Software Engineering Seminar
Semester 2025-III
Workshop No. 2 — Application Design and UI Progress

Team Members:
Santiago Sanchez Moya
Santiago Ávila
Lilly Sofia Ayala Rojas

Computer Engineering Program

School of Engineering

Universidad Distrital Francisco José de Caldas

Professor: Eng. Carlos Andrés Sierra, M.Sc.

Full-time Adjunct Professor

October 2025

Contents

Introduction

This document corresponds to Workshop 2 – Software Architecture and Design of the Software Engineering Seminar course. The purpose of this workshop is to develop the main design artifacts that define the structural and visual foundations of the Essentia platform, a microservices-based web application for managing and exploring perfumes.

The workshop focuses on the design phase of the project, where system components, relationships, and workflows are formally represented through diagrams. These artifacts aim to clarify how the platform is organized, how its modules interact, and how users will experience the system through the web interface.

The deliverables include the UML Class Diagram, Architecture Diagram, Deployment Diagram, Business Process Models, and Web UI mockups. Together, they provide a comprehensive overview of the system's design, ensuring consistency between its conceptual, logical, and physical representations. This stage establishes the foundation for the following phases of implementation and testing within the Essentia project base for iterative development.

1 UML Class Diagrams

The UML Class Diagram defines the internal structure and relationships between the main entities and components of the Essentia platform. The system is divided into two main backend services:

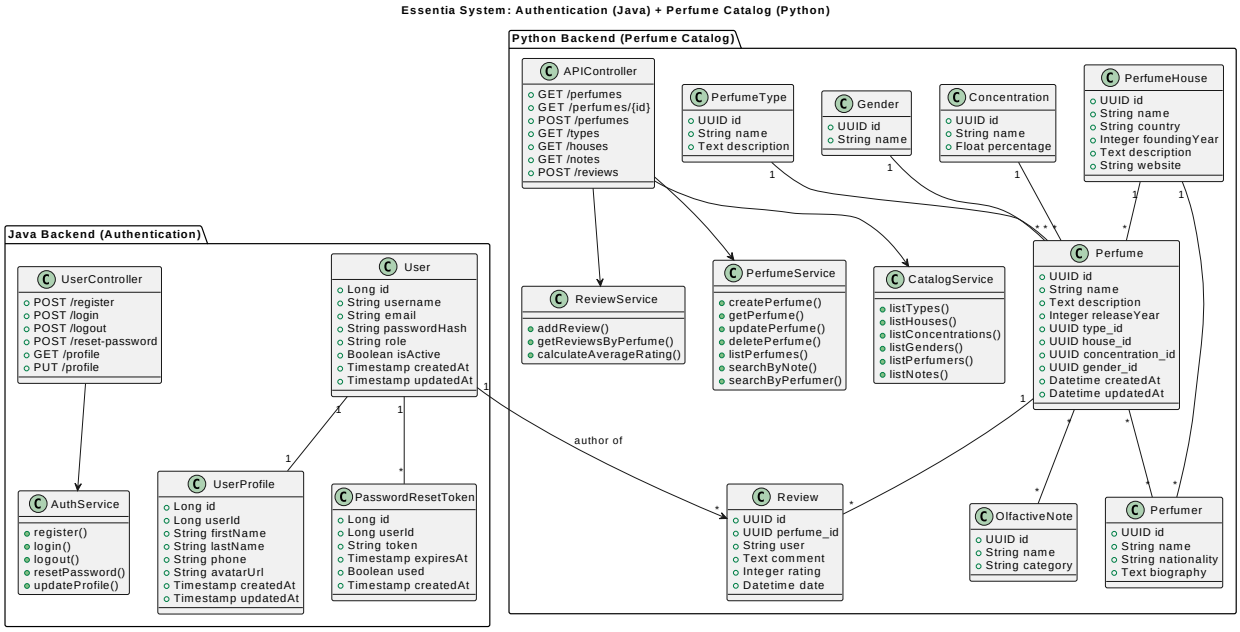

Figure 1: Essentia UML Diagram

- The Java Backend (Authentication) handles user registration, login, and profile management through classes such as UserController, AuthService, and User. It also manages user profiles and password recovery mechanisms.
- The Python Backend (Perfume Catalog) focuses on CRUD operations related to perfumes. It includes entities such as Perfume, PerfumeType, Perfumer, and Review, as well as services like PerfumeService, CatalogService, and ReviewService that define the business logic.

