Photo: Deschamps Thibaut

Vivvoy

Nascimento Neia
C#24
Holberton School
Thonon-les-Bains 74200, FR.

Portfolio Project - API and Technical Documentation (Stage 3)

Research & Project Approval

Project Name: Vivvoy
Tagline: Provide curated and high-quality information about French tourism, cultural heritage, and local experiences.
Prepared by: Néia Nascimento
Duration: December 09 - March 03, 2025

Haute-Savoie, nestled in the French Alps, offers a rich tapestry of experiences that blend artisanal craftsmanship, outdoor adventures, historical exploration, and sustainable development. Vivvoy aims to highlight these unique opportunities, showcasing the region's charm and supporting sustainable tourism practices.

1. Introduction

This document provides an overview of the external and internal API endpoints for the Vivvoy platform. It includes API methods, request formats, and expected responses. Additionally, it incorporates Stage 3 deliverables, including system architecture, database design, sequence diagrams, and SCM/QA strategies.

2. Objectives of Stage 3

Stage 3 aims to translate project objectives and requirements into a detailed technical plan by defining architecture, components, database structures, authentication, sequence diagrams, and API endpoints. This stage ensures that development follows a structured approach with minimal risks.

Importance of Stage 3

By planning technical aspects, source control, and quality assurance in advance, the project minimizes risks, improves clarity, and enhances development efficiency. The documentation aligns all stakeholders on the project’s technical direction.

3. Tasks for Stage 3

Define User Stories and Mockups

Prioritize User Stories

Must Have:

1. *As a tourist, I want to browse available artisan experiences, so that I can find activities that interest me.*
2. *As a tourist, I want to book an artisan experience, so that I can participate in local cultural activities.*
3. *As a tourist, I want to receive a confirmation notification after booking, so that I have proof of my reservation.*
4. *As an artisan, I want to create and manage my profile, so that I can showcase my work and available experiences.*
5. *As an artisan, I want to manage my bookings, so that I can accept or decline reservations.*
6. *As a platform admin, I want to verify new artisan profiles before they go live, so that only authentic artisans can offer services.*
7. *As a user, I want to log in securely with an email and password, so that my data is protected.*

Should Have: 8. *As a tourist, I want to filter artisan experiences by category (e.g., pottery, cooking), so that I can easily find relevant activities.* 9. *As a tourist, I want to leave reviews for an artisan, so that I can share my experience with others.* 10. *As an artisan, I want to respond to reviews, so that I can engage with past visitors.*

Mockups

Mockups have been created for core pages including:

- **Booking Page:** Displays available artisan experiences and booking form.
- **Artisan Profile Page:** Shows artisan details, past work, reviews, and booking options.
- **Homepage:** Provides navigation to key platform features.

These wireframes were designed using **Figma** and are available for review.

Design System Architecture

The **Vivvoy platform** follows a modular architecture designed for **scalability and efficiency**. Below are the key components:

Frontend: HTML, CSS, and JavaScript(potentially)

- Displays artisan listings and booking forms.

- Handles client-side interactions.

Backend: Flask REST API

- Processes user authentication (JWT-based).
- Handles CRUD operations for users, artisans, and bookings.
- Manages authorization and role-based access control.

Database: PostgreSQL

- Stores user profiles, artisan details, and booking records.
- Ensures data integrity with relational constraints.
- Implements indexing strategies for performance optimization.

