NotSpotify

Antaret e grupit:

Rinor Kastrati | 212256094 | [rk56094@ubt-uni.net](mailto:rk56094@ubt-uni.net) | G6

Denis Quka | 212257749 | [dq57749@ubt-uni.net](mailto:dq57749@ubt-uni.net) | G6

Melos Studenica | 212256149 | [ms56149@ubt-uni.net](mailto:ms56149@ubt-uni.net) | G6

Rion Prekadini | 2122 | rp@ubt-uni.net | G6 -> ( Rion was offline )

Project Overview

Our music application is a web-based platform for music enthusiasts to discover and explore new music. The application will allow users to create profiles, search for songs, create playlist.

We have chosen to build our music application using Angular, ASP.NET and MySQL. Angular will be used for the frontend presentation layer, ASP.NET will be used for the backend business layer, and MySQL will be used for the database data layer.

Architecture Overview

Our music application will follow a three-layer architecture pattern, consisting of the following layers:

1. Presentation Layer ( Angular )
2. Business Layer ( ASP.NET )
3. Data Layer (MySQL)

The presentation layer will handle the user interface and user interaction (user experience), the business layer will handle business logic and data processing, and the data layer will handle data storage and retrieval.

We will be following the Agile Methodology (Agile Manifesto).

Presentation Layer

The presentation layer is responsible for handling the user interface and displaying the data to the user.

Angular

Angular is a popular JavaScript framework for building web applications. It provides a range of features and tools for building responsive and dynamic user interfaces. Some of the key features of Angular include:

Components: Components are the building blocks of the user interface. They define the structure and behavior of the user interface.

Templates: Templates define how the components should be displayed. They use HTML syntax to define the structure and content of the user interface.

Directives: Directives are used to add behavior to the components. They allow you to manipulate the DOM, handle user events, and more.

Services: Services are used to provide data and functionality to the components. They can be used to interact with APIs, handle data storage, and more.

Angular Material

In addition to Angular, we will also be using PrimeNG to build the user interface of our music application. PrimeNG is a UI component library for Angular that provides a rich set of pre-built UI components for building responsive and user-friendly applications. PrimeNG is built with accessibility in mind and provides a variety of customizable themes to fit the design needs of your application.

Some of the key components of PrimeNG include:

Data Grid: The Data Grid is a powerful component for displaying and manipulating large sets of data. It provides features like sorting, filtering, pagination, and editing.

Charts: PrimeNG provides a variety of charting components that allow you to visualize your data in different ways, including bar charts, line charts, pie charts, and more.

Dropdowns: Dropdowns are used to display a list of options and allow users to select one or more items. PrimeNG provides a variety of dropdown components, including single and multi-select dropdowns.

Calendar: The Calendar component allows users to select dates and times and provides features like range selection and event highlighting.

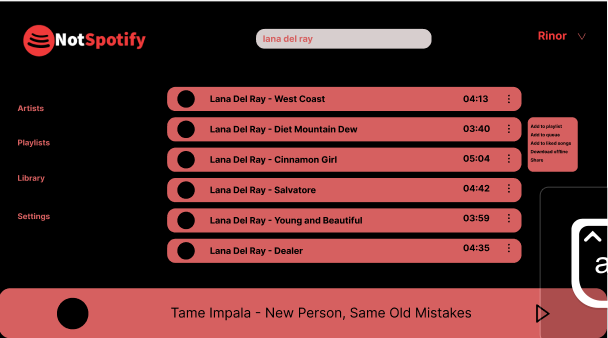
File Upload: The File Upload component allows users to upload files to your application and provides features like drag and drop support, file validation, and progress tracking.

