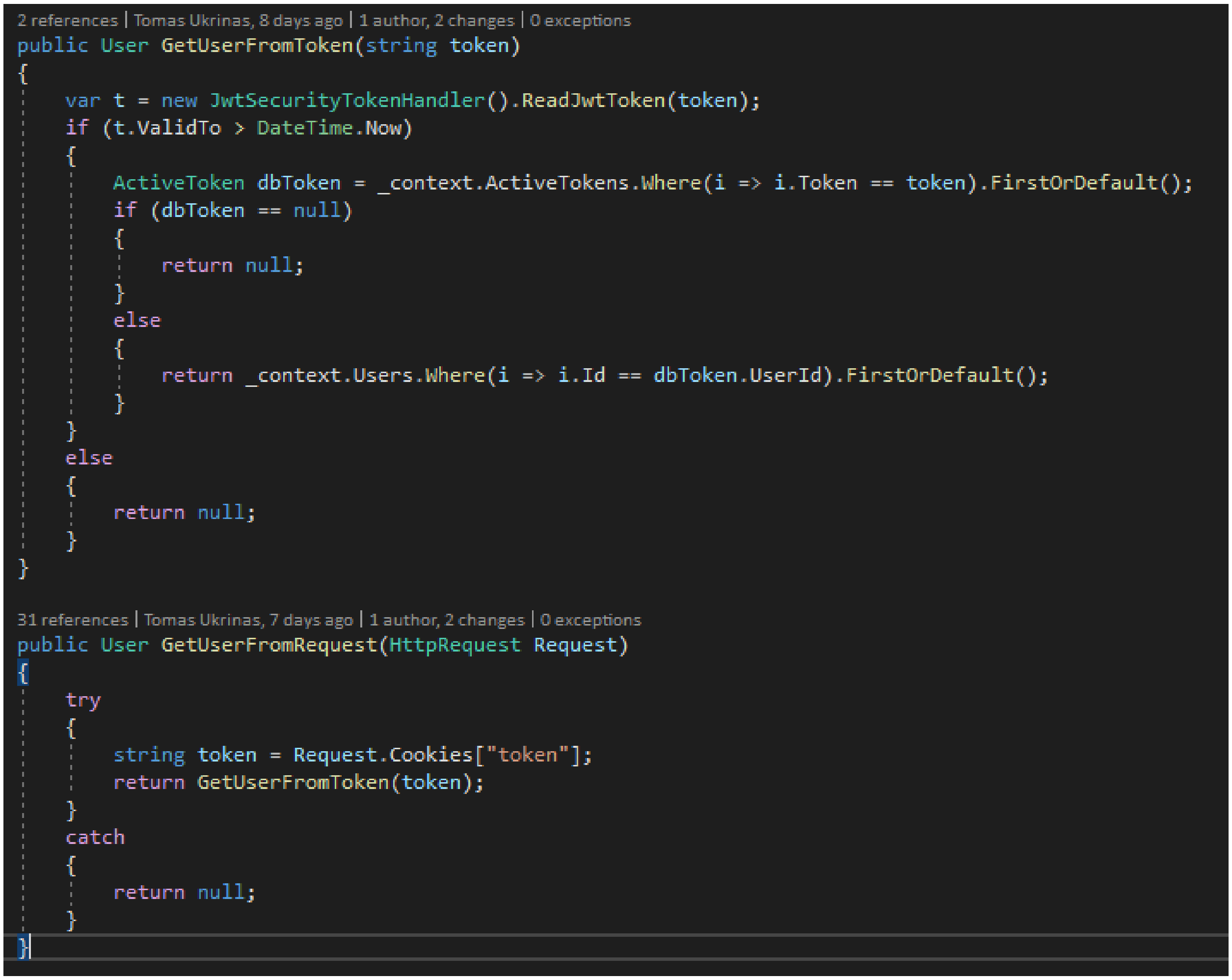
SeeSharp komandos techninė specifikacija

Concurency

Prisijungusiam vartotojui sugeneruojamas token, kuris išsaugomas naudotoj naršyklės sausainiukuose, taip pat mūsų duomenų bazėje. Su kiekviena sekančia užklausa, tikriname token'ą ir autorizuojame naudotoją.

