TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

TUNISIAN TELEVISION: TALVZA EL WATANIYA

- WEB SERVICE

PROFESSOR: MONTASSAR BEN MESSAOUD

- MINI-PROJECT BY

RANIM GSOURI
BA-IT SENIOR STUDENT

Telephone
29965546

Address
av 20 Mars Mourouj

January 2025

TABLE OF CONTENT

OUTLINE

01 Problems

02 Solutions

03 System Design

04 Implementation

05 Visualization

06 Conclusion

PROBLEMS

IDENTIFICATION

01 Time-zone differences for the tunisian Diaspora living abroad

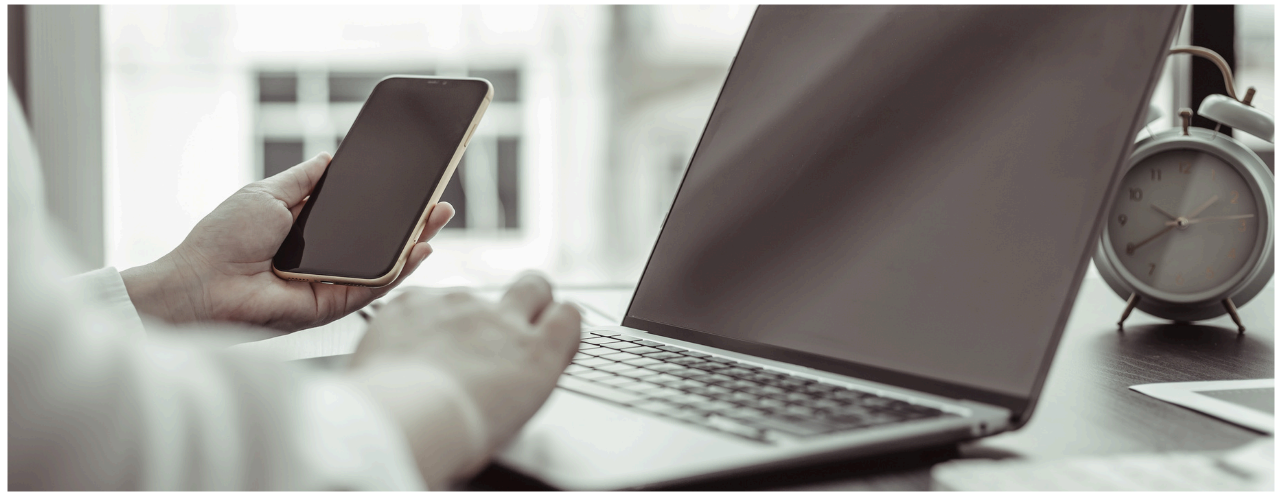
Despite their strong connection to Tunisian culture and media, many expatriates experience difficulties due to geographical and time-zone differences, as well as the lack of centralized platforms providing localized schedules. This gap often leaves them feeling disconnected from their cultural roots, particularly when it comes to missing their favorite TV shows.

