



**Cairo University**

**Faculty of Graduate Studies for Statistical Research**

**Department of Computer and Information Sciences**

CAIRO UNIVERSITY Faculty of Graduate Studies for Statistical Research Department of Computer and Information Sciences

Rahal – The AI-Powered Tourist Guide Software Documentation for Fulfillment of the Diploma Project in Computer Science

**Submitted by:**

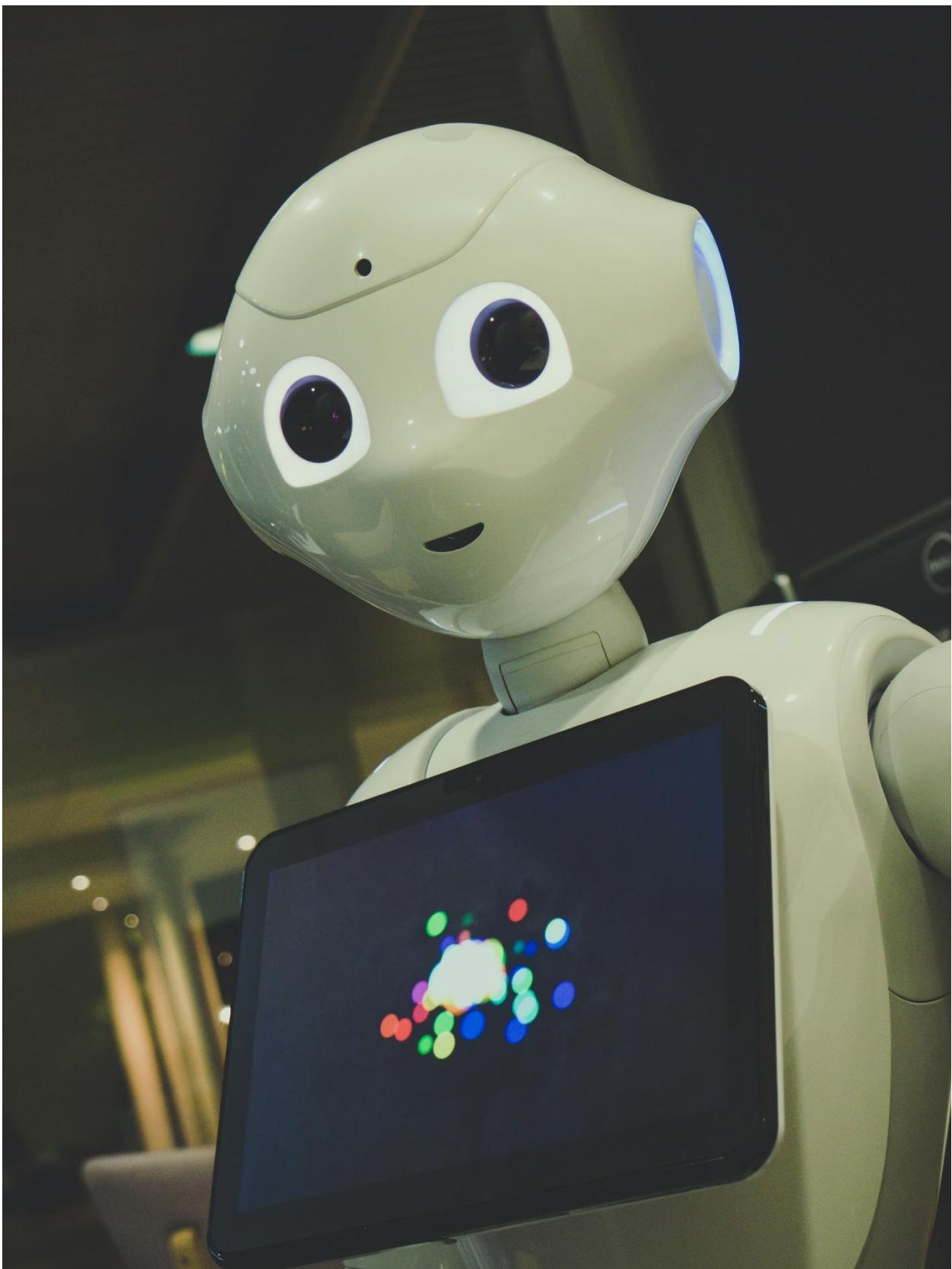
- 1. Hani Abdalbasset ElShafei**
- 2. Muhammad Tarek Ahmad**
- 3. Rania Adel Ibrahim Ibrahim**
- 4. Bashayer Thyab AlFadhli**

**Supervised by:**

**Supervisor Name**

**Cairo, Egypt**

**April 2025**



**RAHAL IN THE FUTURE**

## **1. Introduction:**

In today's fast-paced digital world, tourists are increasingly dependent on mobile applications and smart technologies to enhance their travel experiences. Despite the abundance of information available online, tourists still face fragmented systems that fail to provide personalized, real-time assistance. This is where "Rahal" steps in — a conversational AI-powered tourist guide designed to deliver contextual, location-based information about historical landmarks, local cuisine, directions, and more.

Rahal is built using advanced AI methodologies such as Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs). It offers a scalable, multilingual platform capable of interacting with users through natural language to suggest optimal travel itineraries, recommend nearby restaurants, and provide educational insights about tourist attractions.

The system is accessible across multiple platforms including web applications and messaging apps like Telegram and WhatsApp, making it a practical and intuitive solution for travelers on the go. With this documentation, we aim to provide a complete technical overview of the Rahal system, from architecture and features to use cases, diagrams, testing, and implementation plans.