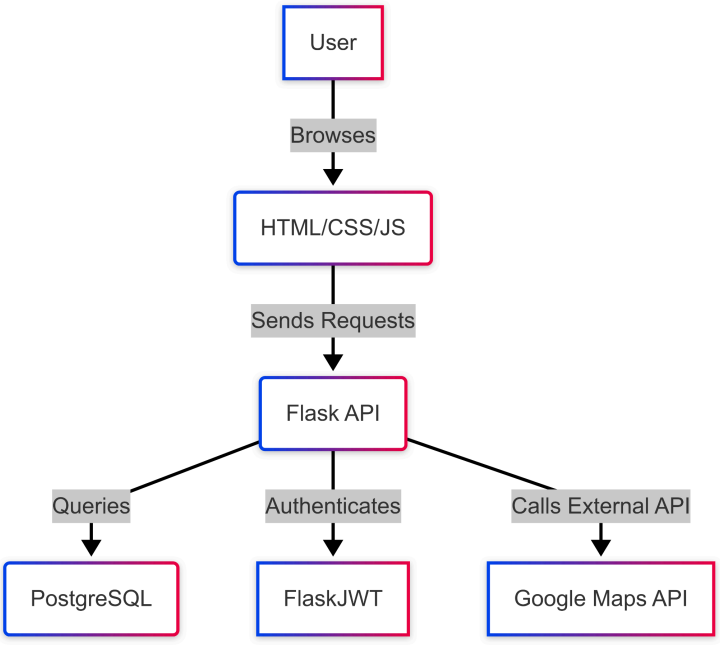
Authentication: Flask-JWT-Extended

- Provides secure user login and token-based authentication.

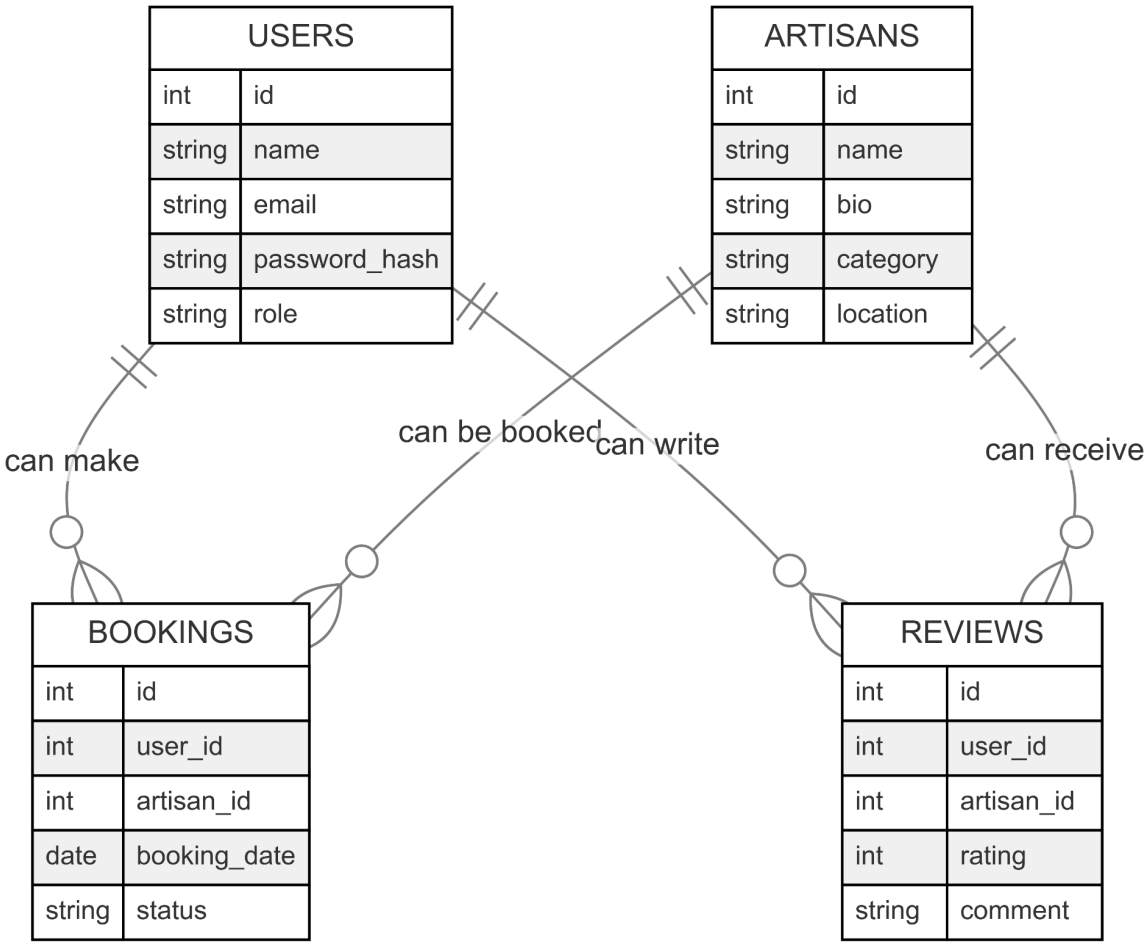
External APIs:

- **Google Maps API** → Fetches location data for artisans.