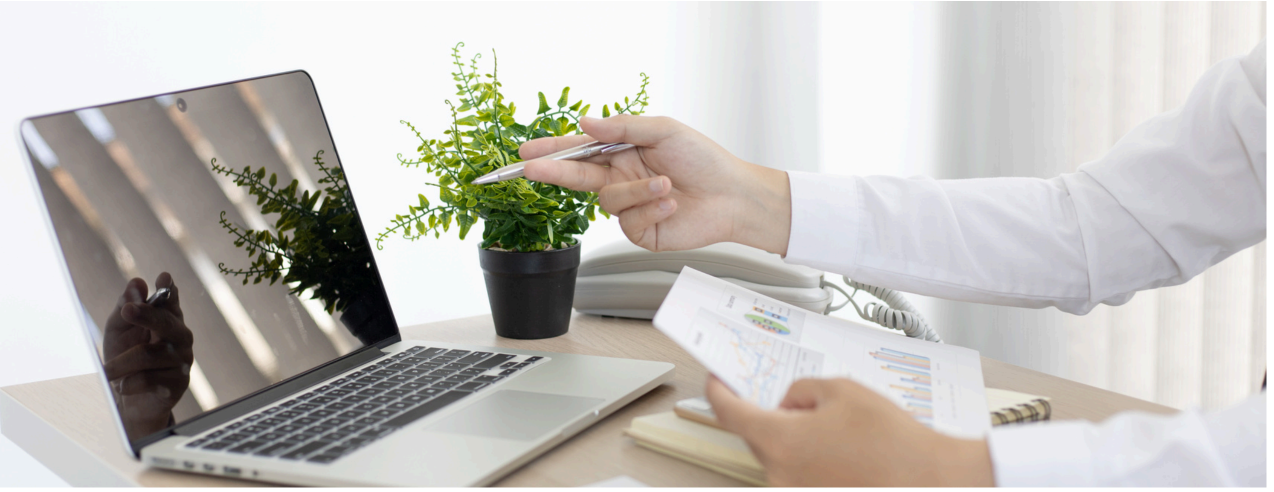
02 Challenges in accessing real-time information about Tunisian television programming

The Tunisian diaspora, comprising individuals living abroad for work, study, or other reasons, faces challenges in accessing real-time information about Tunisian television programming

SOLUTIONS

PRESENTATION





Solution

Talvza Al Wataniya aims to bridge this gap by offering a digital platform tailored for Tunisians abroad. This initiative provides real-time access to TV schedules, show timings, and related content, ensuring that users can stay connected with their homeland’s media. By integrating user-friendly functionalities and leveraging advanced technologies, the platform addresses the diaspora's need for accessibility and engagement.

Solution Overview

The Talvza Al Wataniya platform is designed to:

- Deliver real-time schedules for Tunisian TV channels.
- Adjust show timings based on the user’s geographic location and time zone.
- Provide an intuitive interface for searching and viewing show timings.

This project leverages Flask to ensure seamless communication between the user interface and the backend database. Additionally, the implementation of CRUD operations and database management tools like MySQL guarantees reliability and user-centric functionality. A secure password hashing mechanism is also in place to protect user credentials.

SYSTEM DESIGN:

Admin Dashboard

An interface for managing user accounts, schedules, and platform monitoring.

User interface

A web-based frontend that allows users to interact with the platform, search for TV schedules, and view show details.

Backend

Built with Flask, it facilitates API requests and manages communication between the frontend and the database.

Database:

A MySQL database that stores schedule information, user details, and show metadata.

The sequence diagram outlines the secure user authentication and authorization process. It begins with the user entering their email and password into the frontend interface. As the user types, the frontend ensures the password is displayed as dotted points for privacy. The frontend also validates the email format and communicates with the authentication server to evaluate the password’s strength, providing feedback to the user (e.g., ”Invalid email” or ”Weak password”).

Once the user submits the form, the frontend forwards the credentials to the backend server. Before storing the password in the database, the backend applies a hashing algorithm to secure it, ensuring that even if the database is compromised, the raw password remains protected. The hashed password is then stored in the database, and the backend sends a confirmation message back to the user, indicating that the sign-up process was successful.

When the user logs in, the submitted credentials are sent to the authentication server, which verifies the identity by comparing the provided password (after hashing) with the stored hash in the database. If the credentials match, the server authorizes the user and grants access to the requested resources based on their permissions. This ensures a secure and seamless process for both signing up

and logging in, with robust protections against unauthorized access.

