VR Tour - Man and the Living World Museum

Detailed Design

Performed By:

Bar Yaron – 206481533

Yakir Zafrani – 313327496

Project Director – Dr. Sharon Yalov Handzel

Table of Contents

Scope 3

Use Cases 3

User navigates through the VR museum environment: 3

User interacts with exhibits: 3

User accesses exhibit information: 3

Admin manages exhibit database: 3

Detailed Design 4

System Architecture 4

Performance 4

Data 4

Integration 4

User Experience 4

Scalability 5

Security 5

Technological Requirements 6

Programming Languages and tools 6

VR Hardware Platforms 6

3D Modeling and Animation Tools: 6

Development Tools 6

Documentation Standards 6

Block Diagram 7

DB Scheme 7

Mockup 8

DB side 8

Unity side 9

Scope

This document describes the system architecture, application use cases, and technological requirements of the VR tour project for our client - Man and the Living World Museum.

Use Cases

User navigates through the VR museum environment:

- User puts on the VR headset and enters the virtual museum environment.
- Within the VR environment, the user can look around and move using VR controllers or keyboard/mouse inputs.
- The user can explore different sections of the museum, such as galleries, hallways, and exhibition rooms, by walking or teleporting within the virtual space.

User interacts with exhibits:

- While exploring the museum, the user encounters interactive exhibits represented by 3D models or artifacts.
- The user can approach an exhibit and use VR interactions to select, examine, or interact with objects within the exhibit.
- For example, the user can pick up a virtual artifact, rotate it, zoom in to examine details, or trigger animations or audiovisual presentations associated with the exhibit.

User accesses exhibit information:

- Upon interacting with an exhibit, the user can access detailed information about it.
- Information panels or overlays within the VR environment provide descriptions, historical context, multimedia content (e.g., images, videos), and related articles or resources associated with the exhibit.
- Users can engage with this information to enhance their understanding and appreciation of the exhibit's significance and relevance.

Admin manages exhibit database:

- Admin users have access to backend tools or interfaces for managing the exhibit database.
- Admins can add new exhibits to the database, providing metadata such as exhibit name, description, location, and associated multimedia content.
- They can edit existing exhibit information, update multimedia assets, or remove outdated exhibits from the database as needed.

Detailed Design

System Architecture

The system architecture of the VR museum application may include the following components:

- Client-side VR application developed using Unity for immersive user experience.
- MySQL database for storing and managing exhibit data, metadata, and user interactions.
- Communication protocols (e.g., HTTP, WebSockets) for interaction between client and server components.

Performance

- VR application should maintain a consistent frame rate (e.g., 60 - 90 FPS) for smooth user experience.
- Database queries should execute within milliseconds to ensure quick retrieval of exhibit information.
- Loading times for exhibits and database interactions should be minimal to avoid user frustration.
- Server-side components should be scalable to handle increasing user traffic and database load.

Data

- The system shall utilize a distributed database architecture, ensuring data consistency between the museum managers contents and the 3d VR environment.
- MySQL database schema should include tables for exhibits, exhibit metadata, user interactions, and user profiles.
- Exhibit data should include attributes such as exhibit name, description, location, and associated multimedia content.
- Database should support relational queries and indexing for efficient data retrieval and management.

Integration

- The unity engine shall integrate with the MySQL database to import 2D exhibits from the database into a designated area in the 3D VR environment using industry-standard APIs.

User Experience

- Realistic and immersive 3D environments that closely resemble the physical museum.
- Accessible features for users with disabilities, including alternative navigation options and audio descriptions.
- Onboarding tutorials or guided tours to help users familiarize themselves with VR controls and features.

Scalability

- Application architecture will be designed for horizontal scalability to accommodate increasing user traffic and database load.
- VR museum application should be implemented as dynamically as possible in unity, to allow project growth as the museum and its contents evolve.

Security

- User Identification for database access by logging in with a username and password, in basic token, listed users will be managed in a table in the database.
- Security level to restrict access to the database from different users, users could only perform SELECT, UPDATE, and INSERT operations, will be managed in a table in the database.

Technological Requirements

Programming Languages and tools

- C# (C - Sharp) – for programming in the unity engine and implementing the VR environment along with all of its features.
- XR Plugin – A built in library for unity VR development.
- MySQL - for database querying and data manipulation.
- JavaScript – for creating a UI application to interface the DB.
- HTML - for web page development.
- HTTP/HTTPS for API endpoints for client-server communication.

