TALLINN UNIVERSITY OF
TECHNOLOGY School of Information
Technologies

Mihkel Paloots 223784IADB

OutWeGo

Web Applications with C# project

Supervisor: Andres Käver

1. Introduction **4**
2. Analysis **5**
3. Wireframe **6**
1. User Page 6
2. Admin Page 7
4. ERD model **8**

Author’s declaration of originality

I hereby certify that I am the sole author of this project and this project has not been presented for examination or submitted for defence anywhere else. All used materials, references to the literature and work of others have been cited.

Author: Mihkel Paloots

25.02.2024

1. Introduction

This proposal outlines the development and implementation of OutWeGo, an innovative digital platform designed to revolutionise how individuals discover nightlife events within their cities. OutWeGo aims to serve as a bridge between entertainment venues, such as clubs and bars, and the general public, facilitating the promotion and discovery of nightlife events. By providing a centralized, user-friendly application where venue owners can list events and users can effortlessly find them, OutWeGo addresses the current informational gap in the nightlife industry. This project is motivated by the desire to streamline the process of finding nightlife enter tainment, thereby enhancing the urban nightlife experience for individuals and providing a valuable marketing tool for venues.

The motivation behind OutWeGo lies in the identified need to improve the connection between nightlife venues and potential visitors. The current method to find events is mainly Facebook events. Currently, Facebook doesn't provide an efficient search for events happening on certain dates. OutWeGo proposes a proactive approach, aggregating event information in a single platform, thus saving time and enhancing the decision-making process for users. This platform represents an opportunity for venue owners to reach a wider audience with minimal effort, potentially increasing event attendance.

2. Analysis

OutWeGo has two very important aspects that need deeper analysis. The first one is user-friendliness for visitors. The second one is up-to-date data. Both these aspects are vital for success.

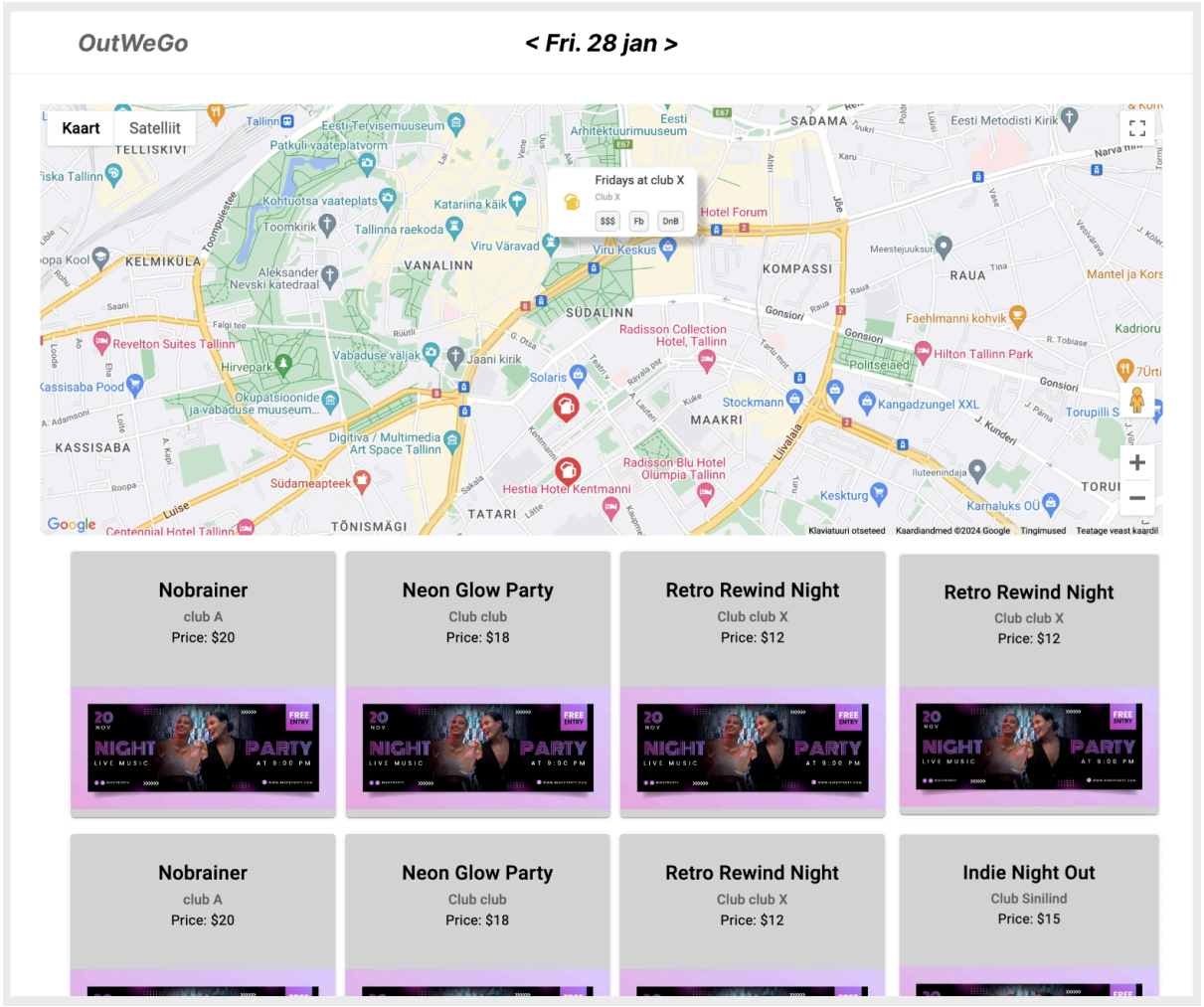
The main component for user-friendliness is a map that displays events. I plan to use Leaflet for map integration because this library provides all the extensive customization options I need and it is open-source. Different types of venues will be displayed differently and clicking on the venue marker will open detailed information about the venue and event happening that night.