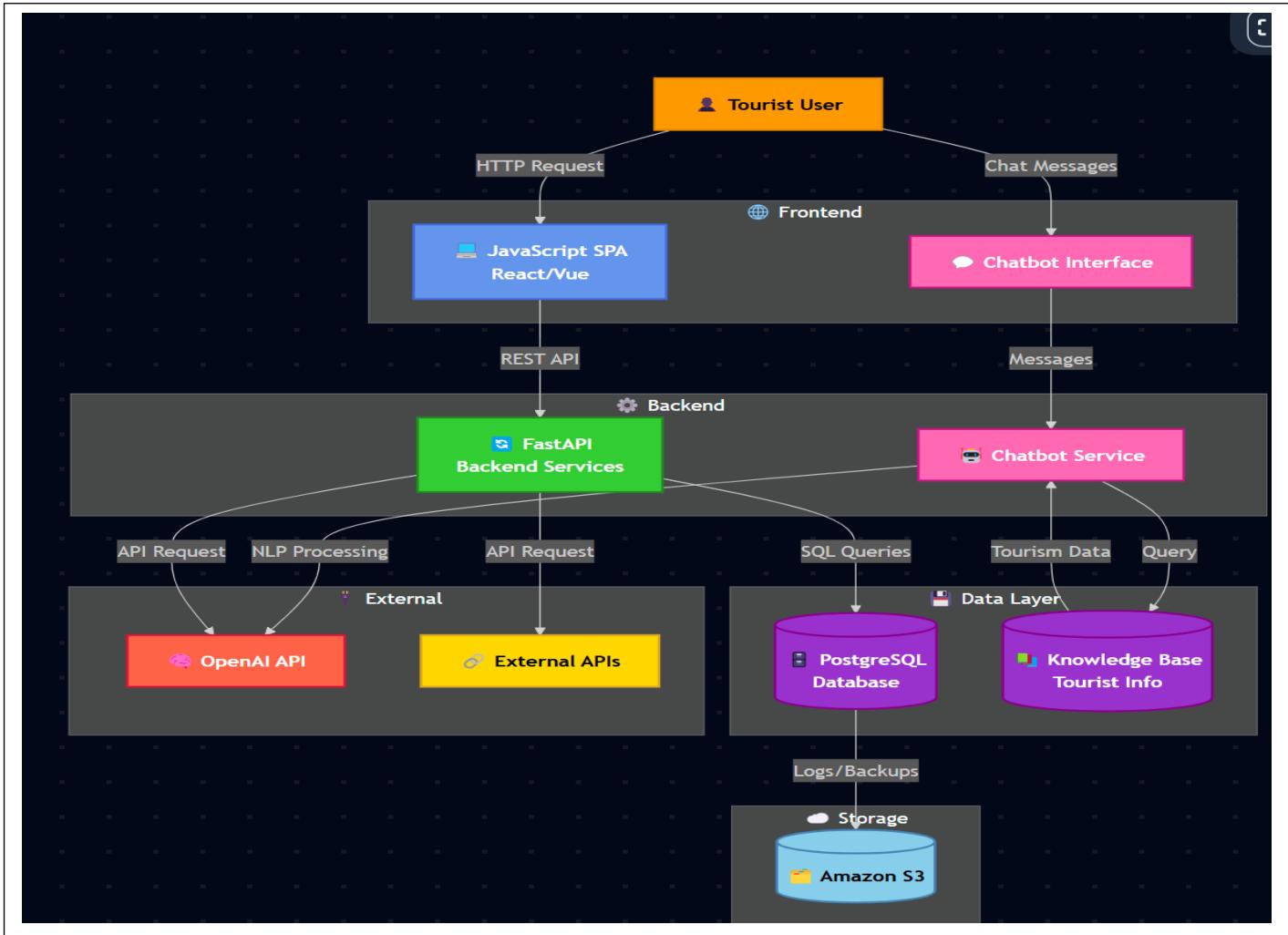
**Scope** Rahal aims to provide real-time, location-aware assistance to tourists, enhancing their experience with historical, cultural, and culinary insights.

**Purpose** To centralize travel information, reduce reliance on multiple fragmented apps, and guide users conversationally via AI.

**System Architecture** Rahal is composed of three main layers:

- Retrieval Layer (e.g., FAISS, Chroma)
- Generation Layer (GPT-4, LLaMA)
- Agent Layer (LangChain/AutoGen)  
Front-end: React / Streamlit  
Back-end: FastAPI

## System architecture:



The presented system architecture effectively combines modern web technologies with AI capabilities to create a tourist-focused platform. The architecture demonstrates strong design principles with clear separation of concerns and well-defined communication pathways.

## System requirements specification

### A-Functional Requirements

#### Requirement specifications

		The system shall be scalable on all devices
REQ2		The system shall provide authentic access
REQ3		The system shall prevent unauthorized users from logging in
REQ4		The system shall save user information used to login
REQ5		The system shall allow user to update their profile and information
REQ6		The system shall provide a responsive AI model that is easy to use
REQ7		The system shall allow user to enter his destination
REQ8		The system shall allow user to enter his location
REQ9		The system shall allow user to enter his preferences
REQ10		The system shall provide trips according to user preferences
REQ11		The system shall save user trips and preferences history
REQ12		The system shall provide attraction places filtered (ex: restaurants) according to user preferences
REQ13		The system shall view attraction place details
REQ14		The system shall view attraction place direction on map
REQ15		The system shall view attraction place price list
REQ16		The system shall view attraction place reviews
REQ17		The system shall view attraction place work hours
REQ18		The system shall view attraction place photo (if available)
REQ19		The system shall view comparison between many attraction places
REQ20		The system shall view attraction place activities
REQ21		The system shall provide user with weather information and advises

## User Stories

Identifier	User story	Size
ST-1	As an authorized user, I can access this website from any device	
ST-2	As an authorized user, I can create profile	
ST-3	As an authorized user, I update my information	
ST-4	As an authorized user, I can view tourist places	
ST-5	As an authorized user, I can view attraction places (ex: restaurants ,hotels)	
ST-6	As an authorized user, I can view attraction places on map	
ST-7	As an authorized user, I can view attraction places according to my preferences	
ST-8	As an authorized user, I can view attraction places price list	
ST-9	As an authorized user, I can view attraction places activities	
ST-10	As an authorized user, I can view comparison between attraction places	
ST-11	As an authorized user, I can view attraction places reviews	
ST-12	As an authorized user, I can view attraction places work hours	
ST-13	As an authorized user, I can view attraction places photos (if available)	
ST-14	As an authorized user, I can get weather advises according to weather state	

## Work backlog

Work item	User story	Iteration number	Estimated work duration
1	ST-1 Access website	1	
2	ST-2 Create profile	1	
3	ST-3 Update profile	1	
4	ST-4 View tourist places	2	
5	ST-5 View attraction places	2	
6	ST-6 View attraction places on map	3	
7	ST-7 View places according to user preferences	4	
8	ST-8 View places price list	5	
9	ST-9 View places activities	5	
10	ST-10 View comparison between places	5	
11	ST-11 View places reviews	5	
12	ST-12 View places work hours	5	
13	ST-13 View places photos	5	
14	ST-14 Weather advises	6	

>Add icon Add description

# Rahal

Project Timeline Backlog +

Open in Calendar Month Today

April 2025

Define project backlog April 6, 2025 → April 7, 2025 Done 100 RAH-2

Set up project Repository & Virtual Env April 6, 2025 → April 11, 2025 Done 100 RAH-1