VR Hardware Platforms

- Oculus Rift, Oculus Quest (compatible with quest 1, quest 2, and quest 3).
- Compatible VR controllers for user interaction.

3D Modeling and Animation Tools:

- Blender, for creating 3D models or using unity asset store / other sources for 3D models and assets.

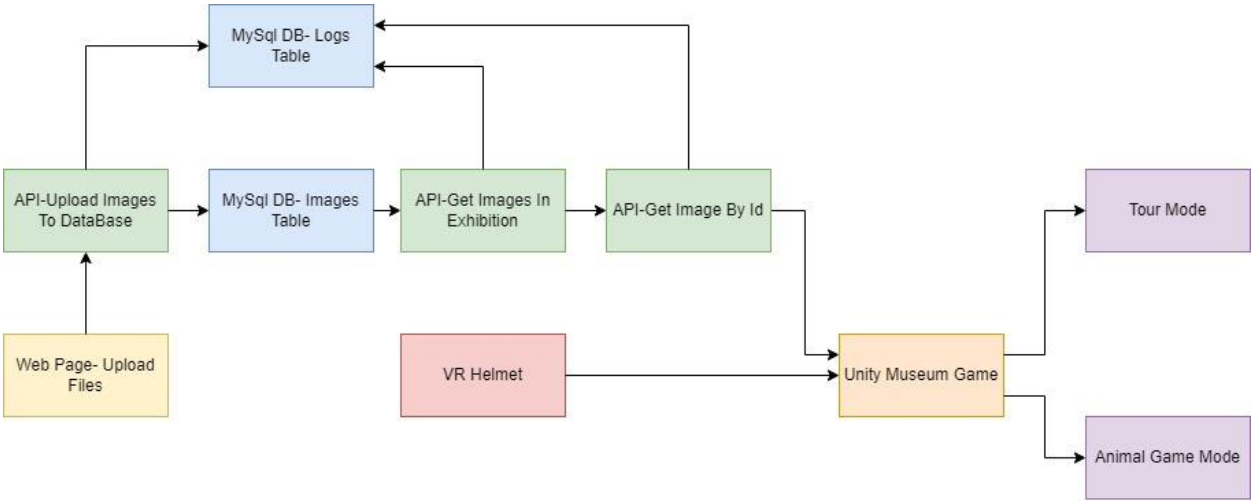
Development Tools

- Git for version control and collaborative development.
- GitHub for hosting and managing code repositories in unity.
- Unity version 2021.3 LTS – for VR application development.
- Microsoft Visual Studio 2022 – IDE for C# scripts that will run in Unity.
- Postman – for web API.
- Java Spring Boot - for DB server infrastructure.
- JetBrains's intellij IDEA – for implementing Java Spring Boot.