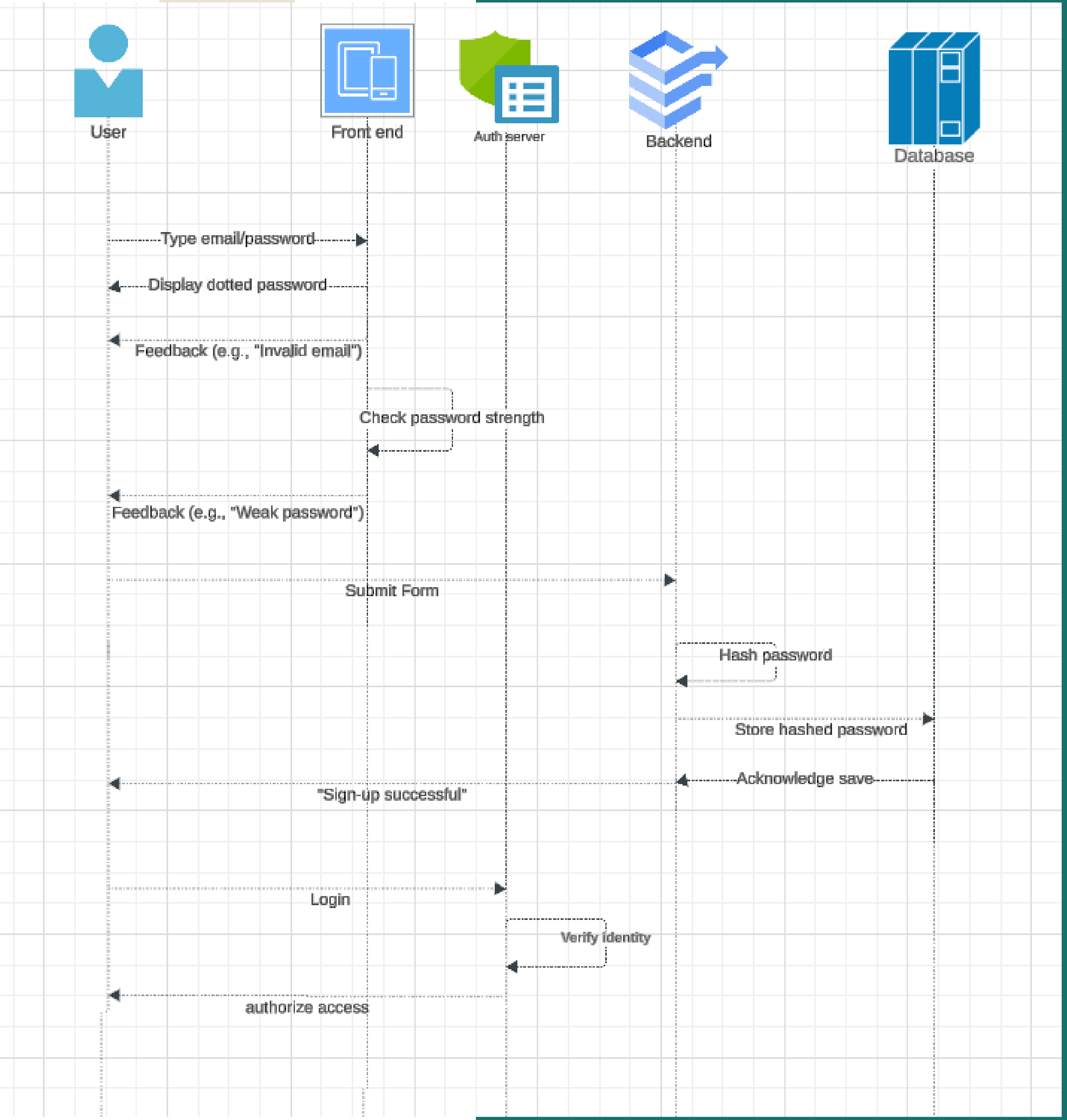
