



Факультет Компьютерных Наук
Департамент Программной Инженерии

Rewind

Приложение для коллективного медиа-потребления

Application for Collective Media Consumption

Выполнили:

Студент группы БПИ228
Рахманов Данила Дмитриевич

Студентка группы БПИ229
Хрулёва Александра Денисовна

Руководитель:

Приглашённый преподаватель Департамента Программной Инженерии
Сосновский Григорий Михайлович

Проблема и актуальность

Цель

Разработать приложение, которое позволит друзьям, членам семьи, коллегам по работе и другим знакомым между собой людям вспомнить яркие моменты их общения

Задачи

- Исследование и анализ существующих аналогов
- Проектирование базы данных
- Выбор архитектуры серверной части приложения
- Разделение кода доступа к БД на изолированные слои
- Использование представлений
- Тестирование работы приложения

Анализ аналогов

При проведении анализа было выявлено 3 прямых и 4 косвенных аналога. Сравнительный анализ подробно описано в программной документации.



Locket Labs, Inc.



Apple



TIMEHOP INC.

VS



Telegram FZ-LLC



V Kontakte OOO



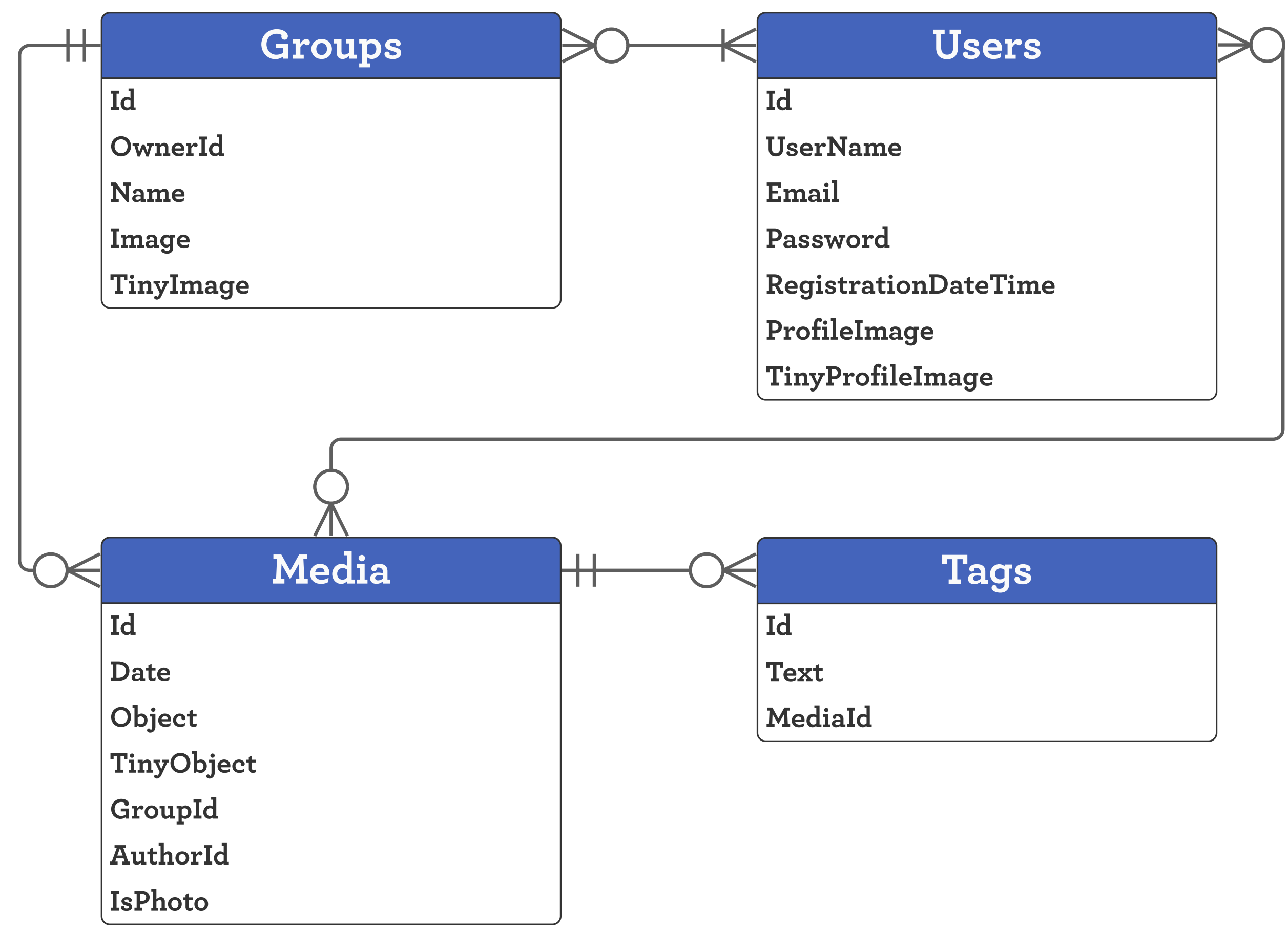
Direct Cursus Computer
System Trading



Google



Структура БД



