

Министерство образования и науки РФ  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии

## **High level design для сайта-блога с постами и комментариями**

Выполнили студенты гр. 5130904/00101

Прудников Д.А.  
Кривенко М.Д.  
Баталова Е.А.  
Федоров Г.Ю.

Руководитель

Маслаков А.П.

Санкт-Петербург  
2023

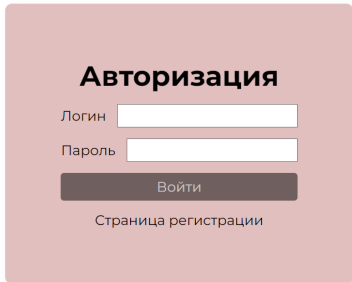
# Table of Contents

1. Макет дизайна frontend - части	4
1.1. Основные страницы	4
1.2. Всплывающие окна	6
1.3. Выпадающие списки	8
2. Стэк технологий	10
3. Общая схема	11
4. Схема классов front	12
5. Схема классов back	13
6. Схемы бд	14
6.1. User	14
6.2. Post	14
6.3. Comment	15
6.4. Notification	15
6.5. PostToUserLike	16
6.6. UserToSubscribedUser	16
7. API	17
7.1. Security	17
7.1.1. Авторизация	17
7.1.2. Регистрация	17
7.2. Профиль пользователя	18
7.2.1. Получение своего профиля	18
7.2.2. Получение чужого профиля	18
7.2.3. Получение профилей по логину	19
7.2.4. Изменение своего профиля	20
7.3. Пост	20
7.3.1. Создание поста	20
7.3.2. Получение всех своих постов	21
7.3.3. Получение всех постов пользователя	22
7.3.4. Получение постов фида новостей	23
7.3.5. Получение поста	23
7.3.6. Изменение поста	24
7.3.7. Удаление поста	25
7.4. Комментарий	25
7.4.1. Получение комментариев для поста	25
7.4.2. Создание комментария	26
7.5. Подписка	26
7.5.1. Подписка/отписка на/от пользователя	26
7.6. Отметка "Нравится"	27

7.6.1. Добавление/удаление отметки "Нравится" .....	27
7.7. Уведомление .....	28
7.7.1. Получение своих уведомлений .....	28

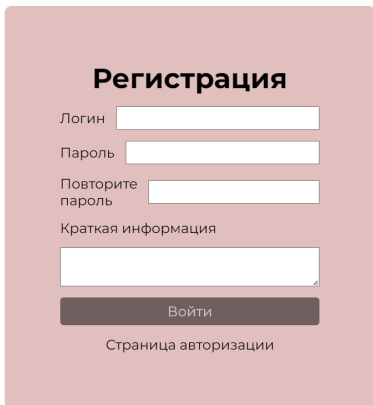
# Chapter 1. Макет дизайна frontend - части

## 1.1. Основные страницы



The login form is centered on a light gray background. It features a pinkish-red rounded rectangle with a subtle drop shadow. Inside, the title "Авторизация" is in bold black text. Below it are two white input fields for "Логин" and "Пароль". A dark gray button with the text "Войти" is positioned below the password field. At the bottom, a link "Страница регистрации" is displayed in a smaller, lighter font.

Figure 1. Страница логина



The registration form is centered on a light gray background. It features a pinkish-red rounded rectangle with a subtle drop shadow. Inside, the title "Регистрация" is in bold black text. Below it are three white input fields for "Логин", "Пароль", and "Повторите пароль". A larger white input field for "Краткая информация" is located below the password fields. A dark gray button with the text "Войти" is positioned below the "Краткая информация" field. At the bottom, a link "Страница авторизации" is displayed in a smaller, lighter font.