Up-to-date data is achievable if venues enter all their events with the correct information. This is not a viable way to achieve this at the beginning, because some venues might not want to spend their time filling events information on an unknown page. The solution to this problem is Apify Facebook events scraper. Apify provides a scraper API that returns information about needed Facebook events. This API can be used with a cron job, to automate maintaining an up-to-date database. Later venue admins can modify data to their liking.

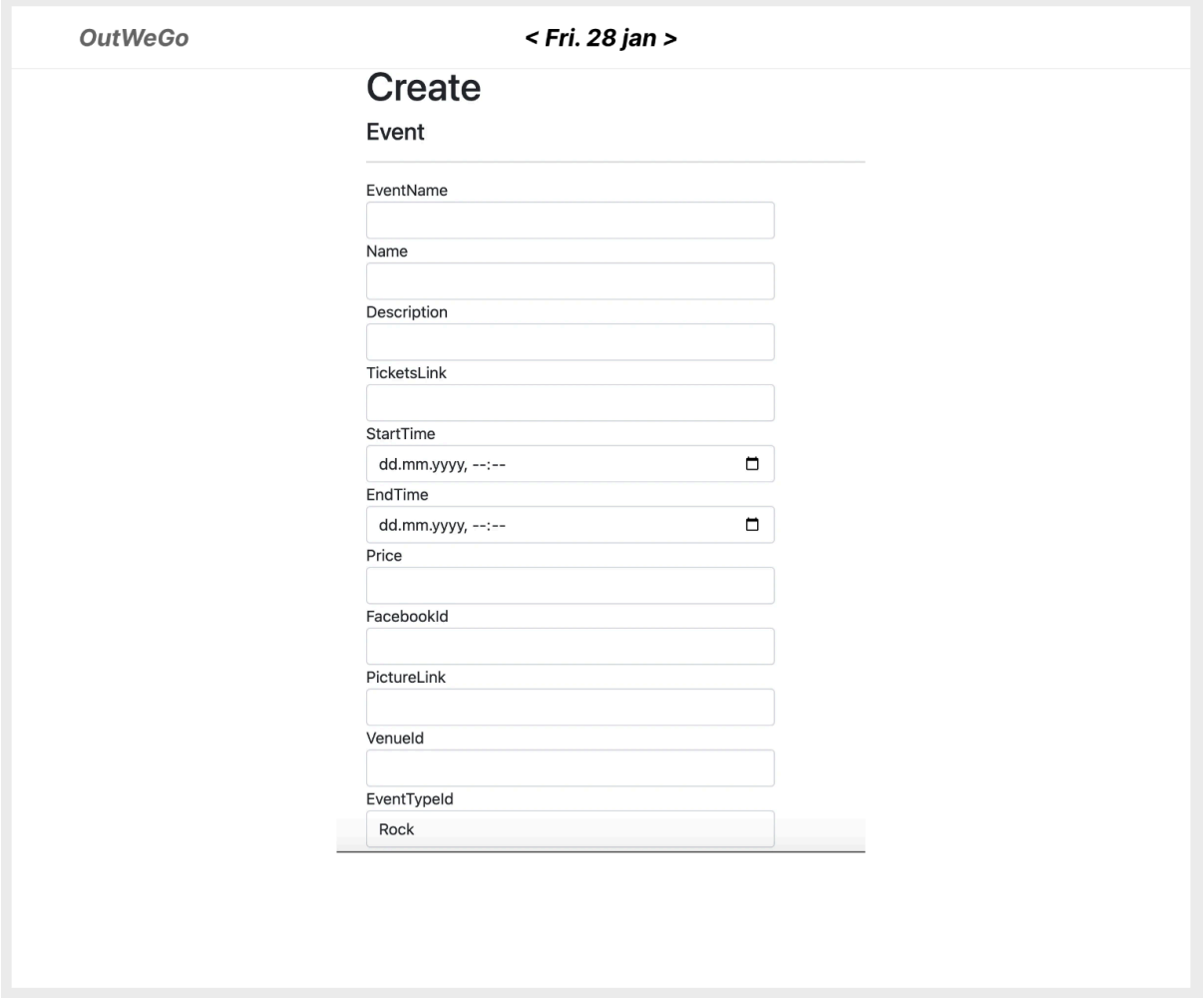
Another aspect is the legal aspect. As OutWeGo will be displaying event banners, agreements with every venue must be made before going live.