System Architecture Diagram:



Database Schema (SQL Implementation):

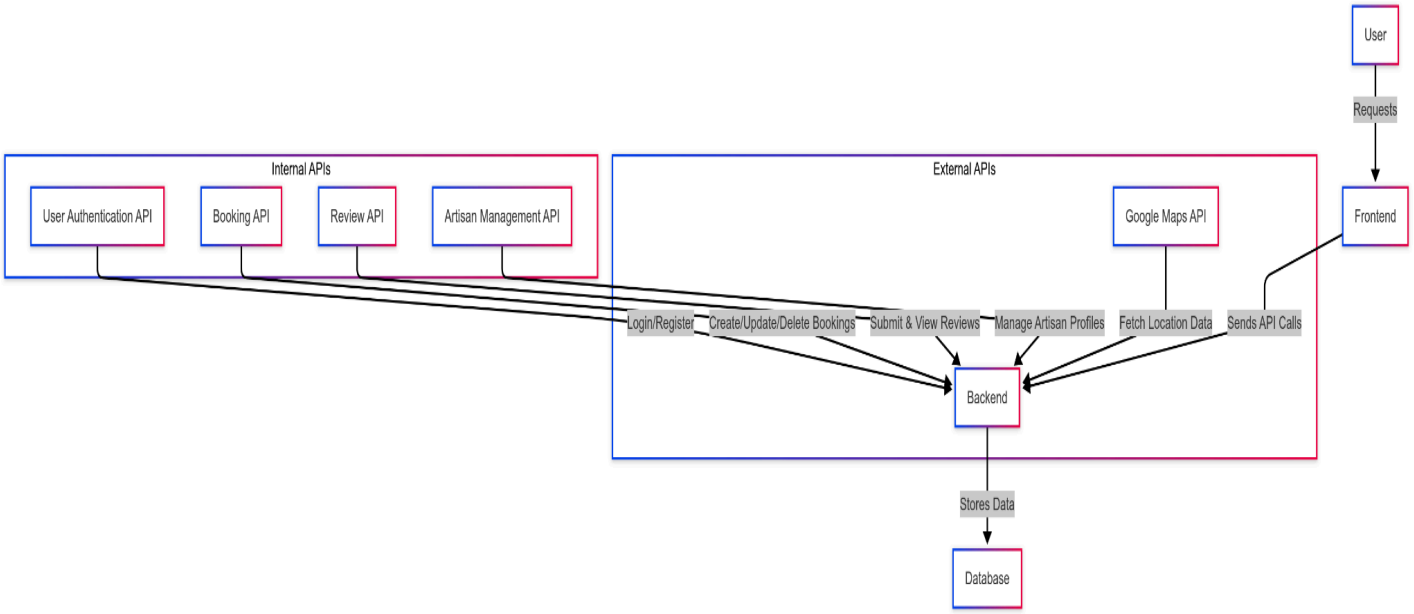


Indexing Strategy for Performance Optimization

INDEXES		
string	idx_users_email	ON users(email)
string	idx_bookings_user_id	ON bookings(user_id)
string	idx_bookings_artisan_id	ON bookings(artisan_id)

4. Document External and Internal APIs

A complete API reference is available including authentication, bookings, and reviews endpoints. This ensures proper integration and external service interaction.



5. Plan SCM and QA Strategies

- **SCM Strategy:** Git branching model, feature-based branches, and review process.
- **QA Strategy:** Unit testing, integration tests, and manual validation before production deployment.

6. Technical Justifications

All architectural and database design choices are based on scalability, security, and maintainability best practices.

9. Conclusion and Next Steps

Stage 3 of Vivvoy's development focuses on technical documentation, API definition, system architecture, database design, sequence diagrams, and quality assurance strategies. Authentication is implemented using Flask-JWT-Extended for security and session management. The front-end approach is based on HTML, CSS, and JavaScript. This document serves as a foundation for MVP development and further refinements.

Next Steps:

1. **Complete final refinements to wireframes/mockups** and collect stakeholder feedback.
2. **Implement authentication endpoints** and validate secure user sessions.
3. **Develop and test user interface components** using the finalized wireframes.
4. **Integrate the Google Maps API** for artisan location services.
5. **Test booking workflows** using real test data to validate the process.
6. **Set up automated CI/CD pipelines** for deployment and testing.
7. **Deploy the beta version** for early user testing and feedback collection.

By following these steps, we ensure a smooth transition from planning to implementation while maintaining alignment with project goals.