Figure 2. Страница регистрации

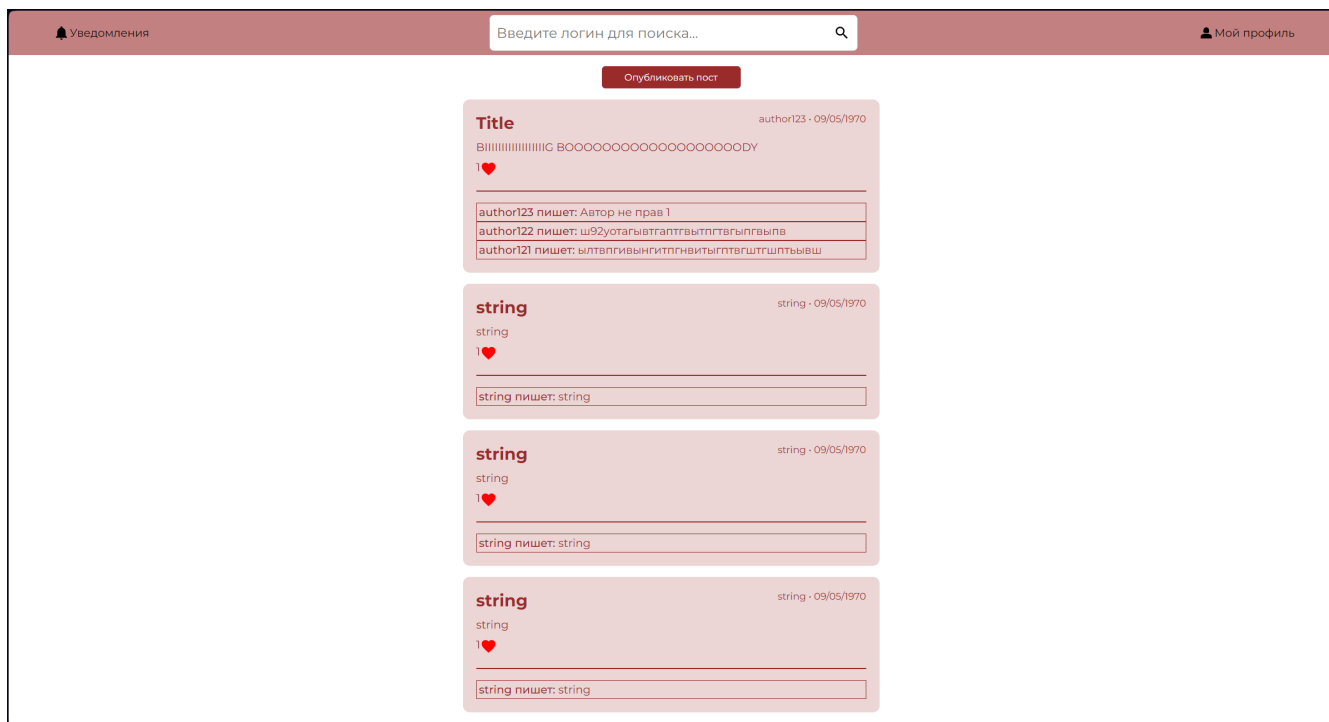


Figure 3. Страница ньюс фида

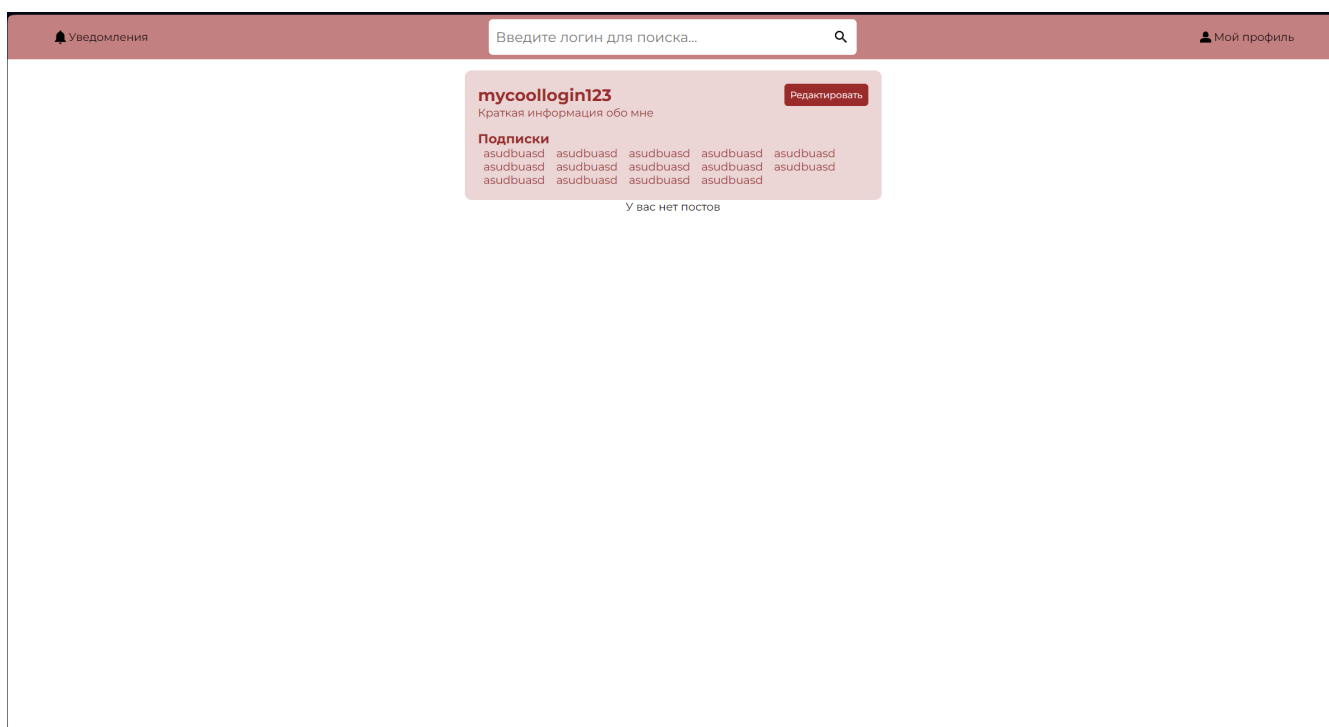


Figure 4. Страница собственного профиля

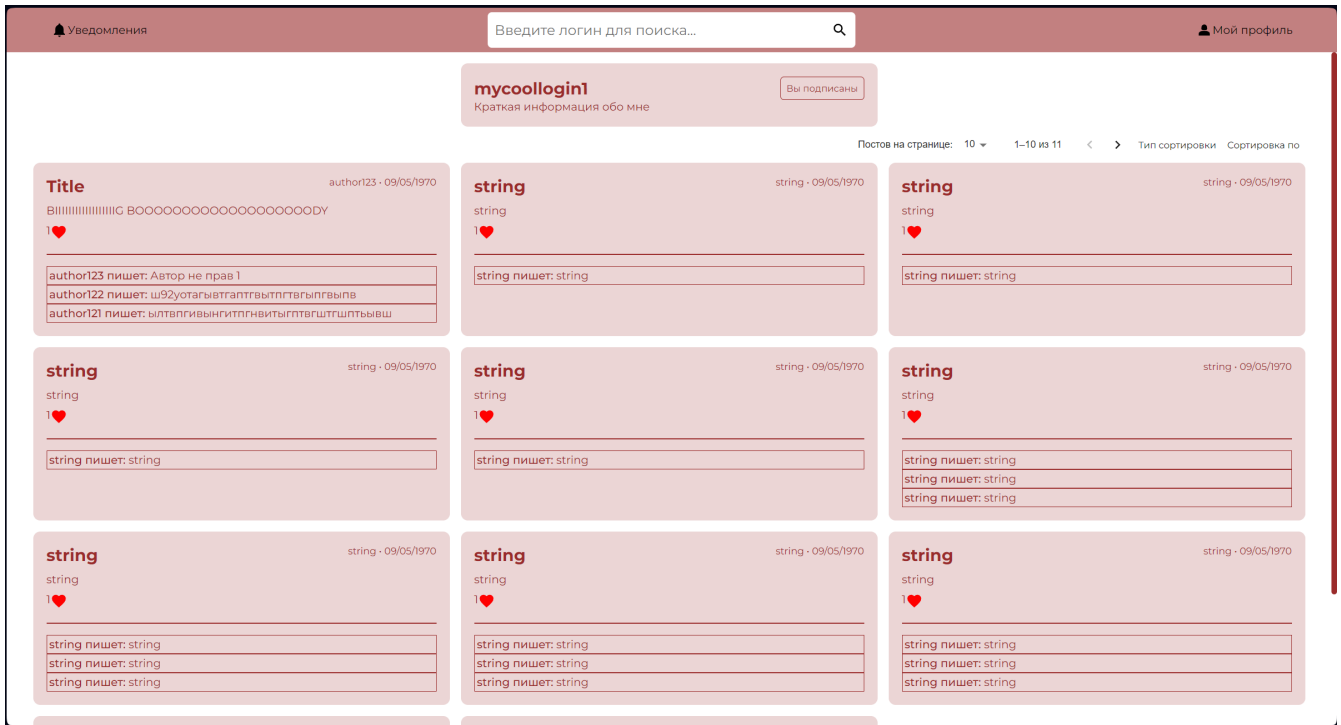


Figure 5. Страница чужого профиля

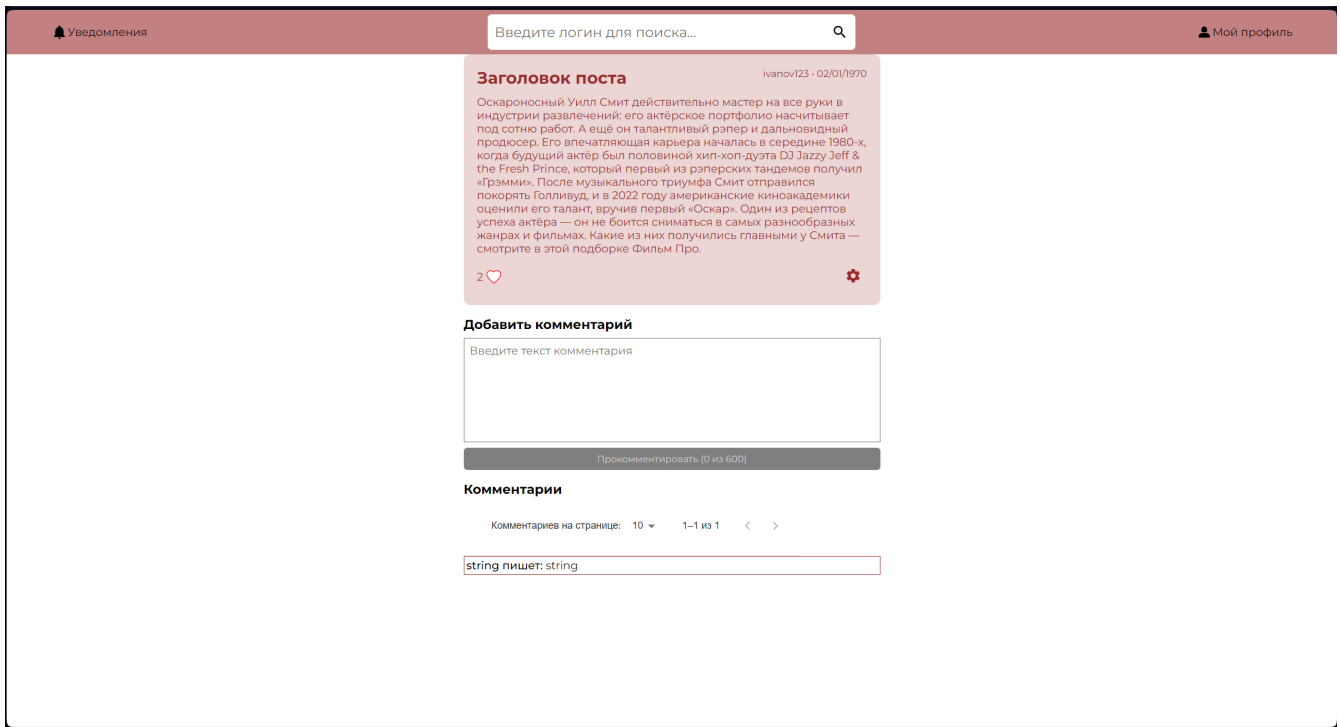


Figure 6. Страница поста

## 1.2. Всплывающие окна

## Создание нового поста

Заголовок

Введите заголовок...

Тело

Введите тело поста...