```

  ▾ tables 7
    > _EFMigrationsHistory
    > Groups
    > GroupUser
    > Media
    > MediaUser
    > Tags
    > Users
  ▾ views 3
    > GroupViews
    > MediaViews
    > UserViews

```


Представления сущностей

```
public class UserView
{
    public int Id { get; set; }
    public string Username { get; set; } = string.Empty;
    public byte[] TinyProfileImage { get; set; } = Array.Empty<byte>();
}
```

```
public class MediaView
{
    [Key] public int Id { get; set; }
    [Required] public byte[] TinyObject { get; set; } = Array.Empty<byte>();
}
```

```
public class GroupView
{
    public int Id { get; set; }
    public string Name { get; set; } = string.Empty;
    public int? OwnerId { get; set; }
    public byte[] TinyImage { get; set; } = Array.Empty<byte>();
    public int GallerySize { get; set; }
}
```

Views

```
var resultResponse = new GroupInfoResponse {
    Id = group.Id,
    DataSize = dataSize,
    Name = group.Name,
    Image = group.Image,
    Owner = _context.UserViews.SingleOrDefault(uv:UserView => uv.Id == owner.Id),
    FirstMedia = firstMedia,
    GallerySize = group.Media.Count,
    FirstMembers = firstMembers
};
```