Each class specifies attributes and methods relevant to its purpose, and associations represent the interactions between components. This modular design supports scalability and separation of concerns, ensuring that authentication and catalog operations remain independent yet interoperable through REST APIs.

2 Architecture Diagram

The Architecture Diagram illustrates the high-level organization of Essentia, structured into three main layers

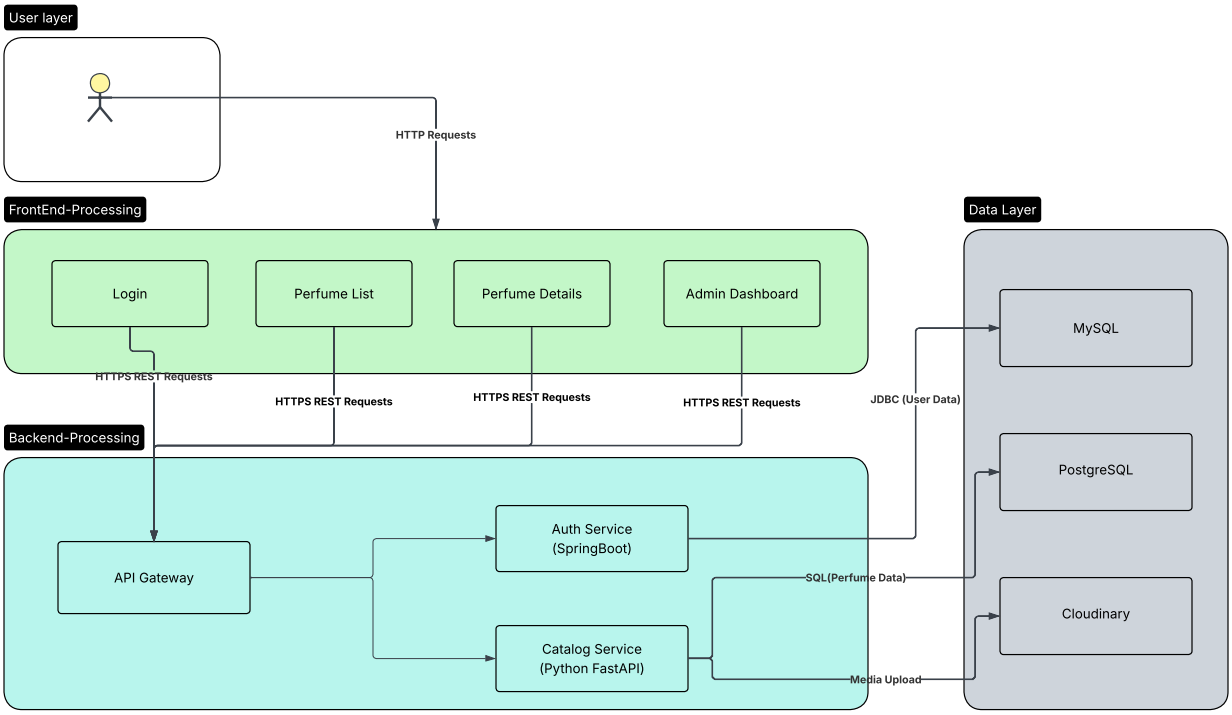

Figure 2: Essentia Architecture Diagram

- User Layer: Represents the end user interacting with the system through the web interface.
- Frontend Layer: Built using React and Next.js, this layer provides views for login, catalog browsing, perfume details, and the admin dashboard. It communicates with the backend via HTTPS REST requests.
- Backend Layer: Contains the API Gateway, which routes requests to the two backend services — the Auth Service (Spring Boot) for user management and the Catalog Service (FastAPI) for perfume data.
- Data Layer: Comprises three main components — a MySQL database for user information, a PostgreSQL database for perfume data, and Cloudinary for image storage.

This layered structure enhances maintainability and modularity while promoting secure, efficient communication between components.

3 Deployment Architecture

The Deployment Diagram shows how Essentia is deployed across various environments and cloud services.

