Kino

Projekt z przedmiotu :
Tworzenie aplikacji internetowych i bazodanowych
Etap 2

Model-przykłady klas

```
[Table("Halls")]
Odwołania: 14
public class Hall
{
    [Key,Column("ID")]
    Odwołania: 2
    public int ID { get; set; }
    [Column("Rows")]
    Odwołania: 4
    public int Rows { get; set; }
    [Column("Columns")]
    Odwołania: 4
    public int Columns { get; set; }
    [Column("Full")]
    Odwołania: 4
    public bool Full { get; set; }
    Odwołania: 4
    public HallTechnologyEnum Technology { get; set; }
    Odwołania: 6
    public IEnumerable<Screening>? Screenings { get; set; }
    Odwołania: 6
    public IEnumerable<Seat> Seats { get; set; }
    Odwołania: 0
    public void Configure(EntityTypeBuilder<Hall> builder)
    {
        builder
        .HasMany(x => x.Seats)
        .WithOne(x => x.Hall)
        .OnDelete(DeleteBehavior.Cascade);

        builder
        .HasMany(x => x.Screenings)
        .WithOne(x => x.Hall)
        .OnDelete(DeleteBehavior.Cascade);
    }
}
Odwołania: 3
public enum HallTechnologyEnum
{
    IMAX,
    ScreenX,
    HDR,
    HFR
}
```

```
[Table("User")]
Odwołania: 13
public class User
{
    [Key, Column("ID")]
    1 odwołanie
    public int ID { get; set; }
    [MaxLength(50),Column("Login")]
    Odwołania: 3
    public string Login { get; set; }
    [MaxLength(50), Column("Password")]
    Odwołania: 3
    public string Password { get; set; }
    [Column("Type")]
    Odwołania: 3
    public UserTypeEnum Type { get; set; }
    [MaxLength(50), Column("Name")]
    Odwołania: 3
    public string Name { get; set; }
    [Column("CanReduce")]
    Odwołania: 3
    public bool CanReduce { get; set; }
    Odwołania: 5
    public IEnumerable<Ticket> ?Tickets { get; set; }
    Odwołania: 5
    public IEnumerable<Opinion> ?Opinions { get; set; }
    Odwołania: 0
    public void Configure(EntityTypeBuilder<User> builder)
    {
        builder
        .HasMany(x => x.Opinions)
        .WithOne(x => x.User)
        .OnDelete(DeleteBehavior.Cascade);

        builder
       .HasMany(x => x.Tickets)
       .WithOne(x => x.User)
       .OnDelete(DeleteBehavior.Cascade);
    }
}
Odwołania: 3
public enum UserTypeEnum
{
    Admin,
    User,
    Reviewer
}
```

DAL-CinemaContext

```
namespace DAL
{
    Odwołania: 17
    public class CinemaContext : DbContext
    {
        Odwołania: 7
        public DbSet<Hall> Hall { get; set; }
        Odwołania: 11
        public DbSet<Movie> Movie { get; set; }
        Odwołania: 5
        public DbSet<Opinion> Opinion { get; set; }
        Odwołania: 5
        public DbSet<Screening> Screening { get; set; }
        Odwołania: 5
        public DbSet<Seat> Seat { get; set; }
        Odwołania: 5
        public DbSet<Ticket> Ticket { get; set; }
        Odwołania: 5
        public DbSet<User> User { get; set; }


        Odwołania: 0
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=Cinema;Integrated Security=True;Connect Timeout=30;"+"" +
                "Encrypt=False;Trust Server Certificate=False;Application Intent=ReadWrite;Multi Subnet Failover=False");
        }
    }
}
```