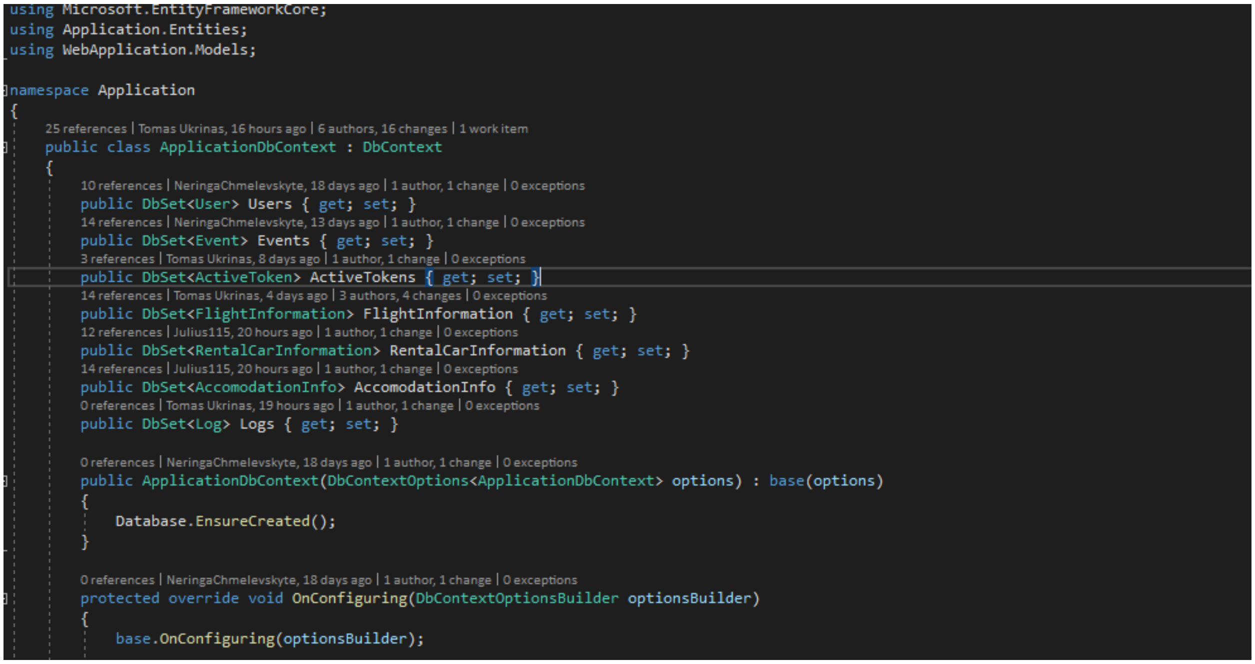


```
2 references | Tomas Ukrinas, 8 days ago | 1 author, 2 changes | 0 exceptions
public User GetUserFromToken(string token)
{
    var t = new JwtSecurityTokenHandler().ReadJwtToken(token);
    if (t.ValidTo > DateTime.Now)
    {
        ActiveToken dbToken = _context.ActiveTokens.Where(i => i.Token == token).FirstOrDefault();
        if (dbToken == null)
        {
            return null;
        }
        else
        {
            return _context.Users.Where(i => i.Id == dbToken.UserId).FirstOrDefault();
        }
    }
    else
    {
        return null;
    }
}

31 references | Tomas Ukrinas, 7 days ago | 1 author, 2 changes | 0 exceptions
public User GetUserFromRequest(HttpRequest Request)
{
    try
    {
        string token = Request.Cookies["token"];
        return GetUserFromToken(token);
    }
    catch
    {
        return null;
    }
}
```