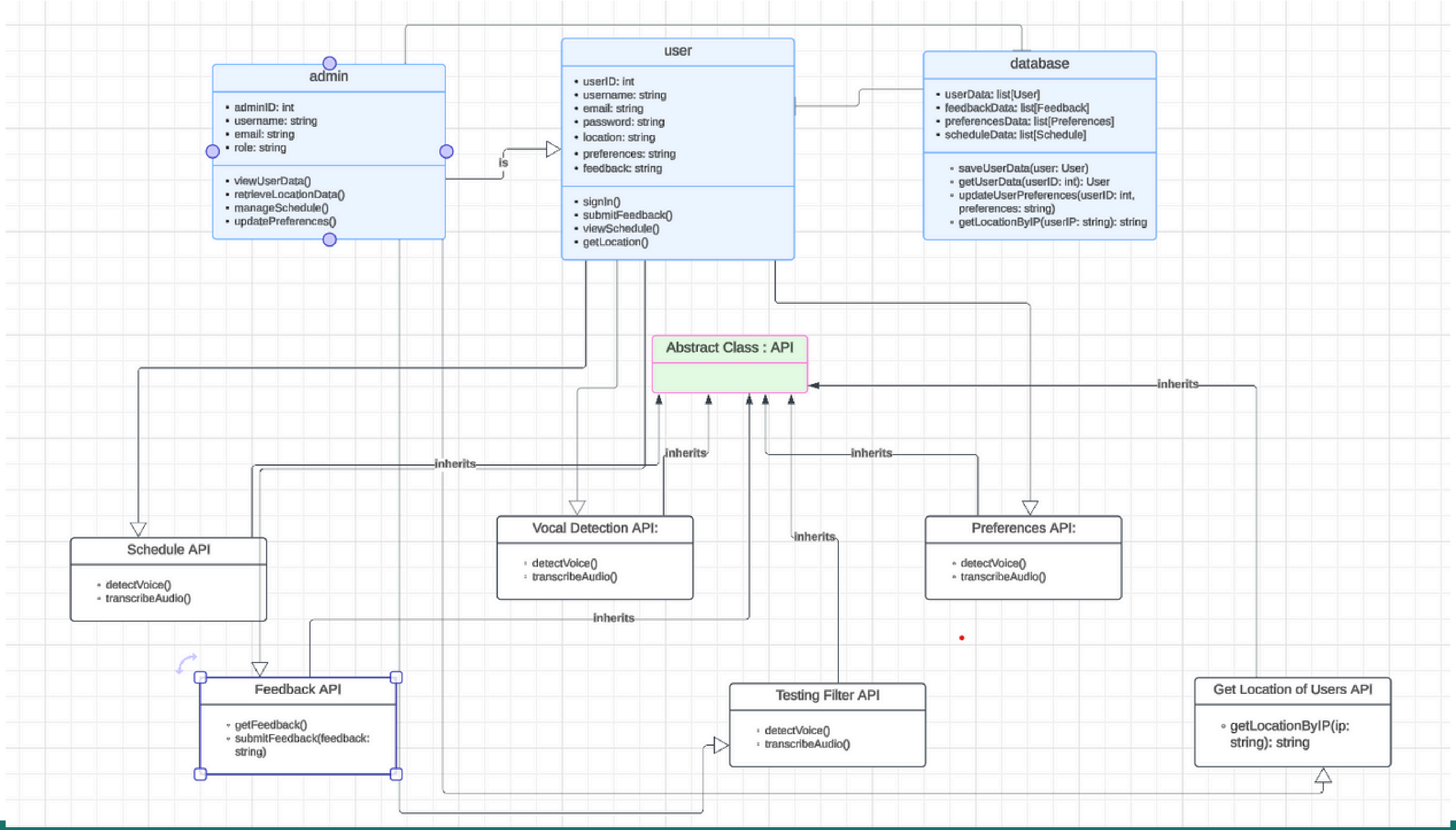