BLL/BLL_EF-przykłady klas i implementacji

```
namespace BLL
{
    Odwołania: 4
    public interface IHallService
    {
        Odwołania: 2
        IEnumerable<HallResponseDTO> GetHalls();
        Odwołania: 2
        HallResponseDTO GetHall(int id);
        Odwołania: 2
        void DeleteHall(int id);
        Odwołania: 2
        void PutHall(int id, HallRequestDTO hallRequestDTO);
        Odwołania: 2
        void PostHall(HallRequestDTO hallRequestDTO);
    }
}
```

```
namespace BLL
{
    Odwołania: 4
    public interface IUserService
    {
        Odwołania: 2
        UserResponseDTO GetUser(int id);
        Odwołania: 2
        void DeleteUser(int id);
        Odwołania: 2
        void PutUser(int id, UserRequestDTO userRequestDTO);
        Odwołania: 2
        void PostUser(UserRequestDTO userRequestDTO);
    }
}
```

```
namespace BLL_EF
{
    Odwołania: 2
    public class HallService : IHallService
    {
        private CinemaContext dbContext;
        Odwołania: 0
        public HallService(CinemaContext dbContext) { this.dbContext = dbContext; }

        Odwołania: 2
        public void PostHall(HallRequestDTO hallRequestDTO)
        {
            Hall hall = new Hall
            {
                Rows = hallRequestDTO.Rows,
                Columns = hallRequestDTO.Columns,
                Full = hallRequestDTO.Full,
                Technology = hallRequestDTO.Technology,
                Screenings = hallRequestDTO.Screenings,
                Seats = hallRequestDTO.Seats
            };
            dbContext.Hall.Add(hall);
            dbContext.SaveChanges();
        }
```

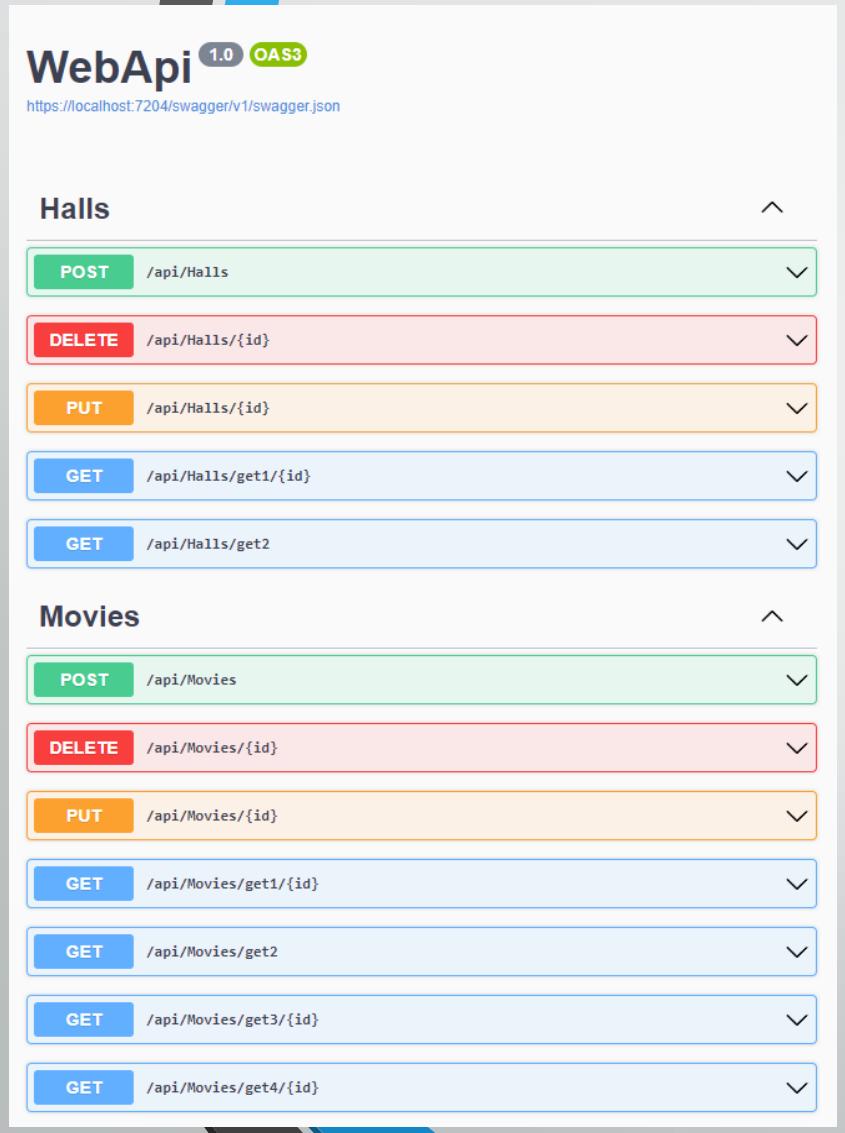
```
namespace BLL_EF
{
    Odwołania: 2
    public class UserService : IUserService
    {
        private CinemaContext dbContext;
        Odwołania: 0
        public UserService(CinemaContext dbContext) { this.dbContext = dbContext; }

        Odwołania: 2
        public void PostUser(UserRequestDTO userRequestDTO)
        {
            User user = new()
            {
                Login = userRequestDTO.Login,
                Password = userRequestDTO.Password,
                Type = userRequestDTO.Type,
                Name = userRequestDTO.Name,
                CanReduce = userRequestDTO.CanReduce,
                Tickets = userRequestDTO.Tickets,
                Opinions = userRequestDTO.Opinions,
            };
            dbContext.User.Add(user);
            dbContext.SaveChanges();
        }
```