UserService.cs

Security

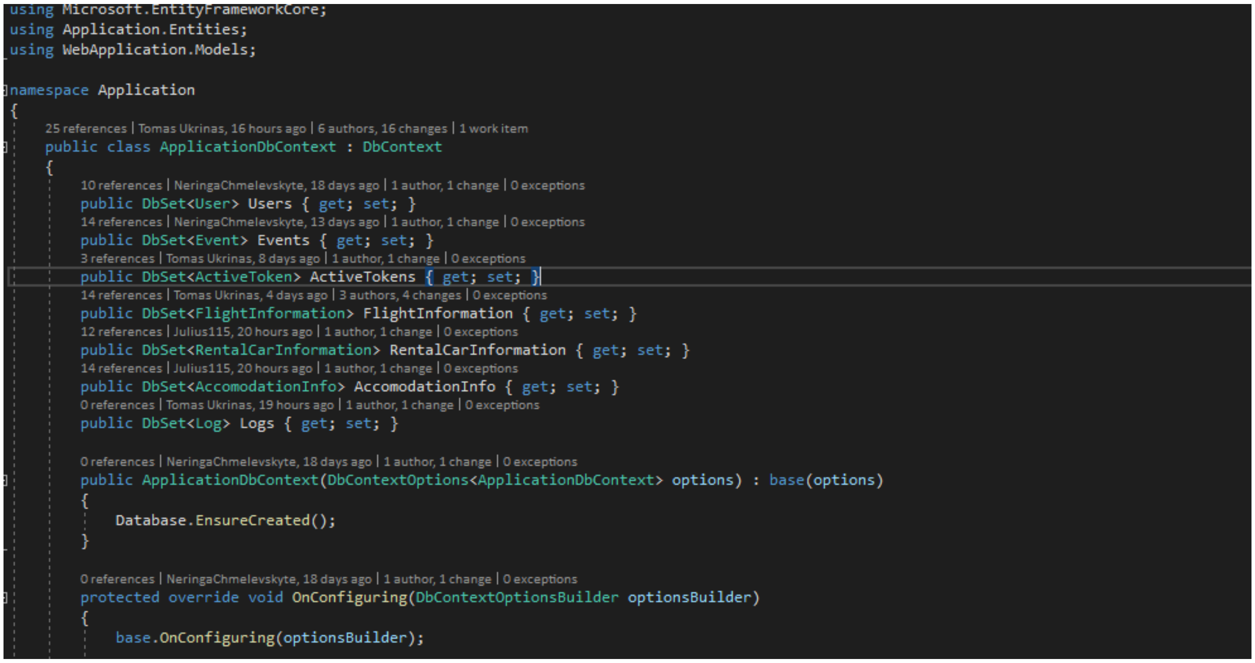
Naudojamas EntityFramework



```
using Microsoft.EntityFrameworkCore;
using Application.Entities;
using WebApplication.Models;

namespace Application
{
    25 references | Tomas Ukrinas, 16 hours ago | 6 authors, 16 changes | 1 work item
    public class ApplicationDbContext : DbContext
    {
        10 references | NeringaChmelevskyte, 18 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<User> Users { get; set; }
        14 references | NeringaChmelevskyte, 13 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<Event> Events { get; set; }
        3 references | Tomas Ukrinas, 8 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<ActiveToken> ActiveTokens { get; set; }
        14 references | Tomas Ukrinas, 4 days ago | 3 authors, 4 changes | 0 exceptions
        public DbSet<FlightInformation> FlightInformation { get; set; }
        12 references | Julius115, 20 hours ago | 1 author, 1 change | 0 exceptions
        public DbSet<RentalCarInformation> RentalCarInformation { get; set; }
        14 references | Julius115, 20 hours ago | 1 author, 1 change | 0 exceptions
        public DbSet<AccomodationInfo> AccomodationInfo { get; set; }
        0 references | Tomas Ukrinas, 19 hours ago | 1 author, 1 change | 0 exceptions
        public DbSet<Log> Logs { get; set; }

        0 references | NeringaChmelevskyte, 18 days ago | 1 author, 1 change | 0 exceptions
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
        {
            Database.EnsureCreated();
        }

        0 references | NeringaChmelevskyte, 18 days ago | 1 author, 1 change | 0 exceptions
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            base.OnConfiguring(optionsBuilder);
```