3. Wireframe

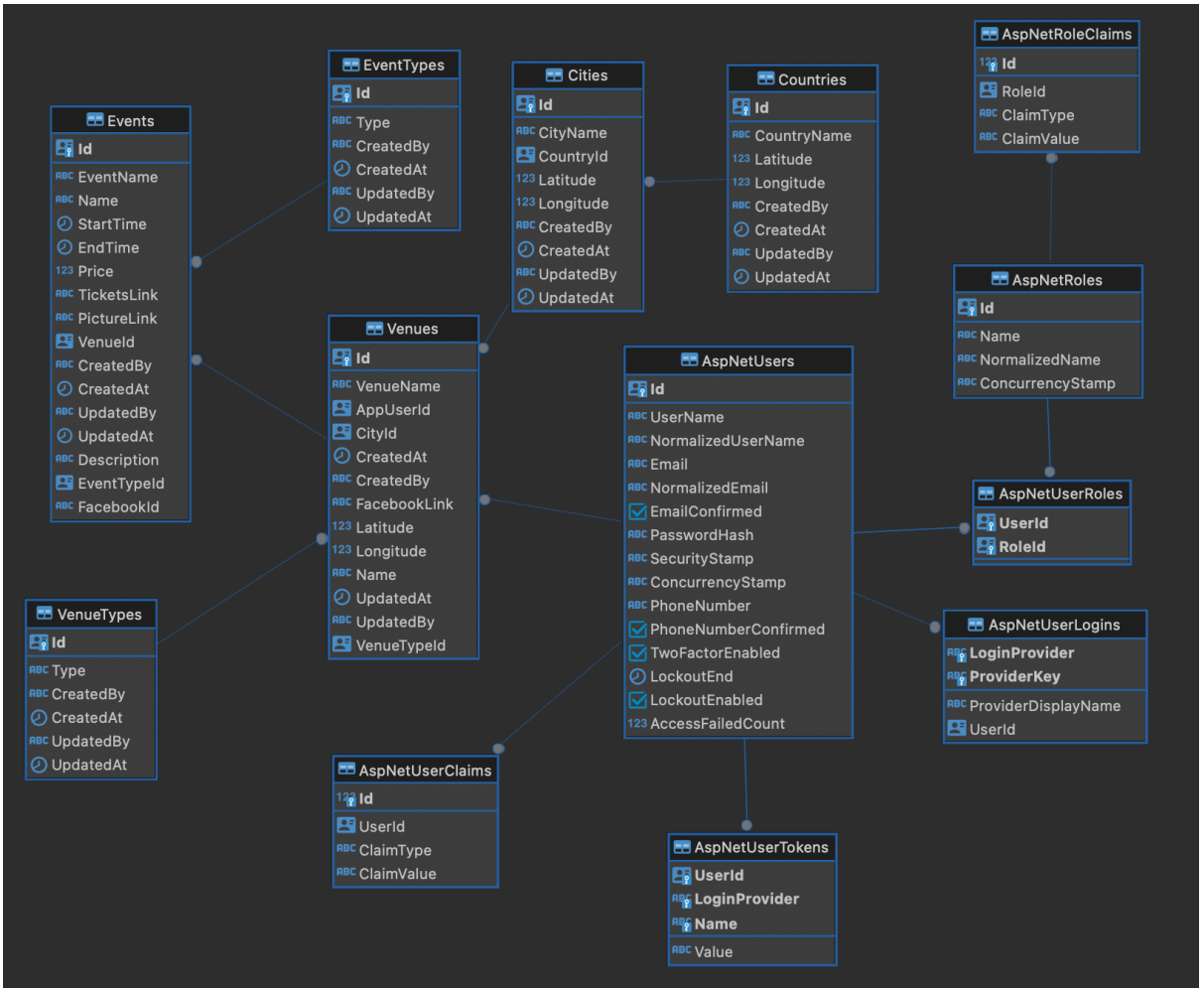
1. User Page



2. Admin Page


OutWeGo
< Fri. 28 jan >
Create
Event
EventName
Name
Description
TicketsLink
StartTime
dd.mm.yyyy, --:--
EndTime
dd.mm.yyyy, --:--
Price
FacebookId
PictureLink
VenueId
EventTypeId
Rock

4. ERD model


Events
Id
EventName
Name
StartTime
EndTime
Price
TicketsLink
PictureLink
VenueId
CreatedBy
CreatedAt
UpdatedBy
UpdatedAt
Description
EventTypeId
FacebookId
EventTypes
Id
Type
CreatedBy
CreatedAt
UpdatedBy
UpdatedAt
Cities
Id
CityName
CountryId
Latitude
Longitude
CreatedBy
CreatedAt
UpdatedBy
UpdatedAt
Countries
Id
CountryName
Latitude
Longitude
CreatedBy
CreatedAt
UpdatedBy
UpdatedAt
AspNetRoleClaims
Id
RoleId
ClaimType
ClaimValue
AspNetRoles
Id
Name
NormalizedName
ConcurrencyStamp
Venues
Id
VenueName
AppUserId
CityId
CreatedAt
CreatedBy
FacebookLink
Latitude
Longitude
Name
UpdatedAt
UpdatedBy
VenueTypeId
AspNetUsers
Id
UserName
NormalizedUserName
Email
NormalizedEmail
EmailConfirmed
PasswordHash
SecurityStamp
ConcurrencyStamp
PhoneNumber
PhoneNumberConfirmed
TwoFactorEnabled
LockoutEnd
LockoutEnabled
AccessFailedCount
AspNetUserRoles
UserId
RoleId
VenueTypes
Id
Type
CreatedBy
CreatedAt
UpdatedBy
UpdatedAt
AspNetUserLogins
LoginProvider
ProviderKey
ProviderDisplayName
UserId
AspNetUserClaims
Id
UserId
ClaimType
ClaimValue
AspNetUserTokens
UserId
LoginProvider
Name
Value