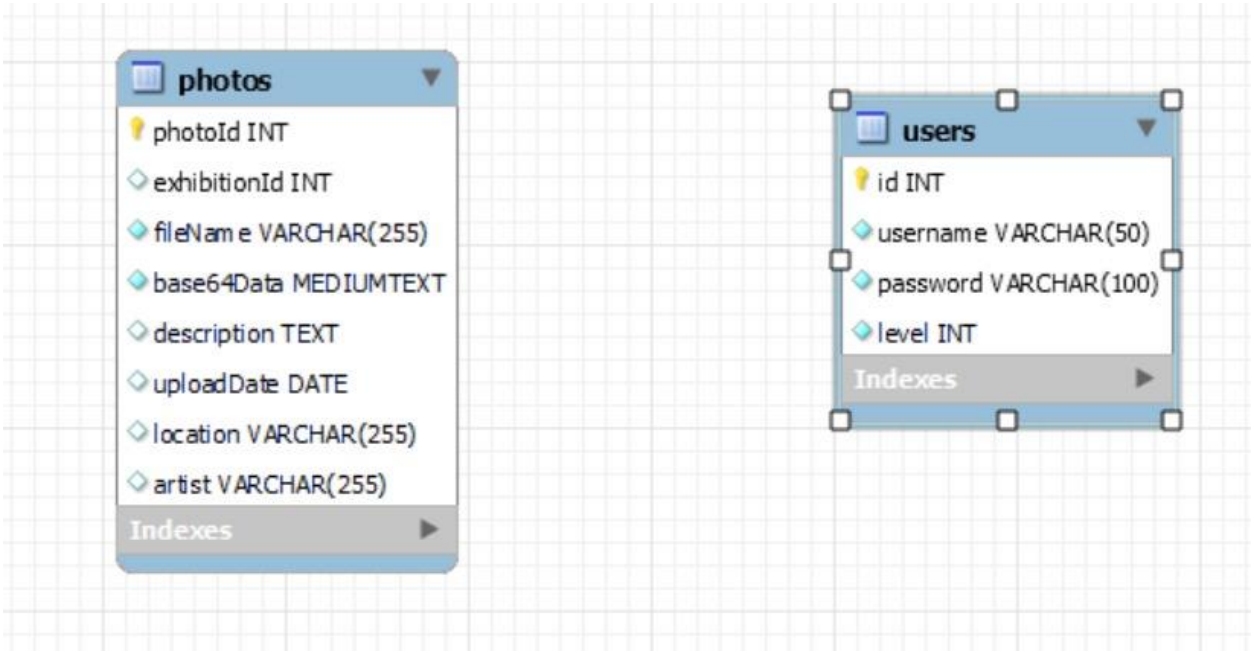
Documentation Standards

- Unity design patterns for clean code and scalable project infrastructure.
- Clear and scalable DB architecture.

