

TALLINNA TEHNIKAÜLIKOOL
School of Information Technologies

Martin Metsküla 220650IADB
Jonathan Sillak 230656IADB

Ela - Event Discovery Application

Web applications in C#

Supervisor: Andres Käver

Tallinn 2024

Author's declaration of originality

I hereby certify that we are the sole authors of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Martin Metsküla

Author: Jonathan Sillak

03.03.2024

1. Project description

Ela is an event discovery app that helps users find, save, and create events. It allows users to browse events by category, location, date, and price. Users can also save events to a list and create profiles. In the future, users will be able to collect memories with friends. However, unlike other event-finding platforms in Estonia, which are often cluttered and unintuitive, Ela takes a unique approach. Inspired by Tinder's swipe format, Ela allows users to swipe through events they are interested in, making the process of finding things to do more efficient and engaging.

Ela is important because it addresses the shortcomings of existing event-finding platforms in Estonia. While other options are often cumbersome and difficult to navigate, Ela offers a streamlined and user-friendly experience. Additionally, by incorporating a Tinder-like swipe format, Ela makes discovering events more fun and engaging, encouraging users to explore new activities and connect with others who share their interests.

We are motivated to work on this project because we believe that Ela has the potential to revolutionize the way people in Estonia discover and participate in events. By providing a user-friendly and engaging platform, Ela can help people connect with their communities, explore new interests, and create lasting memories.

2. ERD schema

Events and EventHosts: The Events table stores information about various events, including details like name, description, timings, venue, and status. The EventHosts table links profiles to events, indicating which profiles are hosting which events. This relationship is essential for managing which users or agencies are responsible for specific events.

Profiles and Users: The Profiles table stores information about user profiles, which could represent individuals, artists, venues or agencies. The Users table holds account-related information, primarily focused on authentication (email and password). The UserProfiles table links users to profiles, establishing a relationship where a single user might have multiple types of profiles or roles. This setup is crucial for a system where a user can operate in different capacities, like an individual attendee or an organization hosting events.

EventLikes: This table allows profiles to like or show interest in events, creating a many-to-many relationship between profiles and events. This supports the core business logic of the app (liking events).

ResponseStatuses: This table seems to track responses or RSVPs to events from different profiles. It's crucial for managing attendance and engagement with events.

3. Technology stack

This section dives into the various technologies that power Ela. We meticulously chose each element to ensure a smooth, efficient, and user-friendly experience for both event attendees and organizers.

Frontend

React Native: Leveraging our existing team expertise in React, we opted for React Native to develop the mobile application. This selection enables the creation of a single codebase for both iOS and Android platforms, promoting development efficiency while delivering a native-like user experience.

Next.js: We leveraged Next.js for the event host dashboard due to its pre-rendering capabilities and familiarity with the framework. Next.js streamlines development by offering pre-built features and optimizations, allowing us to focus on creating a robust and user-friendly dashboard experience.