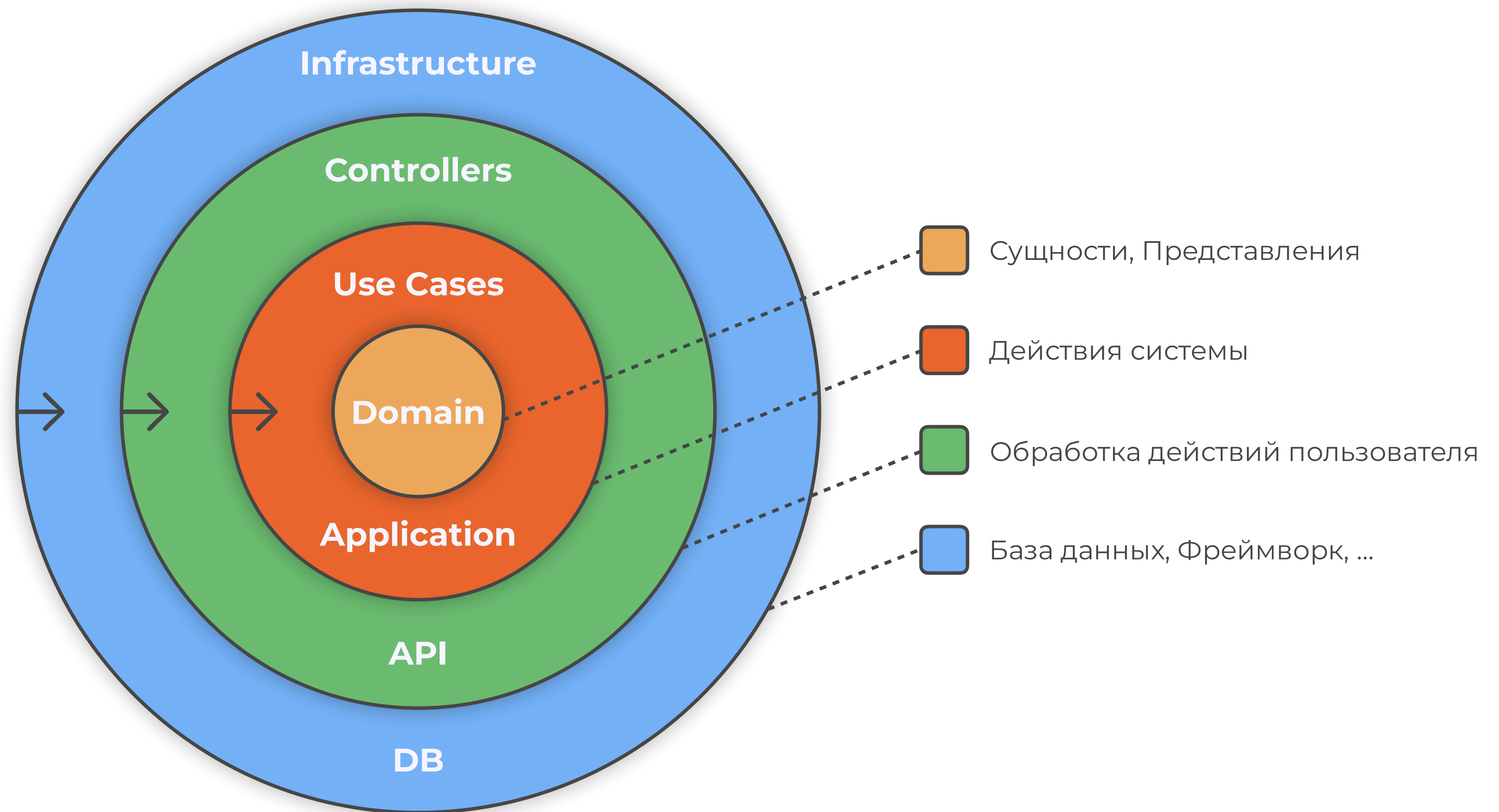
Usage example

```
public class GroupInfoResponse
{
    public int Id { get; set; }
    public int DataSize { get; set; }
    public string Name { get; set; } = string.Empty;
    public byte[] Image { get; set; } = Array.Empty<byte>();
    public int GallerySize { get; set; }
    public UserView? Owner { get; set; } = new();
    public IEnumerable<UserView>? FirstMembers { get; set; }
    public IEnumerable<MediaView>? FirstMedia { get; set; }
}
```