Figure: Sequence Diagram for User Login and Password Hashing.



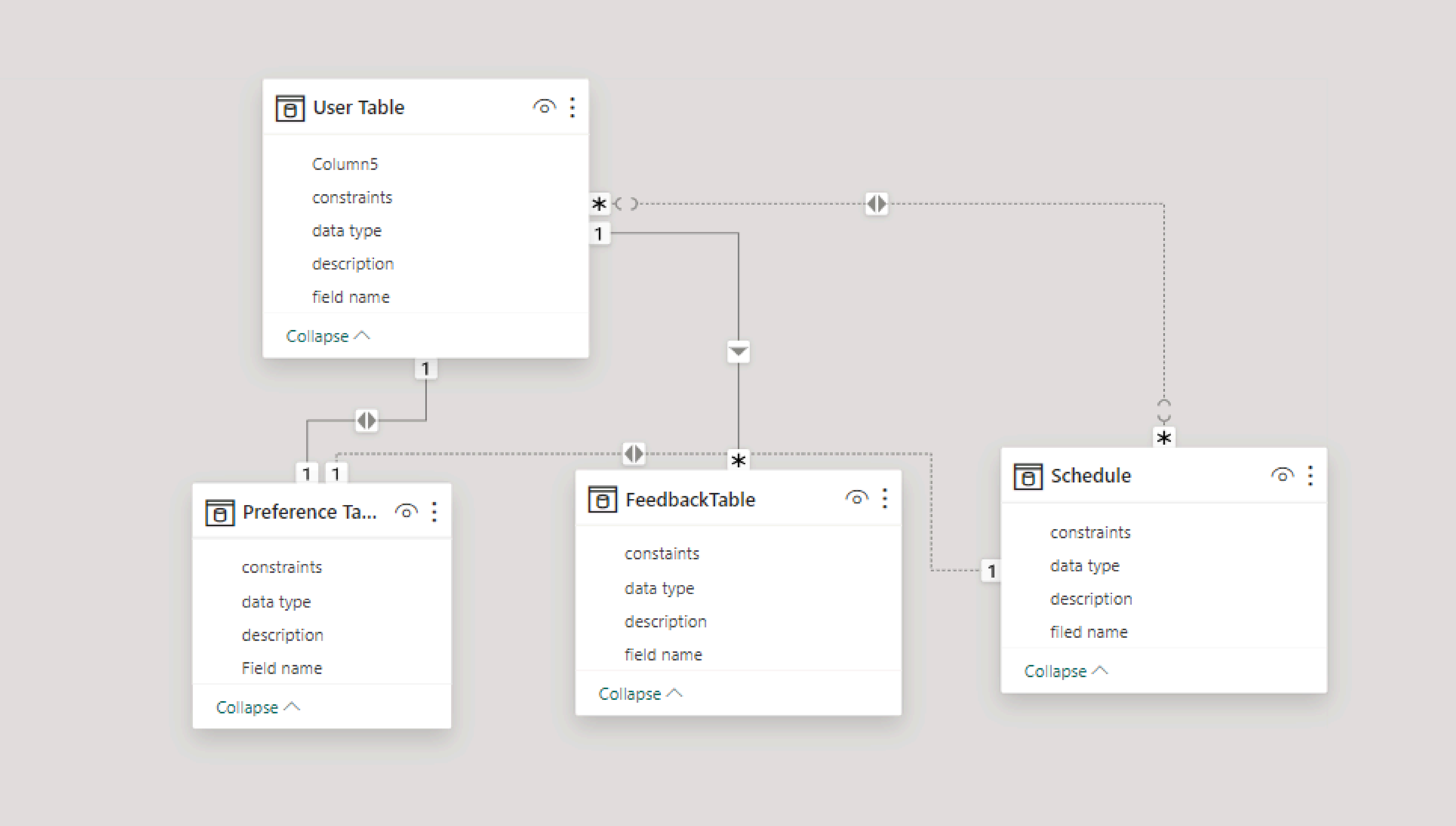
UML CLASS DIAGRAM FOR THE BACKEND ARCHITECTURE.

The UML class diagram represents the fundamental structure of the backend architecture of the platform. It showcases the relationships between various components of the system, primarily focusing on the interactions between the user, the database, and the APIs. Each class in the diagram corresponds to a key component or service within the system, encapsulating attributes and methods essential to its functionality.

The diagram includes classes for user management, including user authentication and location tracking, reflecting the process of signing in, storing user information, and retrieving the location based on the user's IP address.

The database interaction layer is depicted as separate classes, managing data flow and ensuring secure storage of user credentials and preferences. The APIs, such as the Schedule API and Feedback API, are also illustrated, highlighting their role in fetching data from external services or databases and responding to client-side requests.