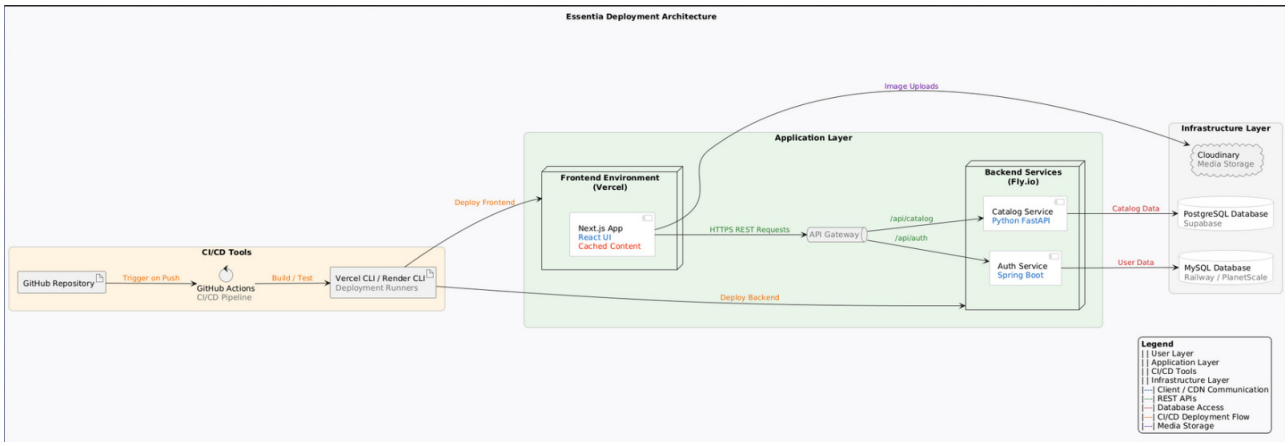

Figure 3: Essentia Deployment Diagram

The CI/CD process begins with the GitHub Repository, where commits trigger automated workflows in GitHub Actions. These workflows build, test, and deploy the services using tools like Vercel CLI or Render CLI.

The Frontend Environment (Vercel) hosts the Next.js application, delivering cached static content and managing client interactions. The Backend Services (Fly.io) run two separate containers — one for the Catalog Service (FastAPI) and another for the Auth Service (Spring Boot).

The backend connects to cloud-based storage and databases: Cloudinary for image uploads, PostgreSQL for catalog data, and MySQL for user data. This distributed deployment model enables independent scalability, resilience, and streamlined continuous integration.

4 Business Model Processes

Perfume Search and Review Flow – Process Description and Role The Perfume Search and Review Flow represents the core user journey and the foundation of the Essentia platform's value proposition. This end-to-end process begins when users access the perfume catalog and seamlessly progresses through the exploration, discovery, and engagement phases. It transforms simple browsing into a dynamic, interactive experience, directly aligning with Essentia's mission to help users "find their perfect fragrance."