Setup Docker files For Database Container and Fastapi April 6, 2025 → April 11, 2025 Done 100 RAH-4

Pois Data Collection April 12, 2025 → April 16, 2025 Done 80 RAH-5

Restaurants Points Data Cleaning April 12, 2025 → April 16, 2025 Done 90 RAH-6

Create Database models according to ERD using SQLAlchemy f... April 14, 2025 → April 18, 2025 Done 90 RAH-8

Uploading Cleaned Restaurant Data to Database April 17, 2025 → April 21, 2025 Done 100 RAH-9

Create JWT Authentications and Tokens For User logging u April 18, 2025 → April 22, 2025 Done 100 RAH-10

Creating To visit app API April 21, 2025 → April 25, 2025 Done 100 RAH-11

Create LLM App API April 27, 2025 → May 5, 2025 Not started RAH-12

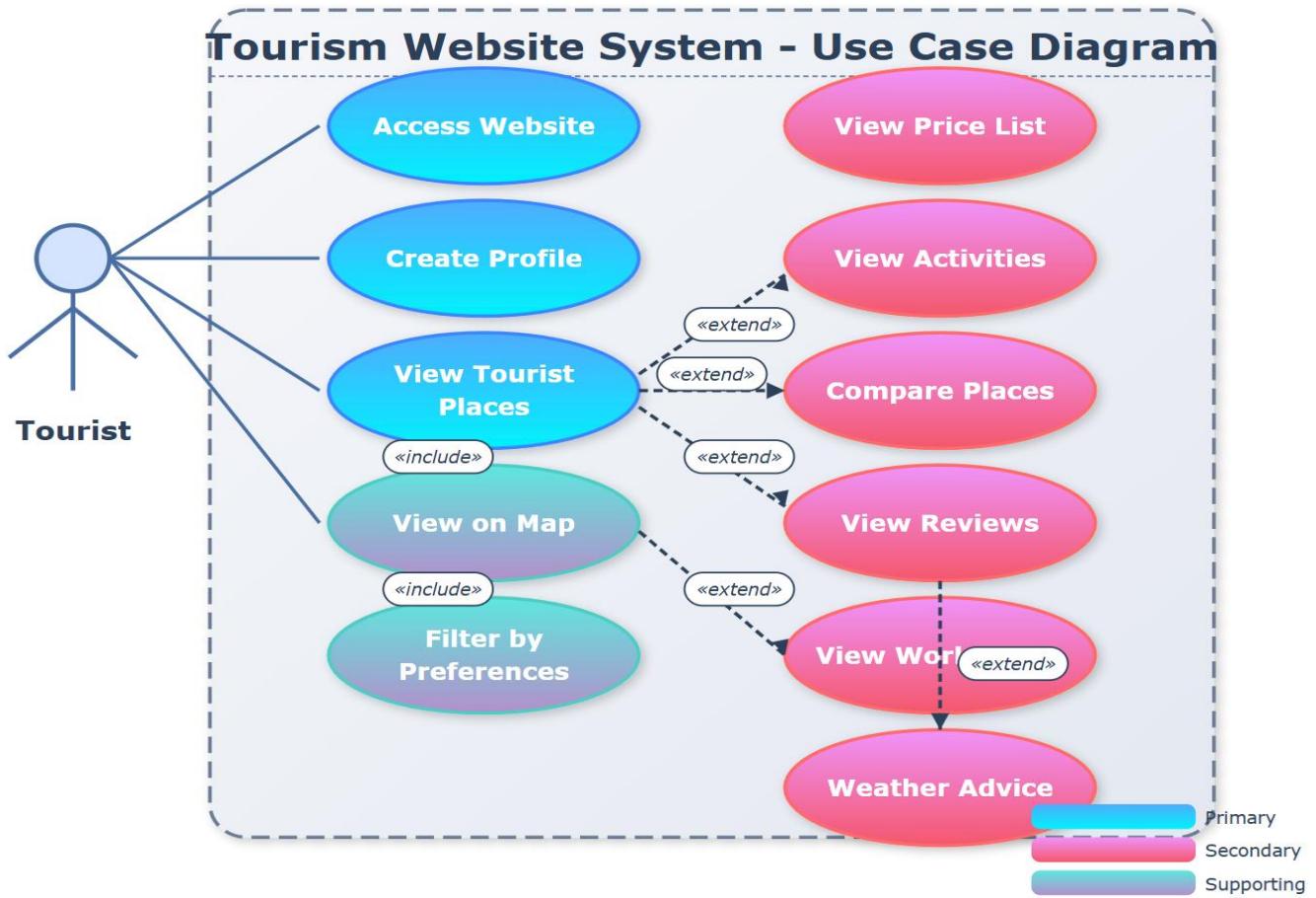
Frontend Pages Design April 14, 2025 → April 30, 2025 Not started RAH-7

3

## Project duration

Phase	Start Date	End Date	Duration
<span>💡</span> Research & Planning	Apr 06, 2025	Apr 20, 2025	2 weeks
<span>🔧</span> System Design & Architecture	Apr 21, 2025	Apr 30, 2025	1.5 weeks
<span>📝</span> Backend & API Development	May 01, 2025	May 21, 2025	3 weeks
<span>🎨</span> Frontend Development	May 07, 2025	May 26, 2025	2.5 weeks
<span>🧬</span> Integration & Testing	May 27, 2025	Jun 03, 2025	1 week
<span>🚀</span> Deployment & Documentation	Jun 04, 2025	Jun 08, 2025	5 days
<span>📦</span> Final Delivery	Jun 09, 2025	Jun 10, 2025	2 days
<span>📅</span> Total Project Duration	Apr 06, 2025	Jun 10, 2025	~9 weeks

Use case diagram:



**Actor Description:**

Actor	Actor's Goal	Use case name
User	Access the tourism website	Access website – (UC-1)
User	Create a personal profile	Create profile – (UC-2)
User	View available tourist places	View tourist places – (UC-3)
User	View tourist places on a map	View on map – (UC-4)
User	Filter tourist places by preferences	Filter by preferences – (UC-5)
User	View price list of tourist places	View price list – (UC-6)
User	View activities available at places	View activities – (UC-7)
User	Compare different tourist places	Compare places – (UC-8)
User	Read reviews from other users	View reviews – (UC-9)
User	Check working hours of places	View work hours – (UC-10)
User	Get weather-related advice	Weather advice – (UC-11)

**Use Case Description:**

Element	Description
Related Requirements	REQ12, REQ13, REQ14
Initiating Actor	Authorized User
Actor's Goal	To view a list of tourist places filtered by preferences and displayed on a map.
Participating Actors	System
Preconditions	- The user is logged in.