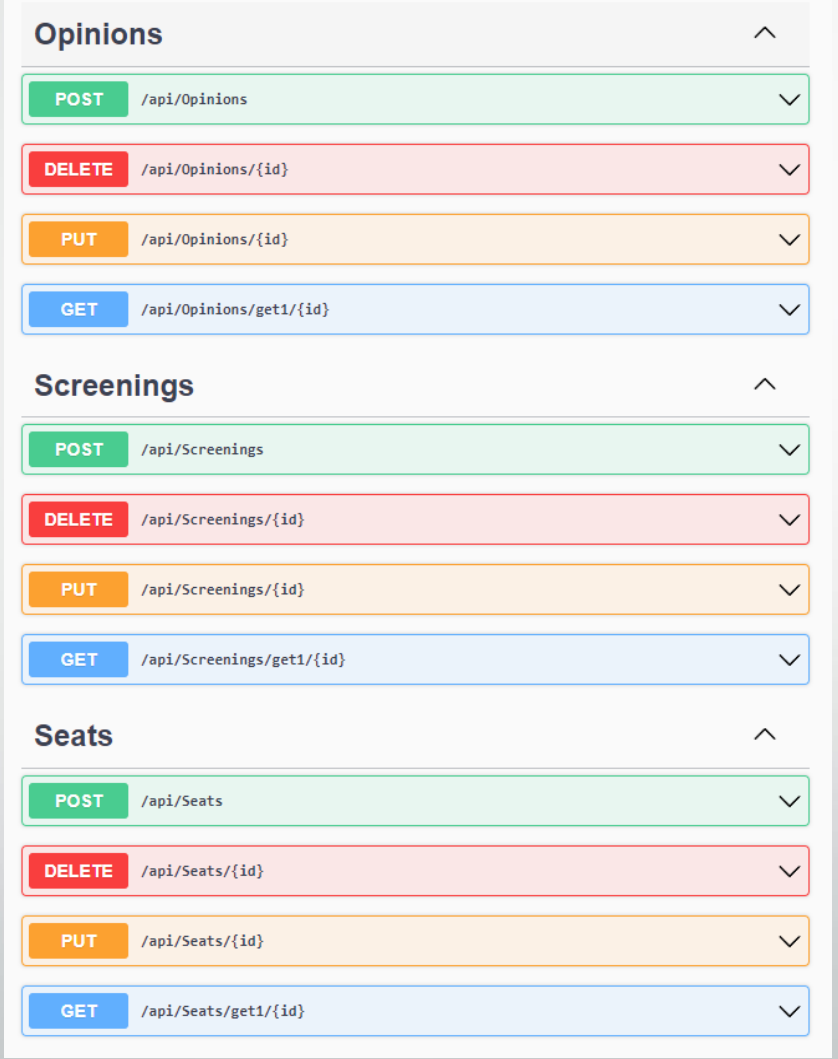
Kontrolery-przykłady implementacji

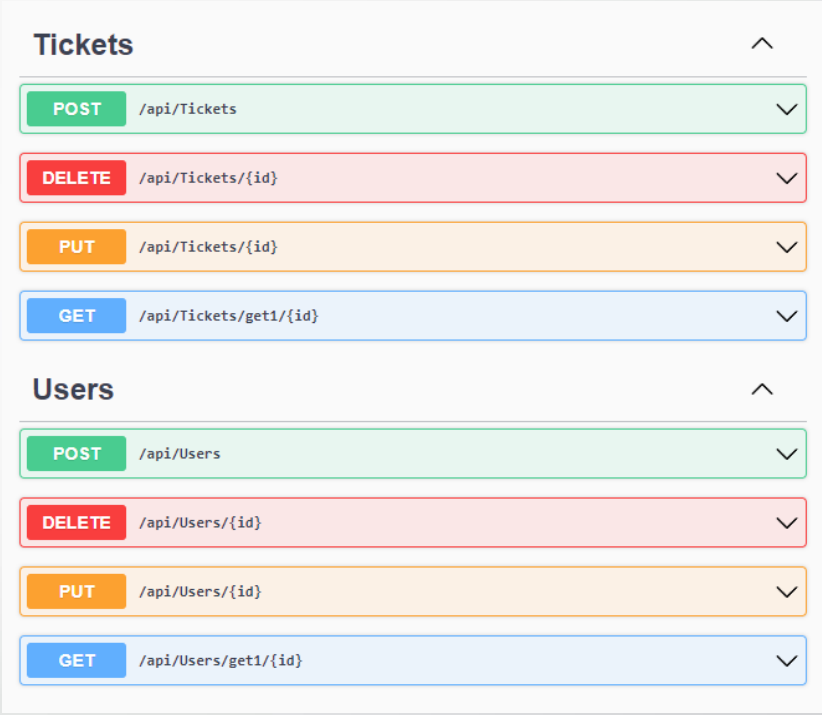
```
namespace WebApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 odwołanie
    public class HallsController : ControllerBase
    {
        readonly IHallService hallService;
        Odwołania: 0
        public HallsController(IHallService hallService) { this.hallService = hallService; }

        [HttpPost]
        Odwołania: 0
        public void Post([FromQuery] HallRequestDTO hallRequestDTO)
        {
            this.hallService.PostHall(hallRequestDTO);
        }
```

```
namespace WebApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 odwołanie
    public class UsersController : ControllerBase
    {
        readonly IUserService userService;
        Odwołania: 0
        public UsersController(IUserService userService) { this.userService = userService; }

        [HttpPost]
        Odwołania: 0
        public void Post([FromQuery] UserRequestDTO userRequestDTO)
        {
            this.userService.PostUser(userRequestDTO);
        }
```