Block Diagram

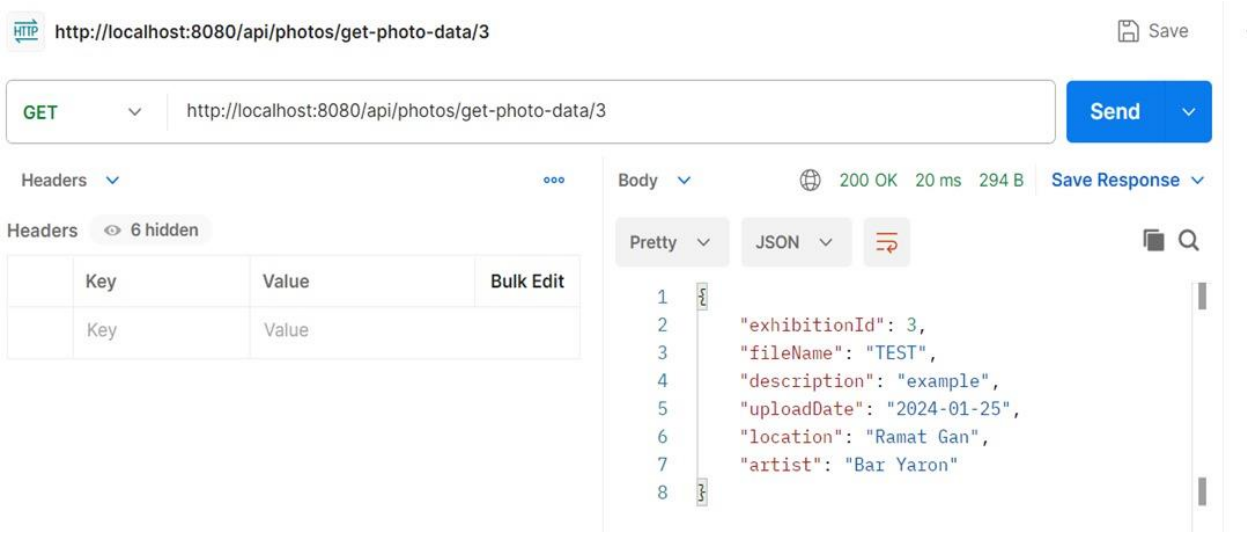


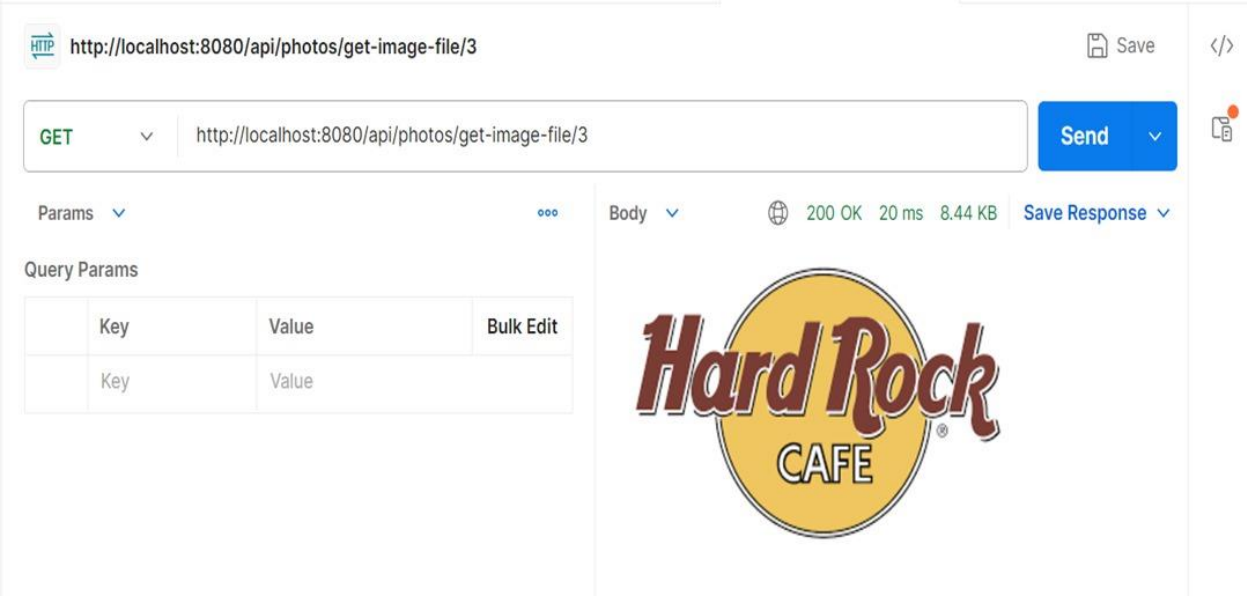
DB Scheme

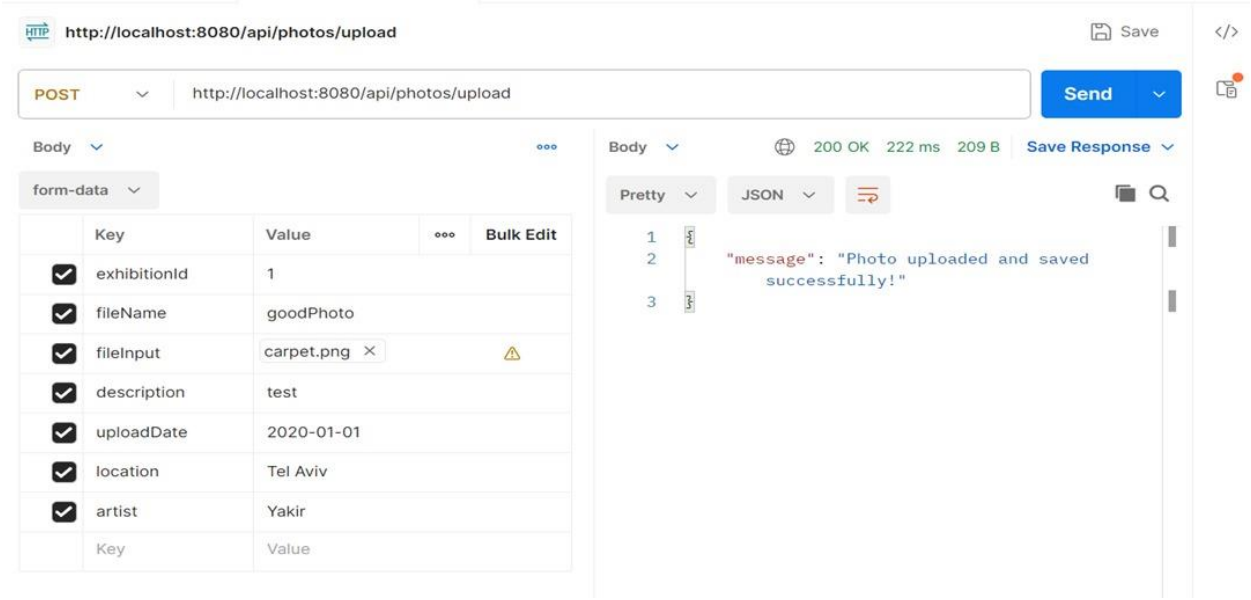


Mockup

DB side







Unity side

Will be added as development continues.