	- The system contains data about tourist places.
Postconditions	The user views a list of tourist places and their map location.
Flow of Events for Main Success Scenario	<ol style="list-style-type: none"> <li>1. User selects "View Tourist Places".</li> <li>2. System retrieves the list based on preferences.</li> <li>3. System displays the places on the screen.</li> <li>4. User clicks a place to see it on the map.</li> </ol>
Flow of Events for Extensions (Alternate Scenarios)	<ol style="list-style-type: none"> <li>2a. No places found for preferences -&gt; System shows "No results found".</li> <li>4a. Map fails to load -&gt; System displays text-only details.</li> </ol>

## Traceability Matrix

Requirement ID	Requirement Description	User Story ID	Use Case ID
REQ1	The system shall be scalable on all devices	ST-1	UC-1
REQ2	The system shall provide authentic access	ST-1	UC-1
REQ3	The system shall prevent unauthorized users from logging in	ST-1	UC-1
REQ4	The system shall save user information used to login	ST-2	UC-2
REQ5	The system shall allow user to update their profile and information	ST-3	UC-2
REQ6	The system shall provide a responsive AI model	ST-4, ST-14	UC-3, UC-11
REQ7	The system shall allow user to enter his destination	ST-4	UC-3
REQ8	The system shall allow user to enter his location	ST-4	UC-3
REQ9	The system shall allow user to enter his preferences	ST-7	UC-5
REQ10	The system shall provide trips according to user preferences	ST-7	UC-5

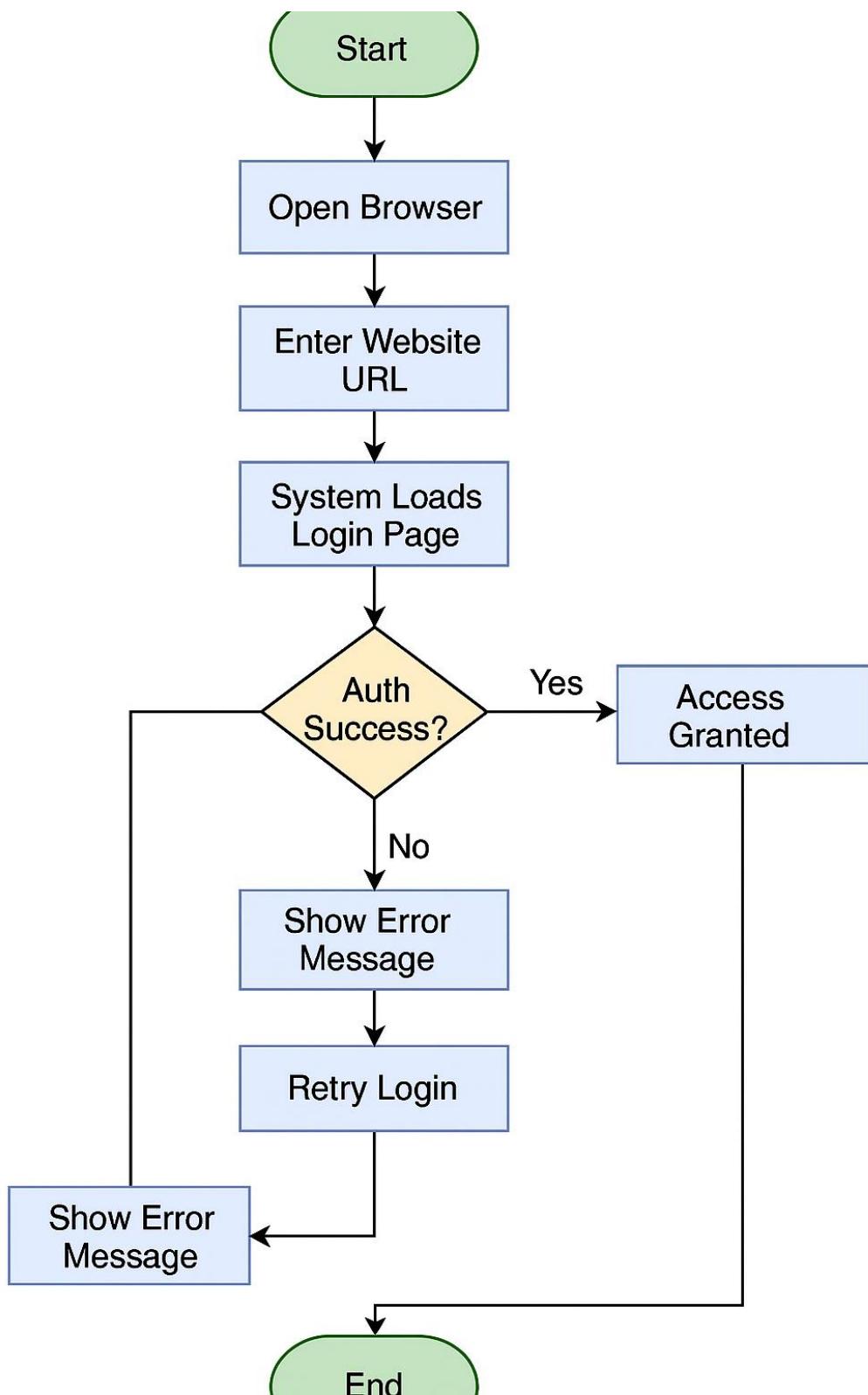
REQ11	The system shall save user trips and preferences history		
REQ12	The system shall provide filtered attraction places	ST-5, ST-7	UC-5
REQ13	The system shall view attraction place details	ST-4, ST-5	UC-3, UC-5
REQ14	The system shall view attraction place direction on map	ST-6	UC-4
REQ15	The system shall view attraction place price list	ST-8	UC-6
REQ16	The system shall view attraction place reviews	ST-11	UC-9
REQ17	The system shall view attraction place work hours	ST-12	UC-10
REQ18	The system shall view attraction place photo	ST-13	UC-3, UC-5
REQ19	The system shall view comparison between many attraction places	ST-10	UC-8
REQ20	The system shall view attraction place activities	ST-9	UC-7
REQ21	The system shall provide user with weather information and advises	ST-14	UC-11