MySQL Database Schema showing the relationships between tables.



IMPLEMENTATION

API Endpoints

Feedback APi

Enables users to provide feedback, which can be retrieved, updated, or deleted. This API offers CRUD (Create, Read, Update, Delete) operations, making it essential for gathering and managing user feedback.

Preference APi

Allows users to customize their preferences, such as setting notifications for TV schedules or other preferences related to the system. These preferences can be both retrieved and updated using the respective endpoints.r

Schedule APi

Provides access to TV schedules based on channels or specific shows. Users can either view a complete schedule for a specific channel or search for a particular show within the schedule.

API Endpoints

Vocal Detection API

This endpoint leverages AI-powered algorithms to detect vocal content, trans-ferring it into the search box, to facilitate the search operation for the user and ensuring that interactions comply with platform guidelines.

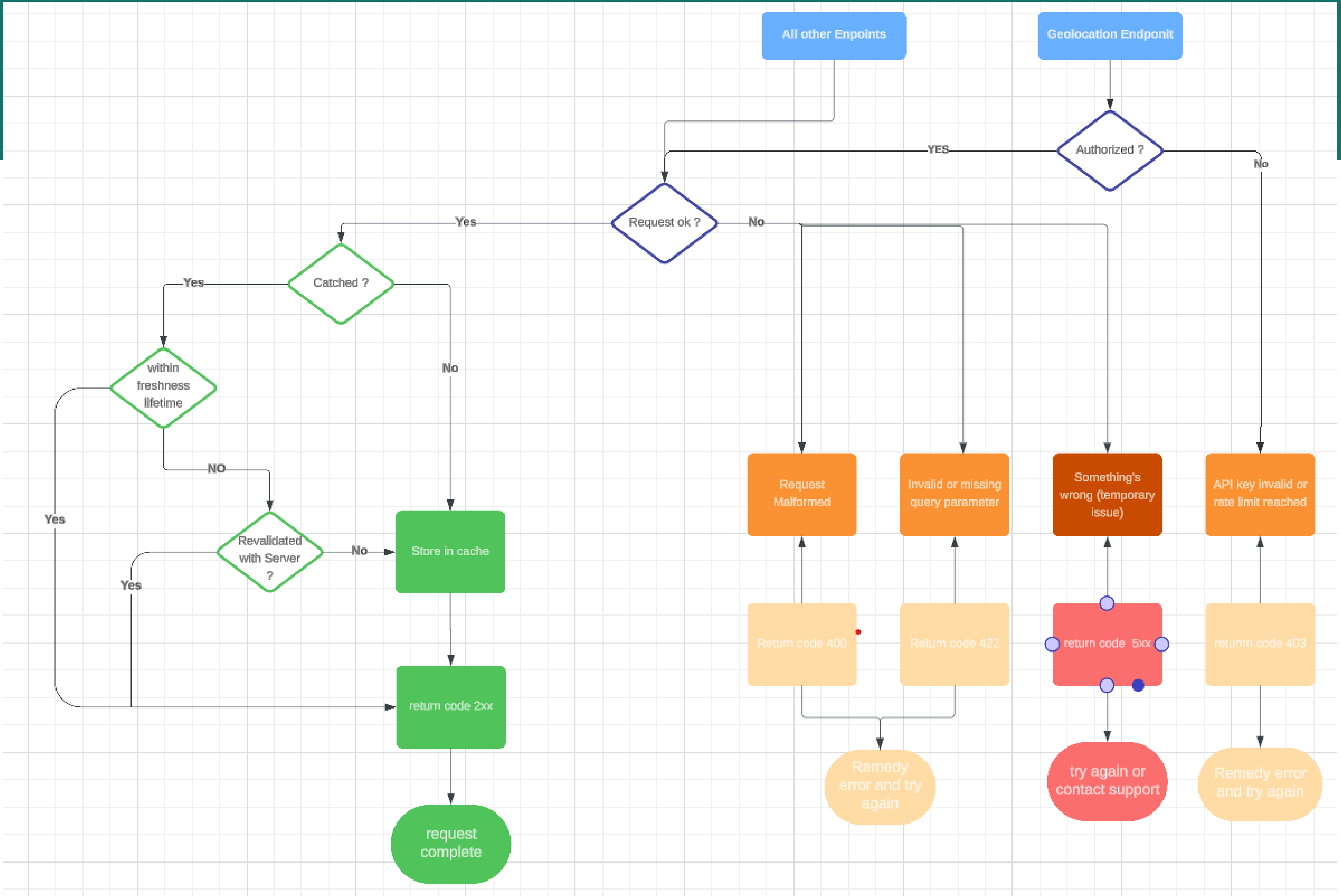
Geolocation API

This external API is integrated into the system using an API key. It retrieves the user's IP address and geolocation data, offering contextualized services based on their location and allowing the admin to know from which countries tunisian people are using his app.

Filter APi

Filterates bad words.

API Workflow Diagram illustrating user interactions with endpoints.


All other Enpoints
Geolocation Endponit
Authorized ?
YES
No
Request ok ?
Yes
No
Catched ?
Yes
No
within freshness lifetime
NO
Yes
Revalidated with Server ?
No
Yes
Store in cache
return code 2xx
request complete
Request Malformed
Invalid or missing query parameter
Something's wrong (temporary issue)
API key invalid or rate limit reached
Return code 400
Return code 422
return code 5xx
returnn code 403
Remedy error and try again
try again or contact support
Remedy error and try again

When a request is sent to an API endpoint, it is validated and processed.

Cached requests within their freshness lifetime are served immediately; otherwise, they are revalidated with the server. Authorized endpoints, like the Geolocation API, check API keys, rejecting invalid ones with a 403. Malformed or incomplete requests return errors (400 or 422), while temporary issues result in a 5xx response. Valid requests are processed, stored in cache if needed, and completed with a 2xx status.