Backend

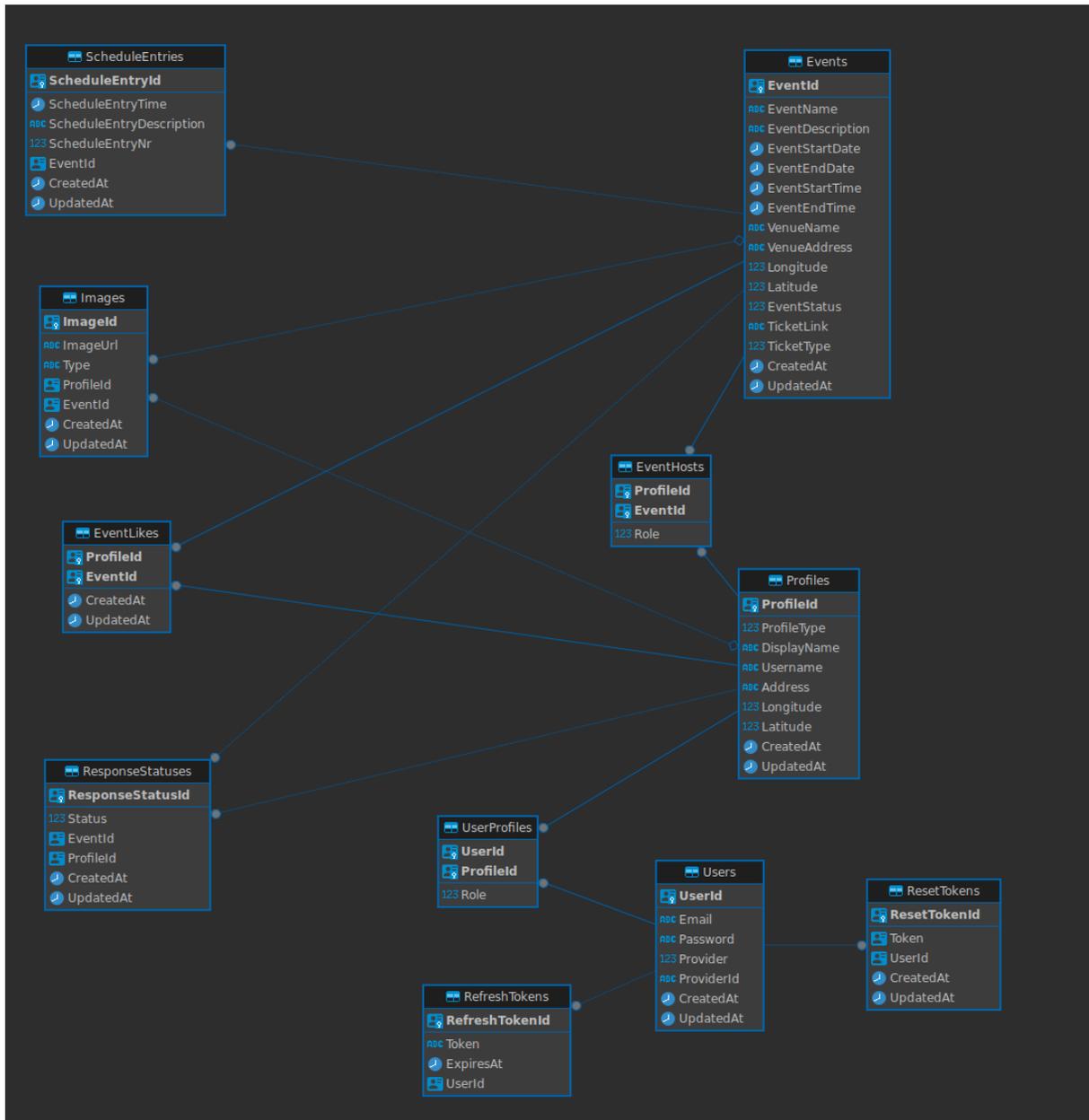
.NET Core API: The backend server utilizes a .NET Core API. This selection is based on its scalability, performance, and strong integration with other Microsoft technologies that might be considered for future enhancements.

PostgreSQL: PostgreSQL was chosen as the database management system due to its open-source nature, robust feature set, and well-established reputation for scalability and reliability.

Docker: Docker containers are employed to facilitate consistent and containerized deployments across various environments. This approach simplifies the deployment and maintenance processes.

JWT Authentication: In addition to supporting Google and Apple Login, Ela also utilizes a custom JSON Web Token (JWT) authentication system.

Diagrams

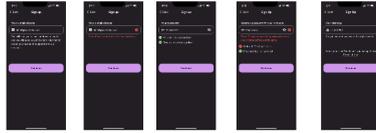


User flow & figma design

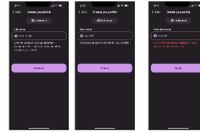
Welcome screen



Sign up: Account



Sign up: Profile



Log In