## Activity Diagrams

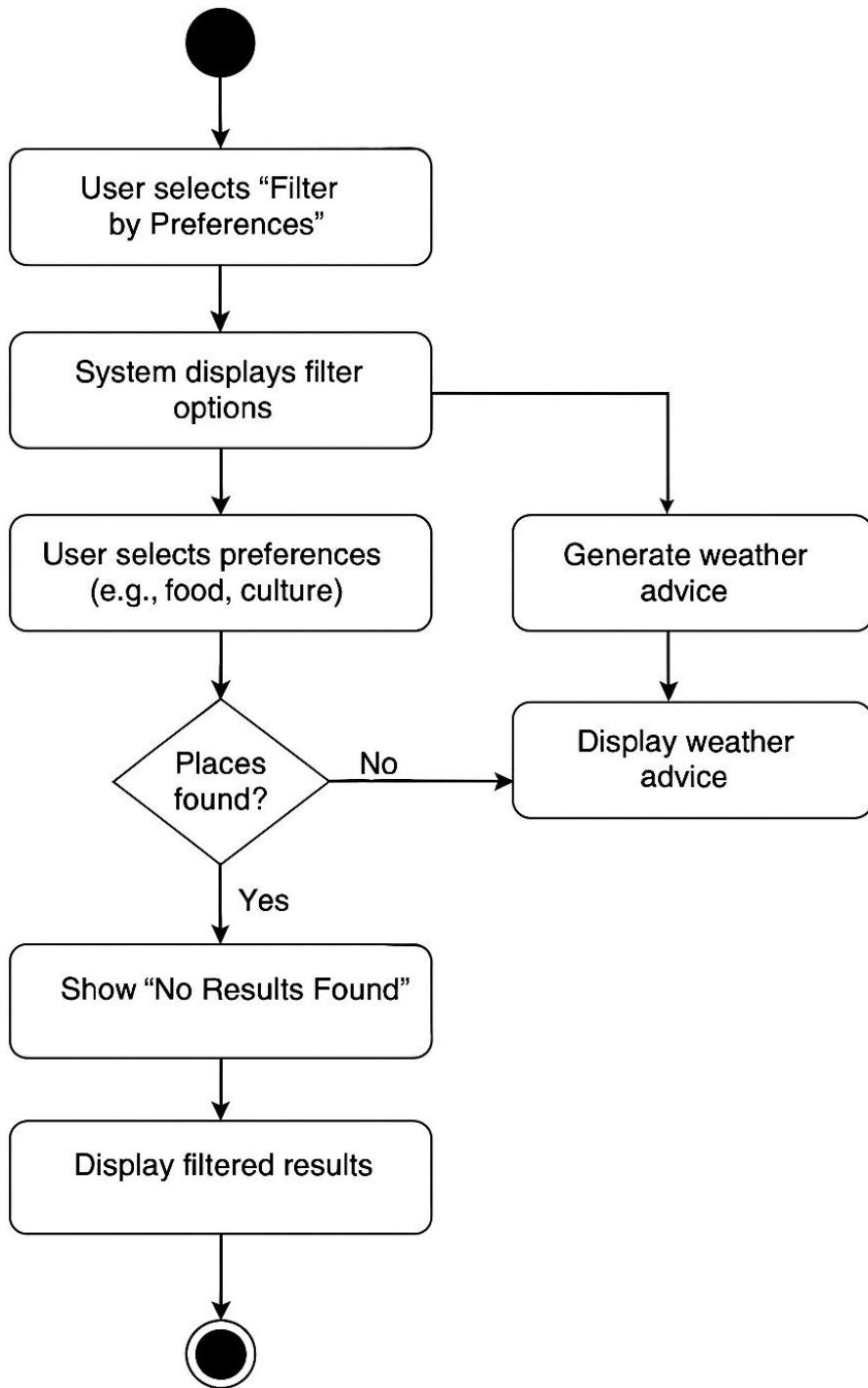
### **UC-1: Access Website**

This activity diagram illustrates the process of accessing the website, entering credentials, and handling login success or failure.



### **UC-5: Filter by Preferences**

This diagram shows how users filter preferences and how the system fetches and displays related tourist places.