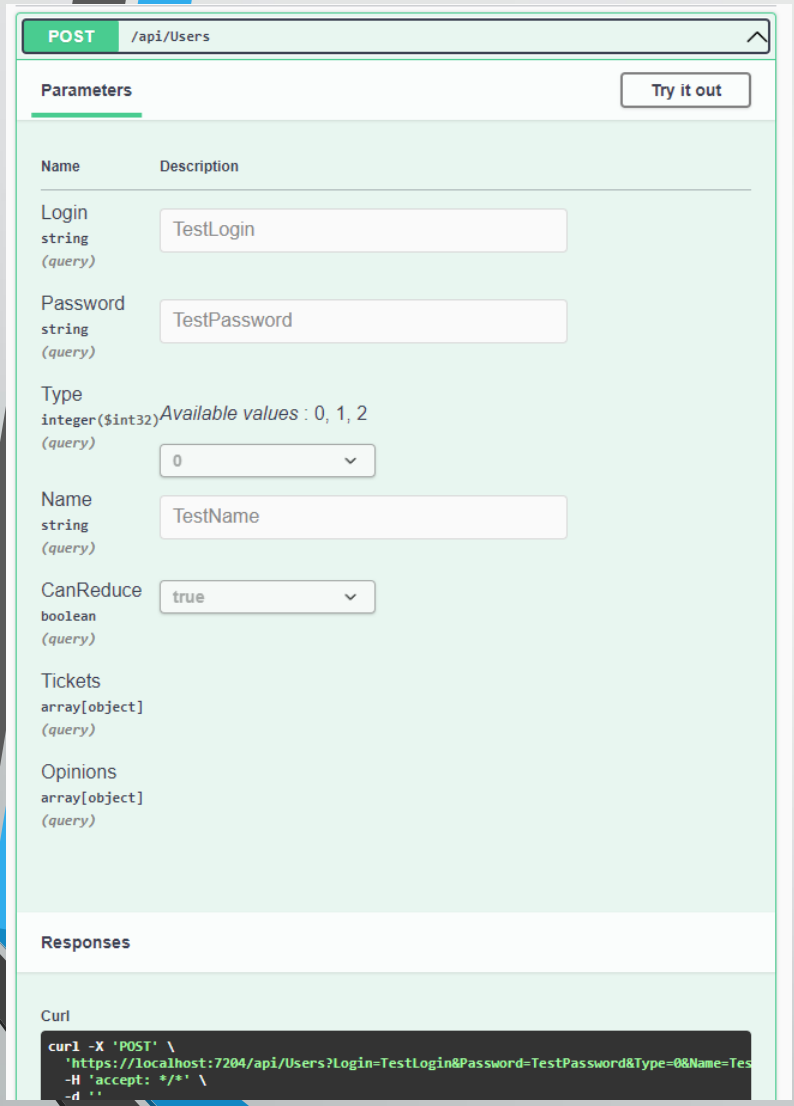
Wszystkie metody w WebAPI-Swagger

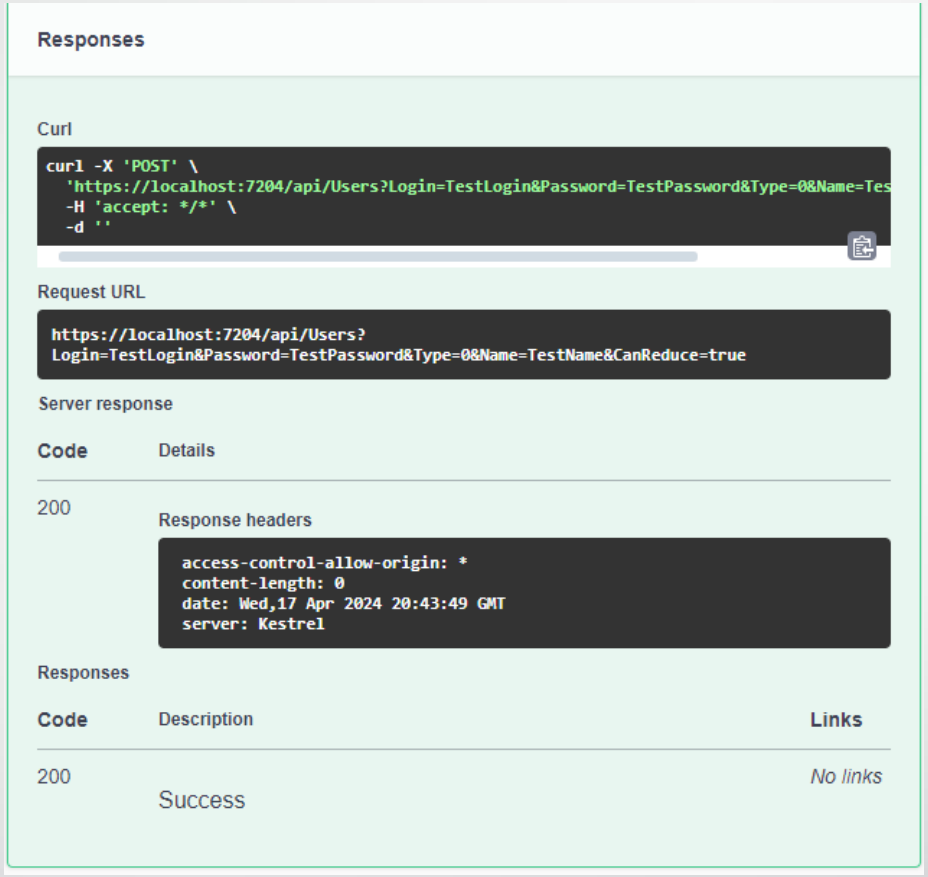


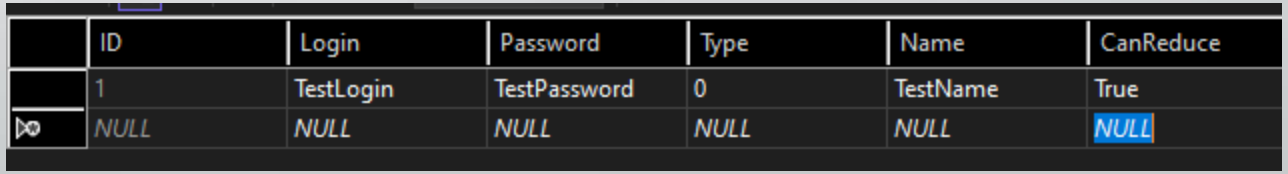