Response Model

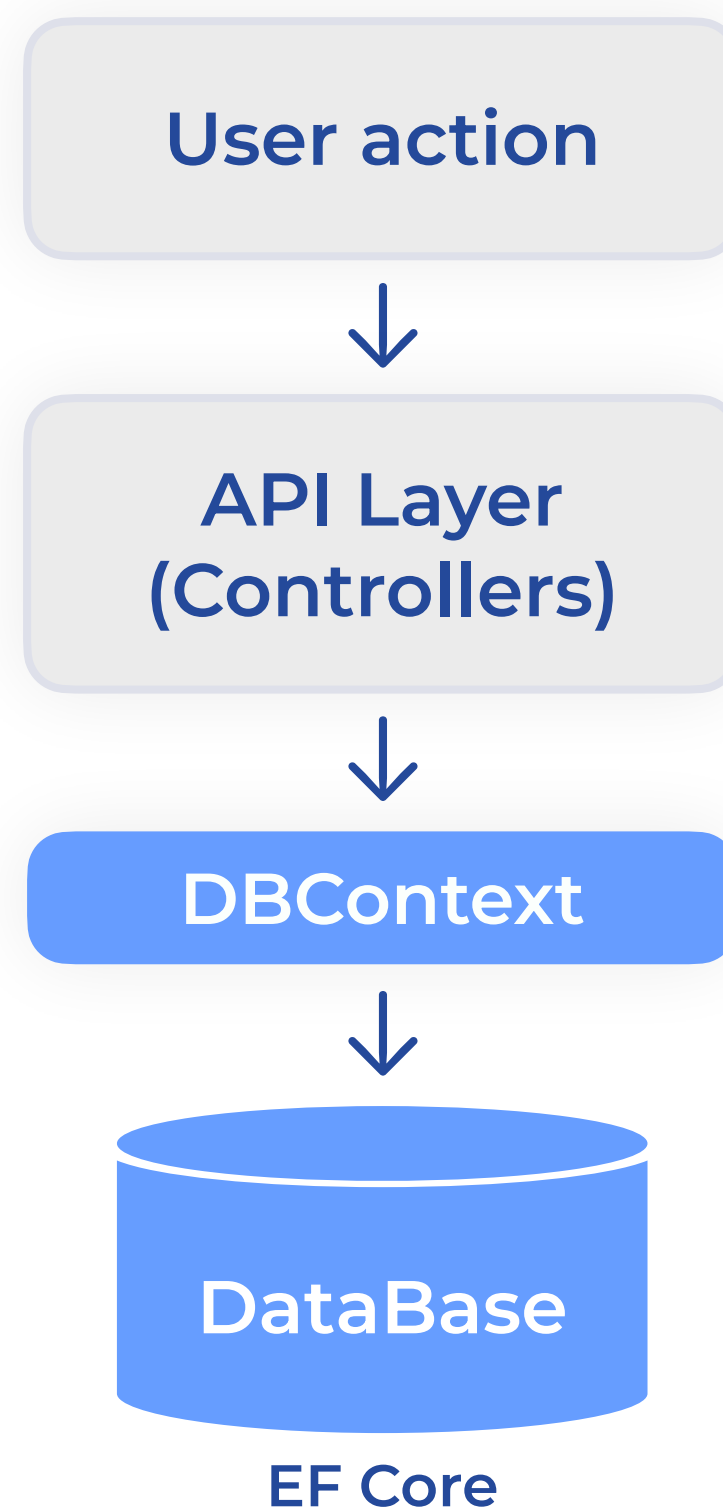
Архитектура приложения

Clean
Architecture

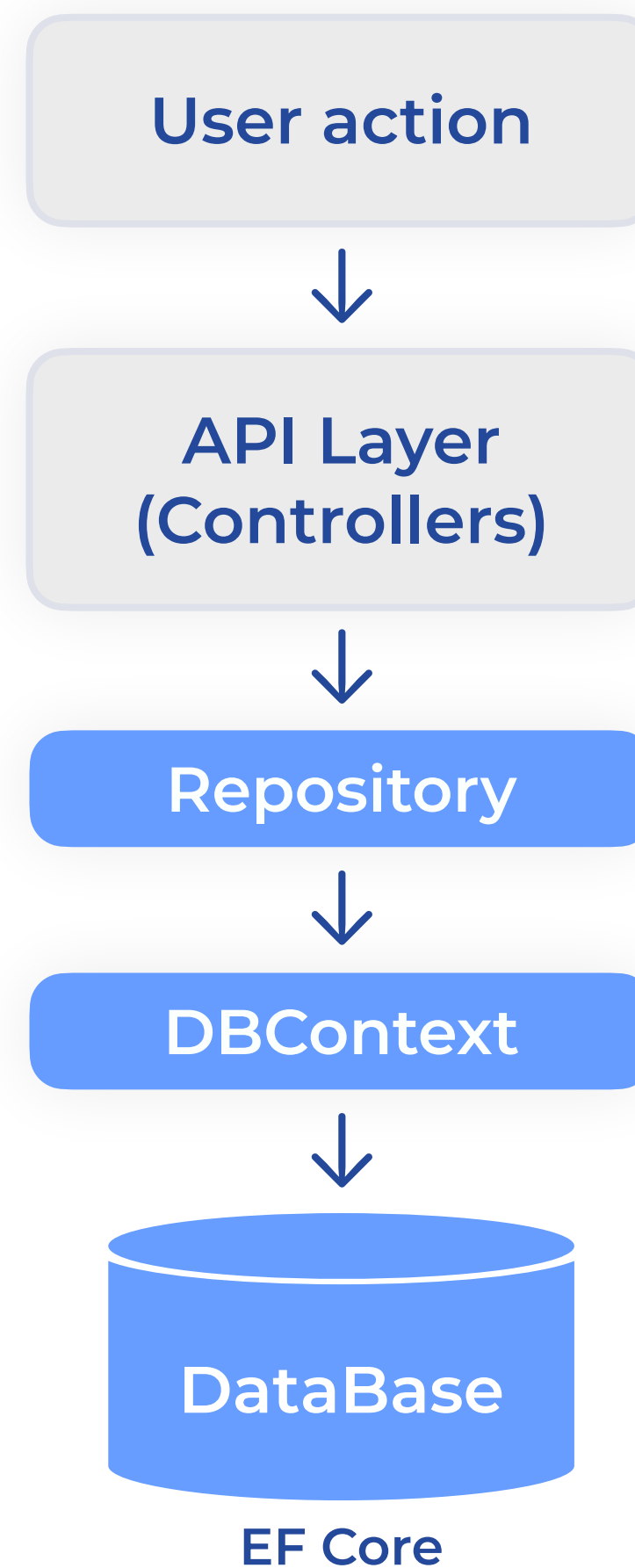


Repository Pattern

No Repository



With Repository



Example

```
[HttpPost(template: "create")]
28 usages 55 tests OK * Danila Rakhmanov
public async Task<ActionResult<int>> CreateGroup(CreateGroupRequest request)
{
    var owner:User? = _userRepository.GetUserByIdAsync(request.OwnerId).Result;
    if (owner == null) return BadRequest(error: "User not found");

    return Ok(await _groupRepository.CreateGroupAsync(owner, request));
}
```