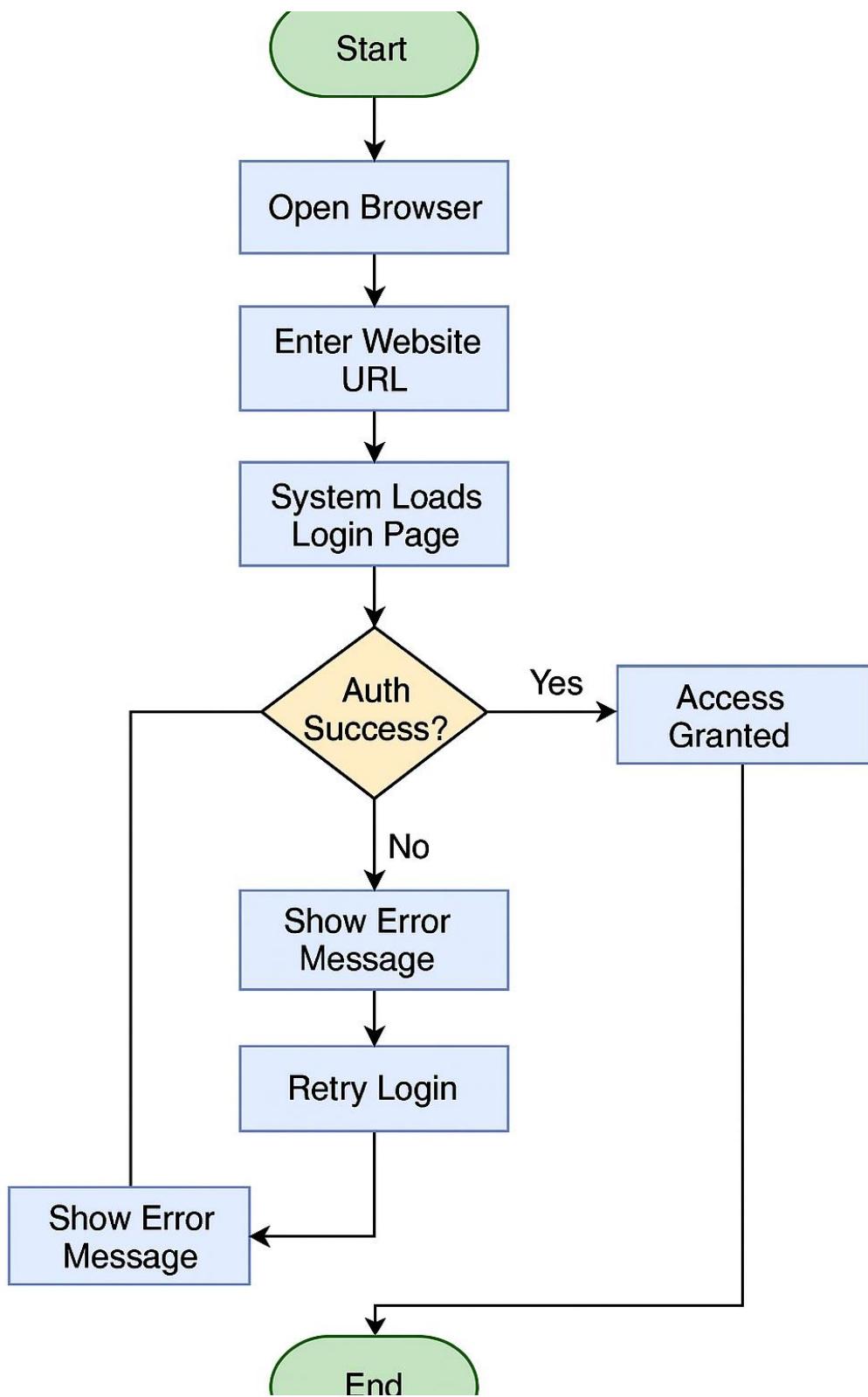
### **UC-11: Weather Advice**

Text-based diagram: The user requests weather advice, the system fetches weather data and displays tailored suggestions.

## Sequence Diagrams

### UC-1: Access Website

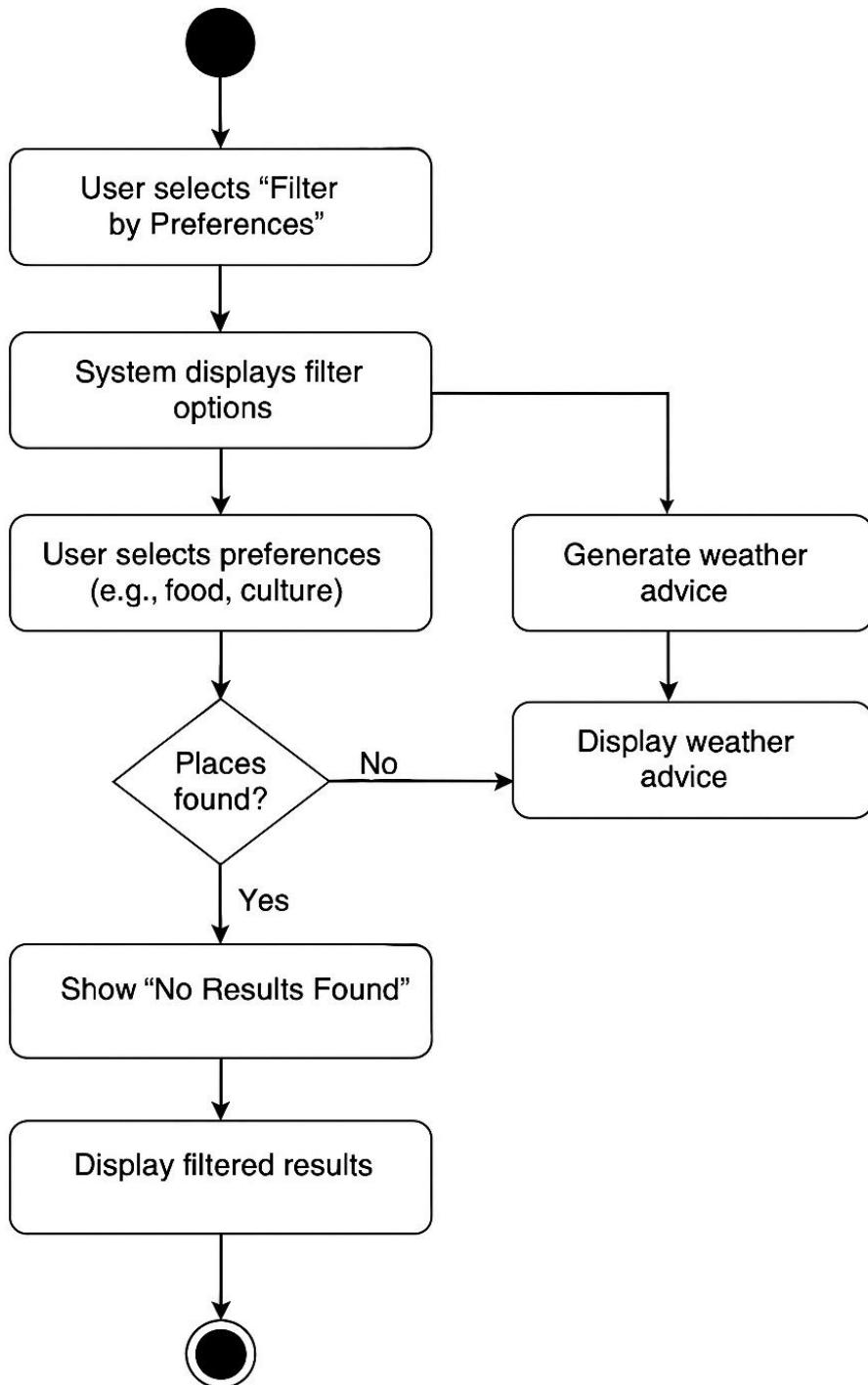
```
User          →          Browser:          Open          URL
Browser       →          Server:          Request        Homepage
Server        →          Auth            System:        Load           Login        Page
User          →          Login           Page:         Enter        username/password
Login         Page        →          Auth            System:        Authenticate    user
Auth          System      →          Server:        Return        auth         result
alt
    Server     →          Browser:        Display        homepage
else
    Server     →          Browser:        Show          error         message
end
```



### UC-5: Filter by Preferences

User	→	App	UI:	Select	preferences	(e.g.,	culture,	food)
App		UI	→	Server:	Send	filter		request
Server			→	DB:	Query	matching		places
DB			→	Server:	Return	filtered		places
Server			→	App	UI:	Display		list