GEOLOCATION MAP

showing user distribution based on IP addresses.

Geolocation Visualization

The external geolocation API plays a vital role in translating IP addresses into geographic data, allowing for accurate representation of these user locations.

For example, the API identifies users from Tunis (Tunisia), Paris (France), Berlin (Germany), and New York (USA), showcasing the platform’s broad appeal.

This geolocation data helps improve decision-making, enabling targeted content delivery, regional analysis, and user engagement strategies.

LIMITATIONS AND FUTURE ENHANCEMENTS

Future Enhancement:

The project envisions the following improvements:

- Multi-language Support: Expanding the platform to include multiple languages for broader accessibility.
- Mobile App Integration: Developing a mobile version for better usability.
- Enhanced Analytics: Providing personalized recommendations based on user behavior.
- OAuth Implementation: Strengthening security with advanced authentication mechanisms.

Limitations

Despite its advantages, the platform has some limitations:

- Dependency on External APIs: Potential issues if third-party services experience downtime.
- Geolocation Accuracy: Challenges in precise location tracking, especially for users on VPNs or mobile networks.
- Scalability Constraints: Initial design may require upgrades to handle high traffic volumes.

CONCLUSION

Concluding remarks:

By addressing key challenges and leveraging modern technologies, This platform fosters a stronger connection between tunisians and their cultural heritage. Its APIs are thoughtfully designed for fetching live TV schedules, providing location-based show timings, and managing user interactions like search queries. These APIs are not only effective for the platform but also reusable across other applications.

Conclusion :

Talvza Al Wataniya bridges the gap between Tunisians and their cultural roots by offering an accessible, user-friendly platform that adapts to modern needs. Through innovative design and scalable technology, it ensures Tunisians abroad can remain engaged with their country's media content. As the platform evolves, it promises to solidify its role as a vital link connecting Tunisians to their heritage, regardless of geographic boundaries.

THANK YOU

FOR YOUR NICE ATTENTION

Jaunuary 2025