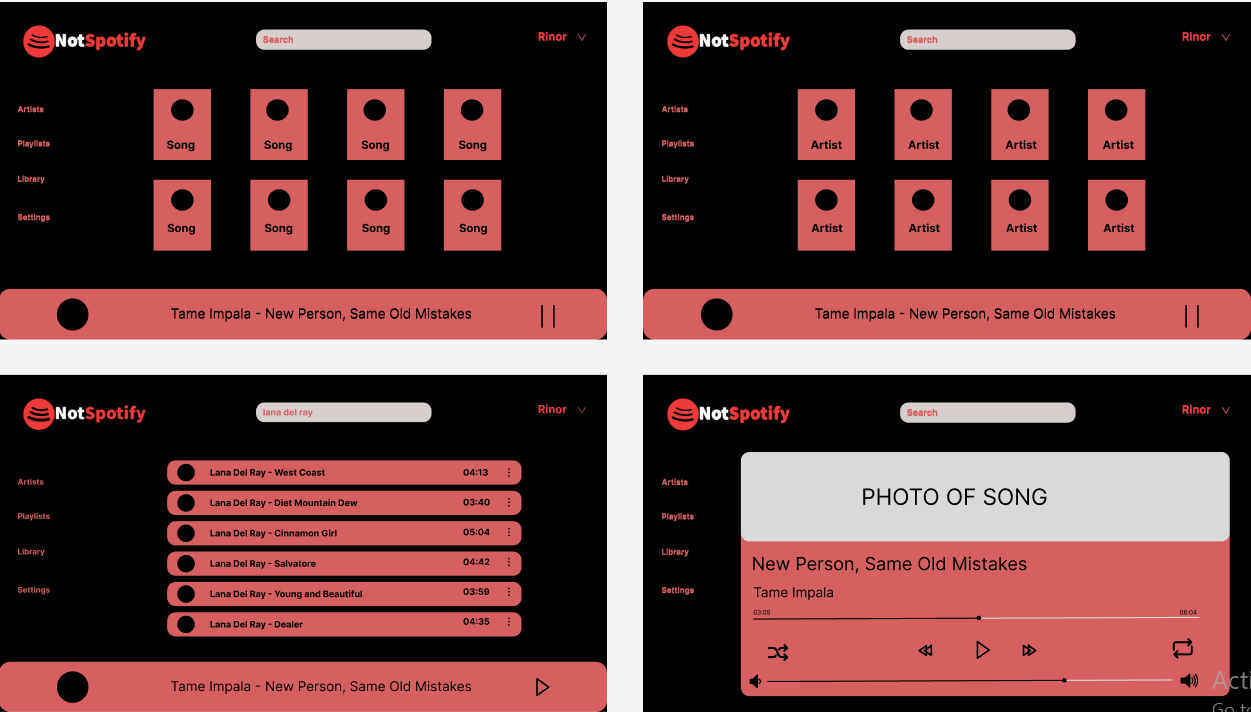
And much more.

Figma

To design the user interface of our music application, we will be using Figma. Figma is a cloud-based design tool that allows us to create wireframes, mockups, and prototypes of the user interface. It provides collaboration and sharing features that make it easy to work with the team.

Using PrimeNG and Figma, we can build a modern and responsive user interface for our music application.

Some of the prototypes : 



Business Layer

The business layer of our application will be implemented using ASP.NET, a popular web application framework developed by Microsoft. The business layer will be responsible for processing user requests, managing data validation, and handling business logic.

We will use the ASP.NET Web API to implement the RESTful API endpoints that the frontend will use to communicate with the backend. The ASP.NET Web API provides an easy way to define and expose endpoints for data retrieval and manipulation.

Data Layer

The data layer of our application will be implemented using MySQL, a popular open-source relational database management system. MySQL will be used to store and manage all data related to our music application, including user profiles, songs, playlists, and user activity.

We will use the Entity Framework Core as an Object-Relational Mapping (ORM) framework to interact with the database. Entity Framework Core will provide a way to define and interact with the database schema in C# code, rather than writing SQL queries directly.

Conclusion

In conclusion, our music application will follow a three-layer architecture pattern, with the presentation layer implemented using Angular, the business layer implemented using ASP.NET, and the data layer implemented using MySQL. By using these technologies and following this architecture pattern, we aim to build a scalable, maintainable, and user-friendly music application.

Entity Relationship Diagram