Przykład działania metody

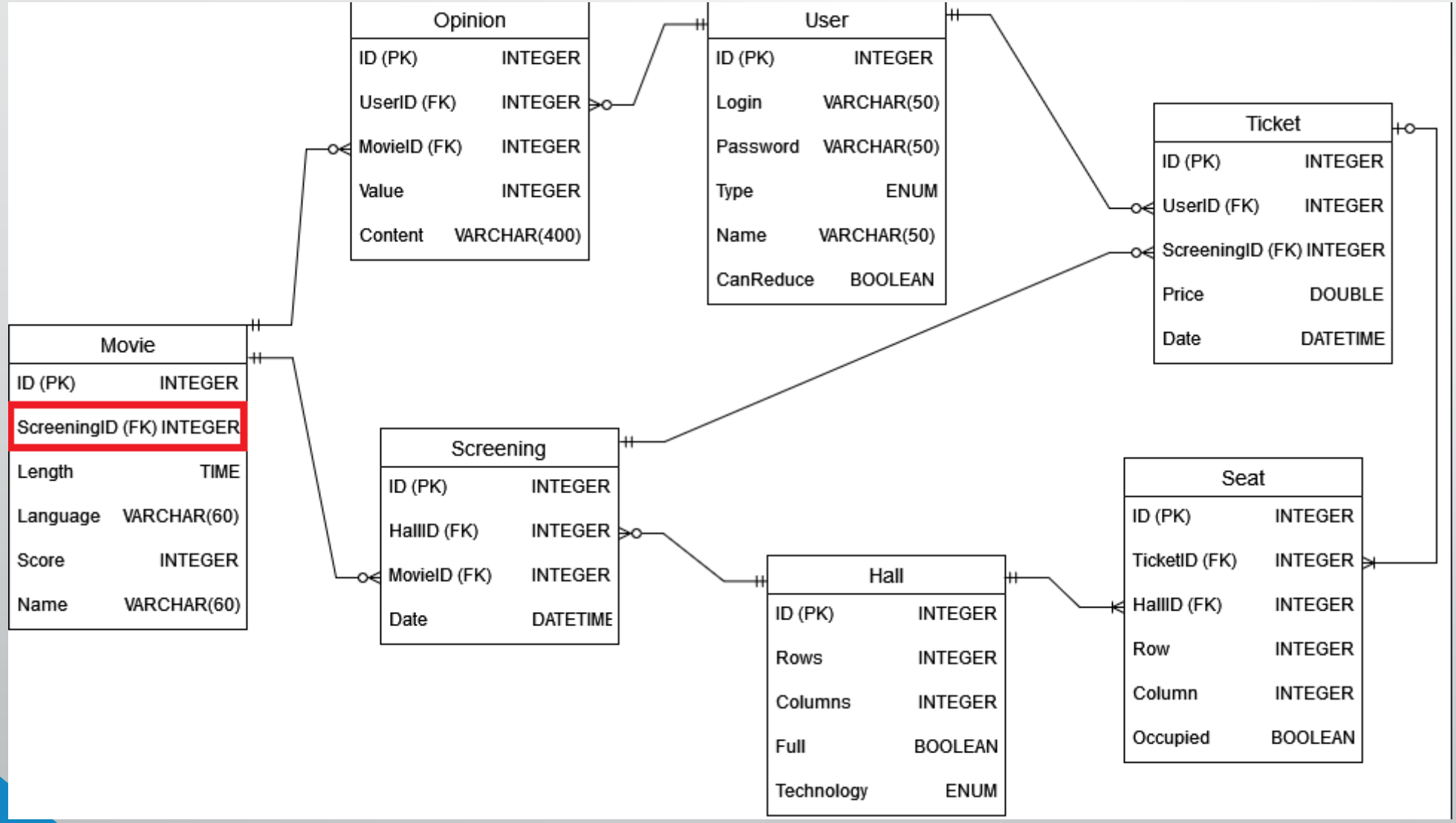






	ID	Login	Password	Type	Name	CanReduce
	1	TestLogin	TestPassword	0	TestName	True
	NULL	NULL	NULL	NULL	NULL	NULL

Zmiana w Diagramie ERD



Dziękujemy za uwagę!

Autorzy:
KacperRygał
Mikołaj Hanusz
IPpp30