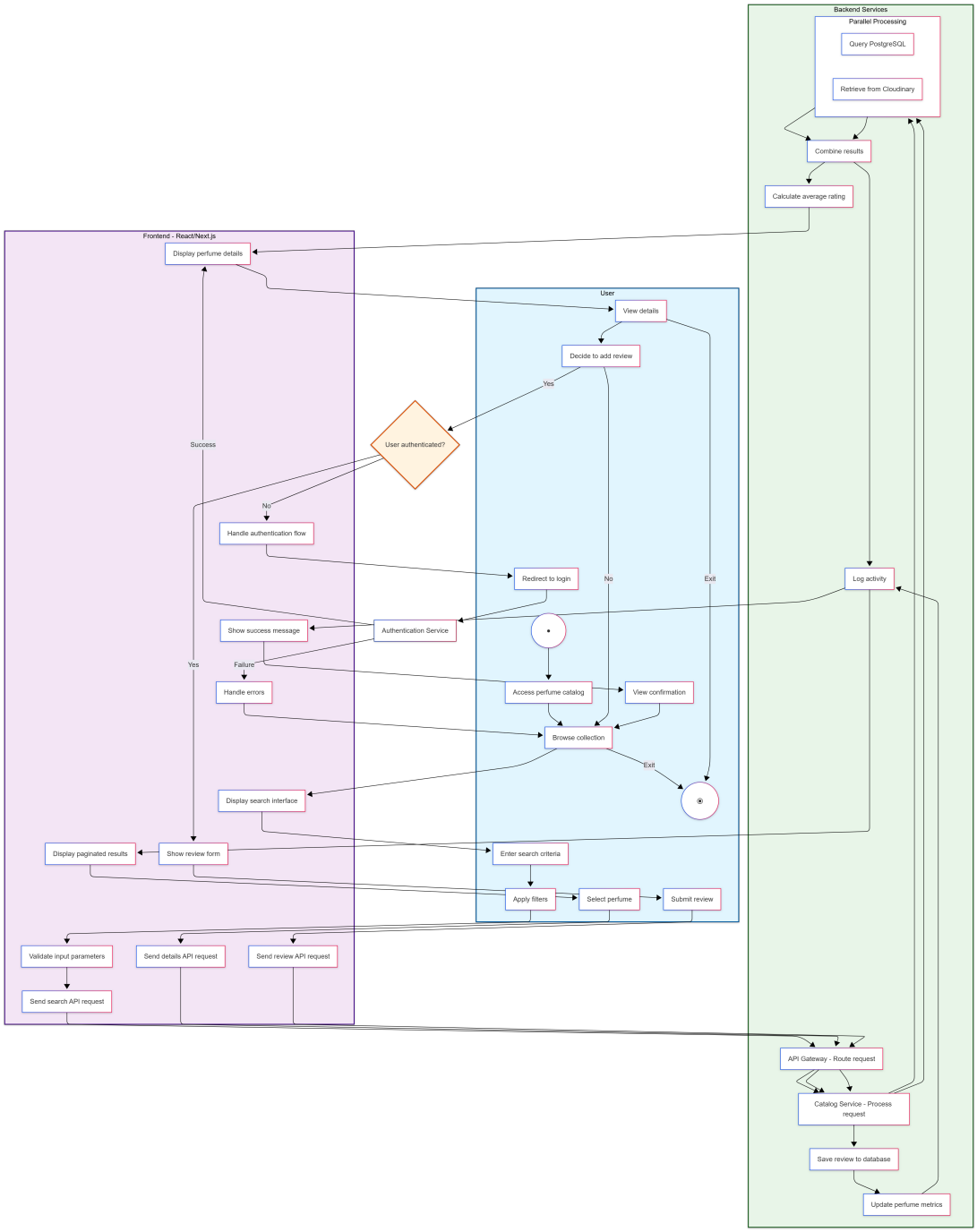

Figure 4: Activity Diagram: Perfume Search and Review Flow

The process initiates as the user enters the catalog interface. The front-end (built with React/Next.js) displays the search interface, validates input, and sends structured requests through the API Gateway. The Catalog Service then coordinates parallel processing, querying PostgreSQL for structured perfume data and retrieving images from Cloudinary. These results are combined, logged, and returned to the front-end as responsive, paginated listings that support intuitive navigation and selection. When a user selects a specific perfume, the system efficiently retrieves detailed information, including

brand, olfactory notes, concentration, and average rating, optimizing both performance and user satisfaction.

The next stage involves community engagement through reviews. If a user decides to contribute a review, the system verifies authentication via the Auth Service. Authenticated users can then submit their reviews, which the backend persists, updates relevant metrics, and logs for auditing and analytics. This cycle not only builds trust and transparency but also generates valuable data that supports future recommendation intelligence.

Architecturally, this process exemplifies Essentia's microservices design, showcasing modularity, scalability, and fault isolation. The API Gateway manages routing between the Catalog and Authentication services, while asynchronous operations and parallel queries minimize latency. Functionally, this flow drives significant business value by increasing user retention, fostering community-driven content, and providing actionable insights through search behavior and rating trends.

Ultimately, this process encapsulates Essentia's core identity: transforming a traditional perfume catalog into a community-powered discovery ecosystem, where each interaction enriches collective fragrance knowledge and enhances the personalized experience for every user.

5 Web UI Progress

The user interface (UI) of Essentia has been designed to offer a smooth, elegant, and intuitive browsing experience for perfume enthusiasts. The current web implementation showcases the main navigation flow through three core sections:

Homepage / Header: When users access the platform, they are greeted with the Essentia logo and navigation bar. The hero section displays the slogan "Discover Your Perfect Fragrance", inviting users to explore the collection. The interface uses soft tones and minimalistic design to highlight the brand's refined aesthetic.

