"International cuisine spot with vegan options"
Context	location	Varchar	"Student Union, 2nd Floor"
	opening_hours	Varchar	"Mon-Fri: 8AM-8PM; Sat-Sun: 10AM-6PM"
	cuisine_type	Varchar	"Multi-Cuisine"
	contact	Varchar	"dining.global@uni.edu"
	accessibility	Varchar	"Elevator access, wheelchair ramps"
Media	banner_url	Varchar	"https://uni-cdn.com/global-eats-banner.jpg"
	banner_position	Varchar	"left center" (default)
Metadata	notes	Text	"Accepts meal swipes"

BACKEND DATABASE DESIGN

Pinecone Index (Vector Embeding)



1. Syncs vector embeddings from original database; auto-updates with changes.



2. Enables **semantic search** for agents to interpret natural queries.

FUTURE DEVELOPMENT

RECOMMENDATION ACCURACY



AI AGENT & MCP FIX



CROWDEDNESS DETECTION



UIUX IMPROVEMENT



REAL DATA INTEGRATION



SHORT TIPS FORUM



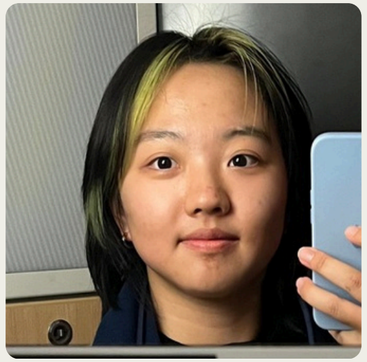
PROJECT MANAGEMENT & COLLABORATION



SDE Track

Natalie Chan

- Product Management
- Frontend (React native)
- Backend integration
- UI/UX



AI/ML Track

Qinxuan Li

- Product Management
- AI Agent Development
- Backend Database Setup
- UIUX



PM Track

Charlotte Lo

Business Research



PM Track

Chang Gao

UI/UX

Key Challenges

- **Limited manpower** → Reduced features to stay on track.
- **MCP-API integration failure** → Frontend-to-agent data routing enabled personalization.

IMPERIAL amazon aws

Thank you!