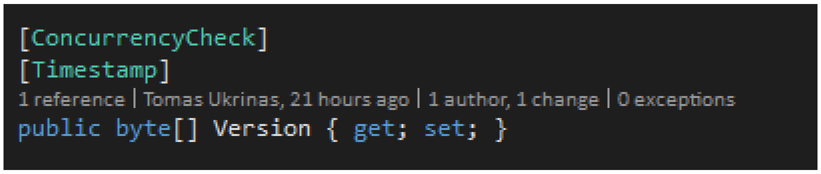
ApplicationDbContext.cs

DataAccess



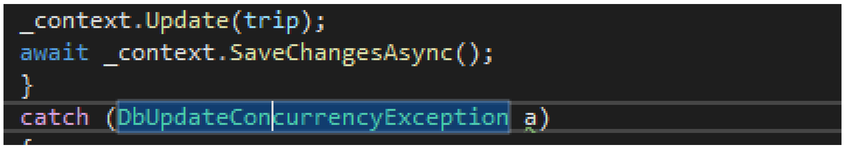
```
using Microsoft.EntityFrameworkCore;
using Application.Entities;
using WebApplication.Models;

namespace Application
{
    25 references | Tomas Ukrinas, 16 hours ago | 6 authors, 16 changes | 1 work item
    public class ApplicationDbContext : DbContext
    {
        10 references | NeringaChmelevskyte, 18 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<User> Users { get; set; }
        14 references | NeringaChmelevskyte, 13 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<Event> Events { get; set; }
        3 references | Tomas Ukrinas, 8 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<ActiveToken> ActiveTokens { get; set; }
        14 references | Tomas Ukrinas, 4 days ago | 3 authors, 4 changes | 0 exceptions
        public DbSet<FlightInformation> FlightInformation { get; set; }
        12 references | Julius115, 20 hours ago | 1 author, 1 change | 0 exceptions
        public DbSet<RentalCarInformation> RentalCarInformation { get; set; }
        14 references | Julius115, 20 hours ago | 1 author, 1 change | 0 exceptions
        public DbSet<AccomodationInfo> AccomodationInfo { get; set; }
        0 references | Tomas Ukrinas, 19 hours ago | 1 author, 1 change | 0 exceptions
        public DbSet<Log> Logs { get; set; }

        0 references | NeringaChmelevskyte, 18 days ago | 1 author, 1 change | 0 exceptions
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
        {
            Database.EnsureCreated();
        }

        0 references | NeringaChmelevskyte, 18 days ago | 1 author, 1 change | 0 exceptions
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            base.OnConfiguring(optionsBuilder);
```

Optimistic locking



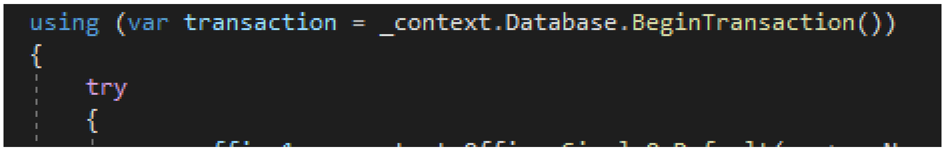
```
[ConcurrencyCheck]
[Timestamp]
1 reference | Tomas Ukrinas, 21 hours ago | 1 author, 1 change | 0 exceptions
public byte[] Version { get; set; }
```

Trip.cs



```
_context.Update(trip);
await _context.SaveChangesAsync();
}
catch (DbUpdateConcurrencyException a)
```

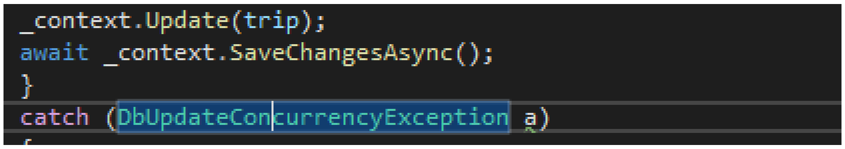
Memory management



```
 using (var transaction = _context.Database.BeginTransaction())
 {
     try
     {
```