GroupsController

```
public async Task<int> CreateGroupAsync(User owner, CreateGroupRequest request)
{
    var group = new Group
    {
        OwnerId = owner.Id,
        Name = request.GroupName,
        Image = Array.Empty<byte>()
    };

    group.Users.Add(owner);
    _context.Groups.Add(group);
    await _context.SaveChangesAsync();

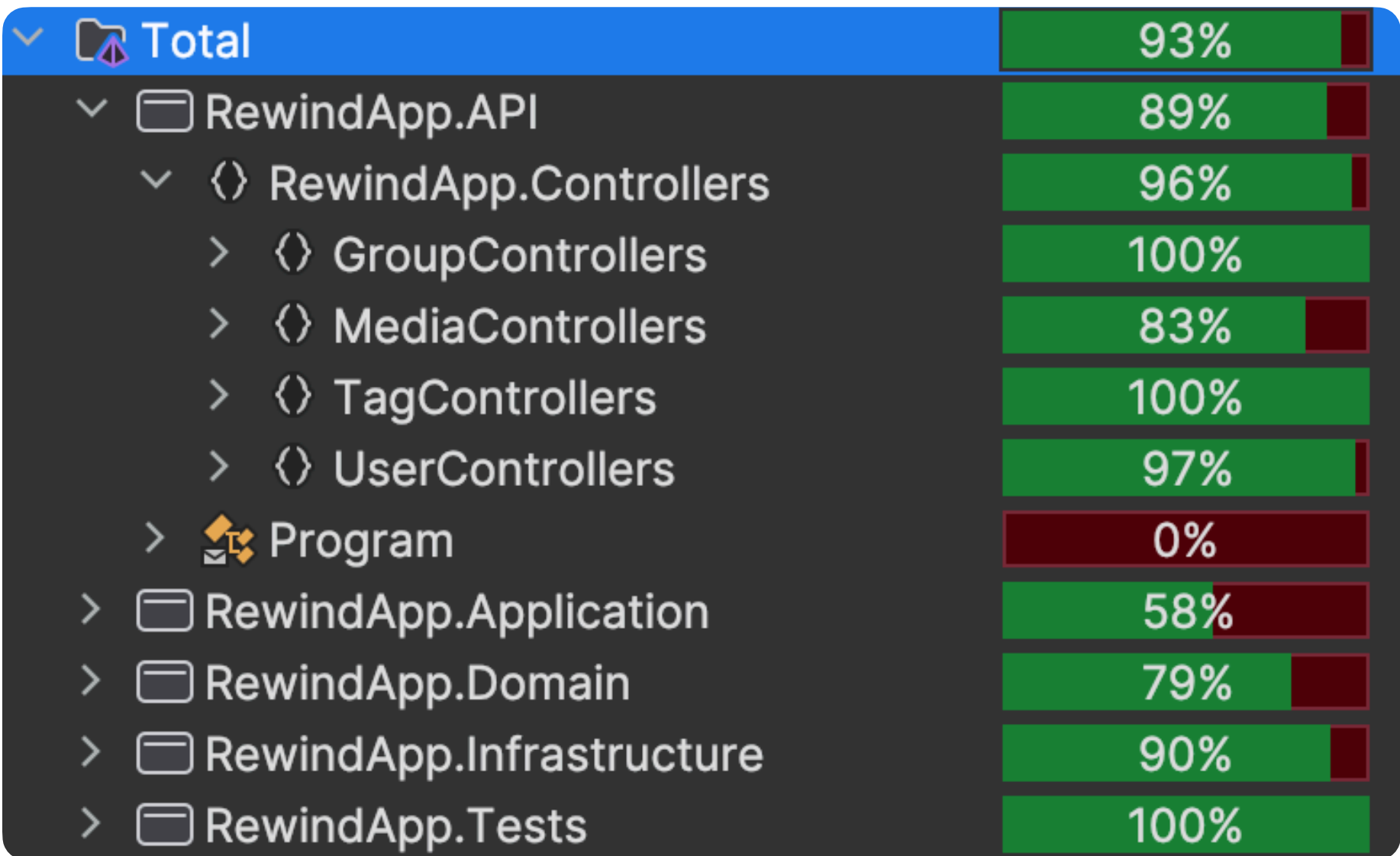
    return group.Id;
}
```

GroupRepository

UnitTests



Runned 79 tests



Tests coverage

Результаты

При разработке проекта выполнены следующие задачи:

- Проведён анализ аналогов
- Спроектирована структура БД
- Реализована выбранная архитектура серверной части приложения
- Реализован Repository Pattern
- Для сбора нужных данных использовались представления
- Проведено тестирование работы приложения при помощи UnitTest'ов

Дальнейшее развитие

- Реализовать возможность загрузки видео в галерею группы
- Разработать телеграмм-бота, имитирующего работу приложения
- Реализовать подробную фильтрацию медиа-объектов
- Реализовать масштабирование БД

Демонстрация

Спасибо за внимание!

Рахманов Данила Дмитриевич | ddrakhmanov@edu.hse.ru

Москва 2024