App UI → Map Service: Show location of selected place



### UC-11: Weather Advice

User → App UI: Request weather advice

App UI → Weather API: Fetch current weather

Weather API → App UI: Return weather data

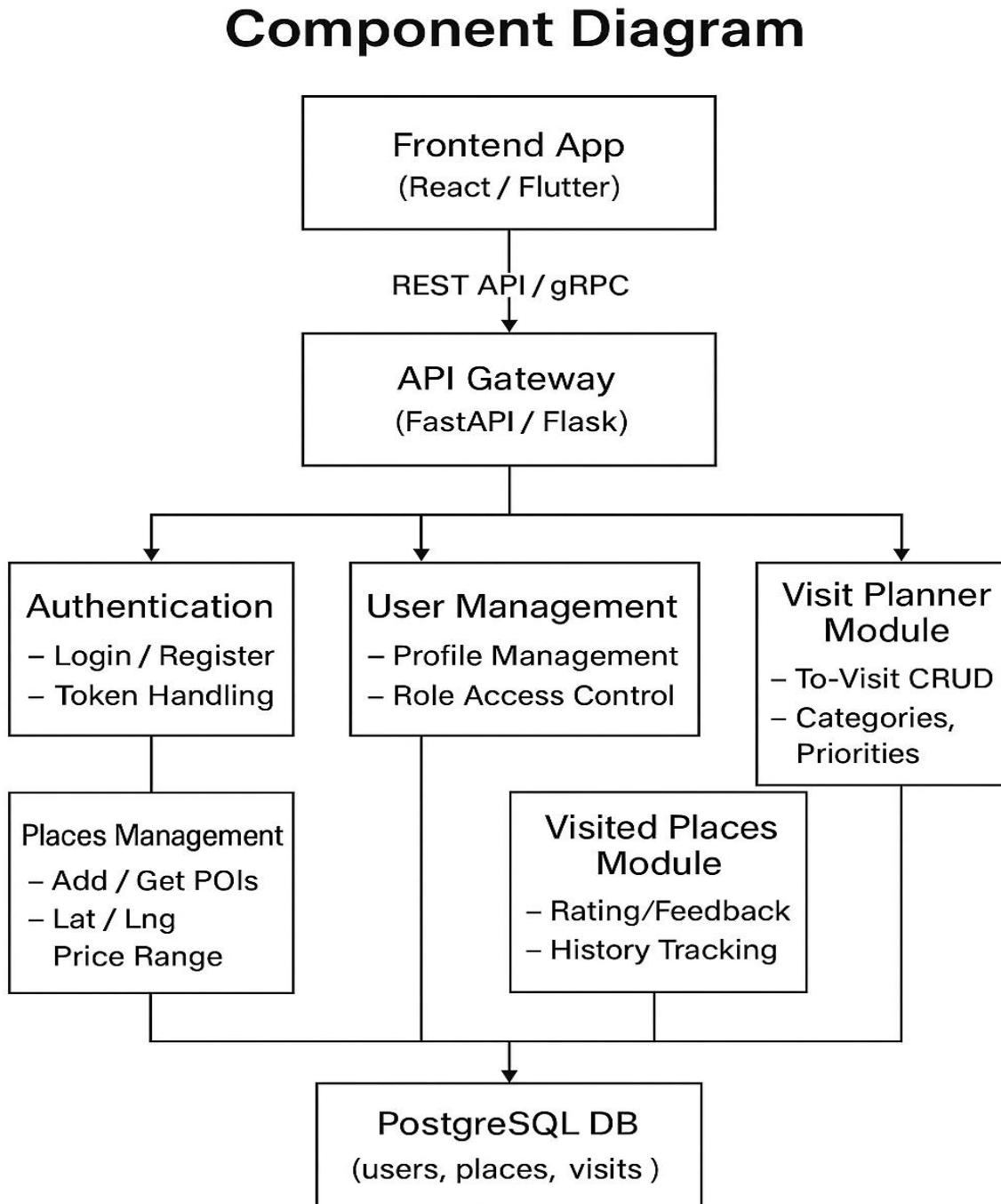
App UI → Advice Engine: Generate advice

Advice Engine → App UI: Return tailored advice

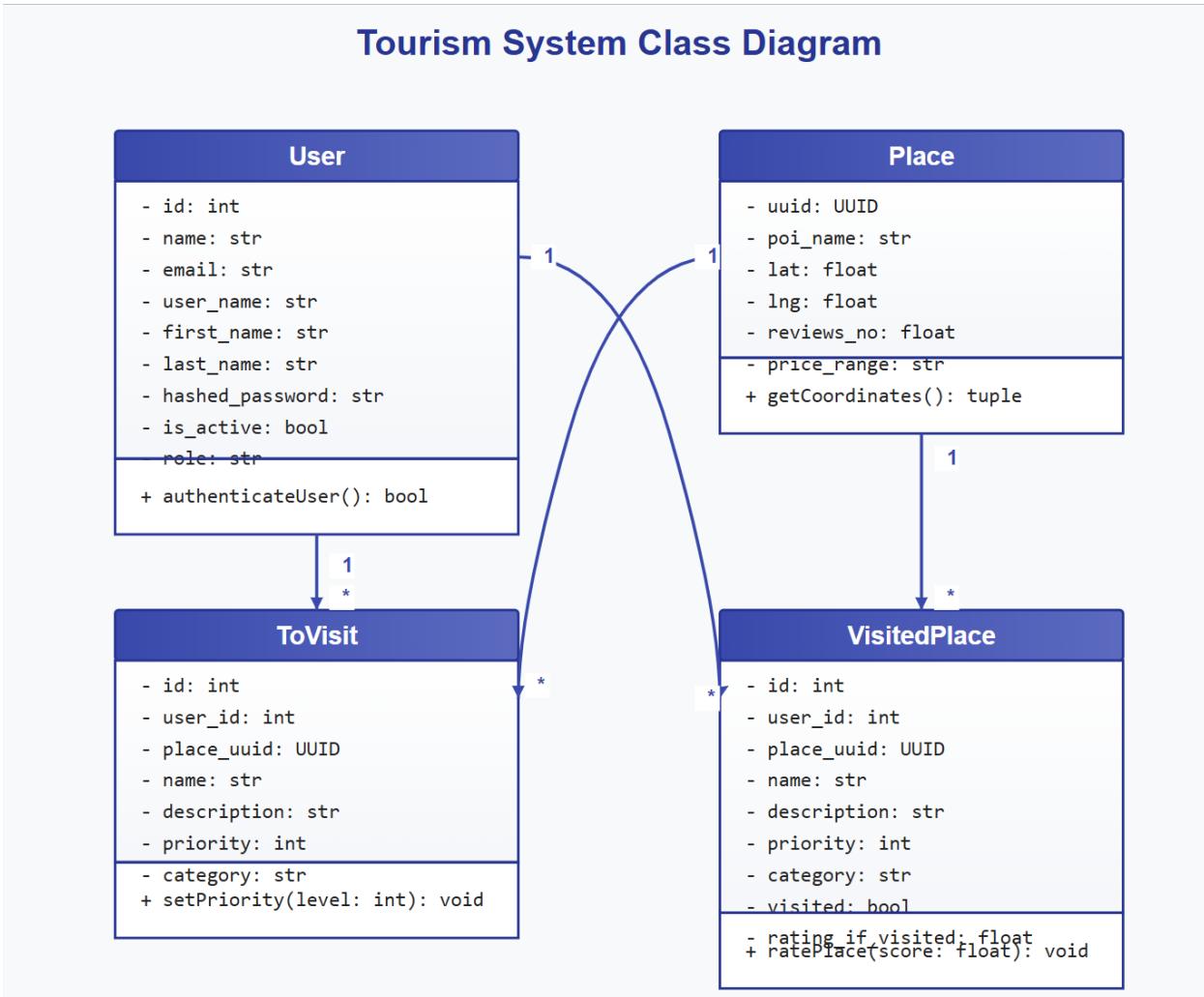
App UI → User: Display weather & advice

System Design

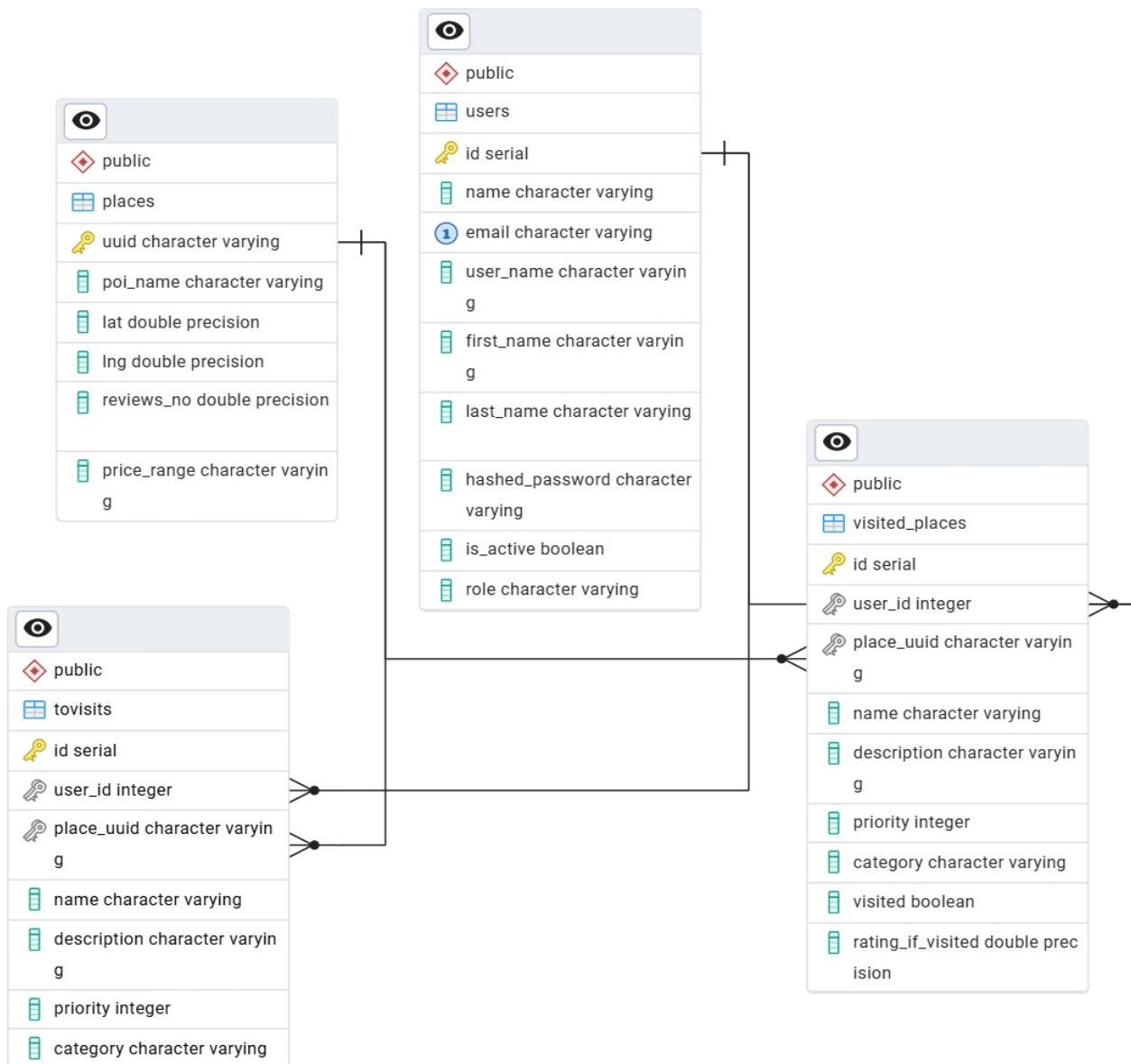
Component Diagram



## Class Diagram



## Entity Relationship Diagram (ERD)



## Test cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Status
TC01	User Registration	- Navigate to registration page - Fill out all required fields - Submit form	User is created and redirected to the login page	Pass
TC02	User Login	- Enter correct credentials - Click login	Dashboard loads successfully	Pass
TC03	Add Place to Visit List	- Login - Search for a place - Click "Add to Visit List"	Place is saved in the tovisits table	Pass
TC04	View Visit List	- Login - Navigate to "Visit List"	User sees all saved places with priority & category	Pass
TC05	Mark Place as Visited	- Open visit list - Click "Mark as Visited"	Record is moved to visited_places with visited=true	Pass
TC06	Rate a Visited Place	- After marking as visited - Provide a rating	Rating is saved in rating_if_visited	Pass
TC07	Add New Place (Admin)	- Login as admin - Go to "Add Place" - Fill POI details - Submit	Place appears in the global list of available places	Pass
TC08	Display POIs by Category	- Select category filter - View list	Only POIs in the selected category are displayed	Pass
TC09	User Cannot Add Duplicate Place to Visits	- Add a place - Try adding same place again	App shows warning: "Already in your list"	Pass
TC10	Remove Place from Visit List	- Select a place - Click "Remove"	Place is removed from tovisits table	Pass
TC11	Secure Password Storage	- Register a new user - Inspect DB	Password is hashed in the database	Pass
TC12	Role-Based Access	- Try admin-only page as regular user	Access is denied or redirected	Pass
TC13	Data Validation During Input	- Leave fields blank or enter invalid formats	Form displays relevant validation errors	Pass

## RAG System Component