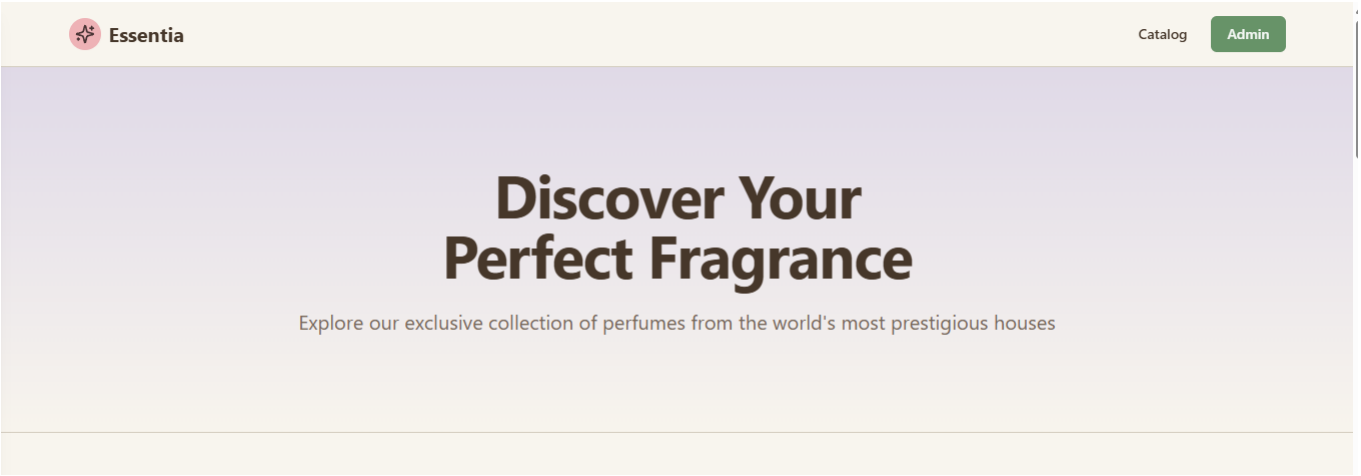

Figure 5: Essentia's Header

Search and Filter Section: Below the header, users can filter and search perfumes by brand, category, or type using a responsive search bar and dropdown filters. This section dynamically updates the results based on user input, enabling quick and efficient navigation through the catalog.

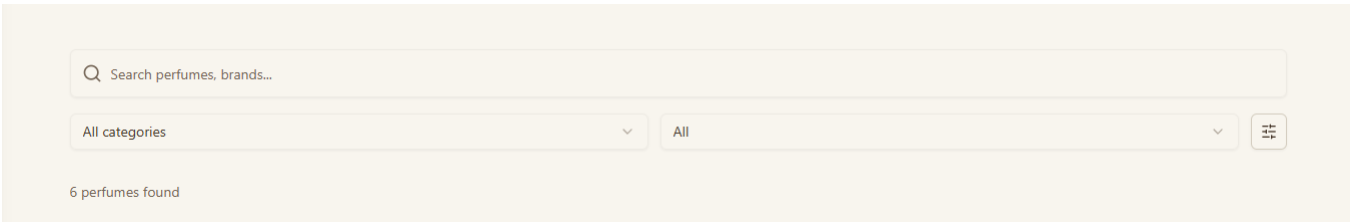

Figure 6: Essentia's SearchBar

Perfume Listing: The main section displays a grid of perfume cards, each showing an image, brand name, perfume title, rating, price, and category (e.g., Floral, Woody). This layout provides a visually engaging way to browse perfumes, ensuring both clarity and accessibility.

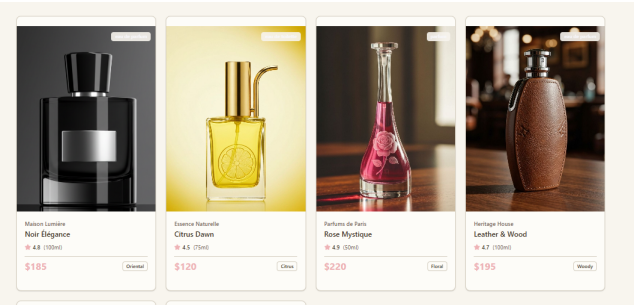

Figure 7: Essentia's perfume Listing

This UI represents the early phase of the web interface, focusing on user interaction, responsive design, and consistent visual identity. Future iterations will incorporate authentication and administrative CRUD functionalities.

References

[1] *Cloudairy* Recovered from `https://cloudairy.com/blog/understanding-key-software-architecture-diagram-patterns`

[2] How to draw Architectural Diagrams. *LucidChart* Recovered from `https://www.lucidchart.com/blog/how-to-draw-architectural-diagrams`

[3] Architecture Diagram *Avolutions Software* Recovered from `https://www.avolutionsoftware.com/news/enterprise-architecture-it-architecture-diagrams-modeling/`

[4] What is Activity Diagram? *Visual Paradigm* Recovered from `https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/`

[5] UML Activity Diagram Tutorial *Sparx Systems* Recovered from `https://sparxsystems.com/resources/tutorials/uml2/activity-diagram.html`

[6] Unified Modeling Language (UML) *Object Management Group* Recovered from `https://www.uml.org/`