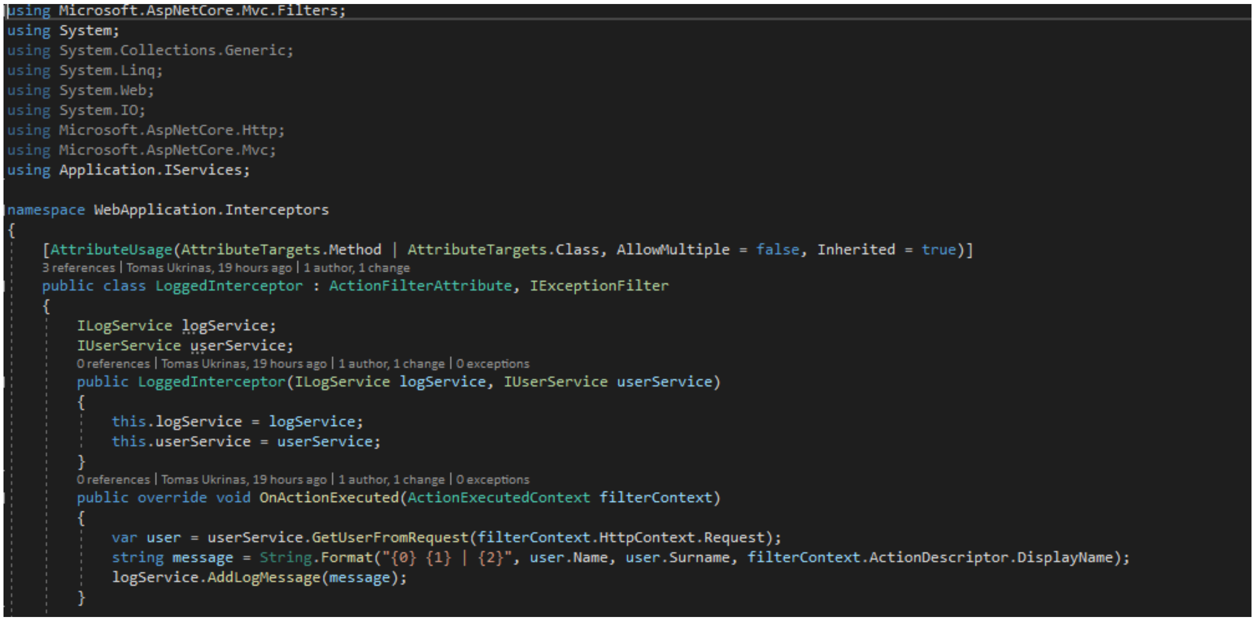
TripController.cs

Asynchronous/non-blocking communication



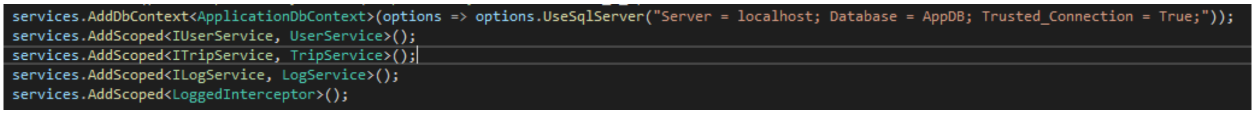
```
_context.Update(trip);
await _context.SaveChangesAsync();
}
catch (DbUpdateConcurrencyException a)
```

Cross-cutting functionality/Interceptors



```
using Microsoft.AspNetCore.Mvc.Filters;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Application.IServices;

namespace WebApplication.Interceptors
{
    [AttributeUsage(AttributeTargets.Method | AttributeTargets.Class, AllowMultiple = false, Inherited = true)]
    3 references | Tomas Ukrinas, 19 hours ago | 1 author, 1 change
    public class LoggedInterceptor : ActionFilterAttribute, IExceptionFilter
    {
        ILogService logService;
        IUserService userService;
        0 references | Tomas Ukrinas, 19 hours ago | 1 author, 1 change | 0 exceptions
        public LoggedInterceptor(ILogService logService, IUserService userService)
        {
            this.logService = logService;
            this.userService = userService;
        }
        0 references | Tomas Ukrinas, 19 hours ago | 1 author, 1 change | 0 exceptions
        public override void OnActionExecuted(ActionExecutedContext filterContext)
        {
            var user = userService.GetUserFromRequest(filterContext.HttpContext.Request);
            string message = String.Format("{0} {1} | {2}", user.Name, user.Surname, filterContext.ActionDescriptor.DisplayName);
            logService.AddLogMessage(message);
        }
```

Glass-box extensibility



```
services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer("Server = localhost; Database = AppDB; Trusted_Connection = True;"));
services.AddScoped<IUserService, UserService>();
services.AddScoped<ITripService, TripService>();
services.AddScoped<ILogService, LogService>();
services.AddScoped<LoggedInterceptor>();
```