Опубликовать

Заккрыть

Figure 7. Создание нового поста

## Редактирование данных поста

Заголовок

Заголовок поста

Тело

Оскароносный Уилл Смит действительно мастер на все руки в индустрии развлечений: его актёрское портфолио насчитывает под сотню работ. А ещё он талантливый рэпер и

Сохранить

Заккрыть

Figure 8. Изменение поста

# Обновление информации профиля

Краткая информация

Краткая информация обо мне

Пароль

Введите пароль...

Повторите пароль

Введите пароль...

Сохранить

Заккрыть

Figure 9. Изменение информации в профиле

## 1.3. Выпадающие списки

Вас отметили. [Перейти](#)

Вас отметили. [Перейти](#)

Вас отметили. [Перейти](#)

Figure 10. Уведомления

asdasd



login1 asoifdjisafisauifuisanuifnuisnafuis

login2 asoifdjisafisauifuisanuifnuisnafuissadsads

login2 asoifdjisafisauifuisanuifnuisnafuissadsads

login2 asoifdjisafisauifuisanuifnuisnafuissadsads

login2 asoifdjisafisauifuisanuifnuisnafuissadsads

Figure 11. Поиск



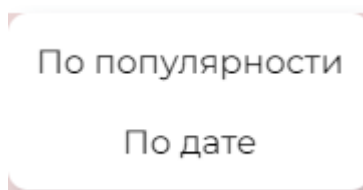


Figure 12. Тип сортировки

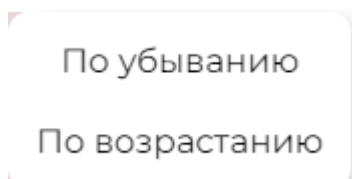


Figure 13. Порядок сортировки

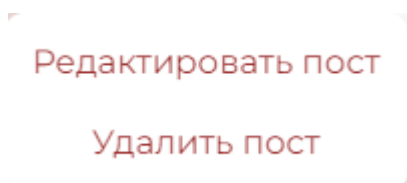


Figure 14. Функции поста (удалить/редактировать)

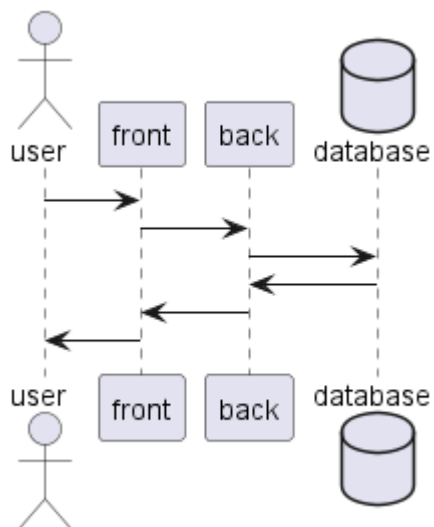
# Chapter 2. Стэк технологий

Front: React, Vite, MaterialUI, Typescript, axios, dayjs, react-query

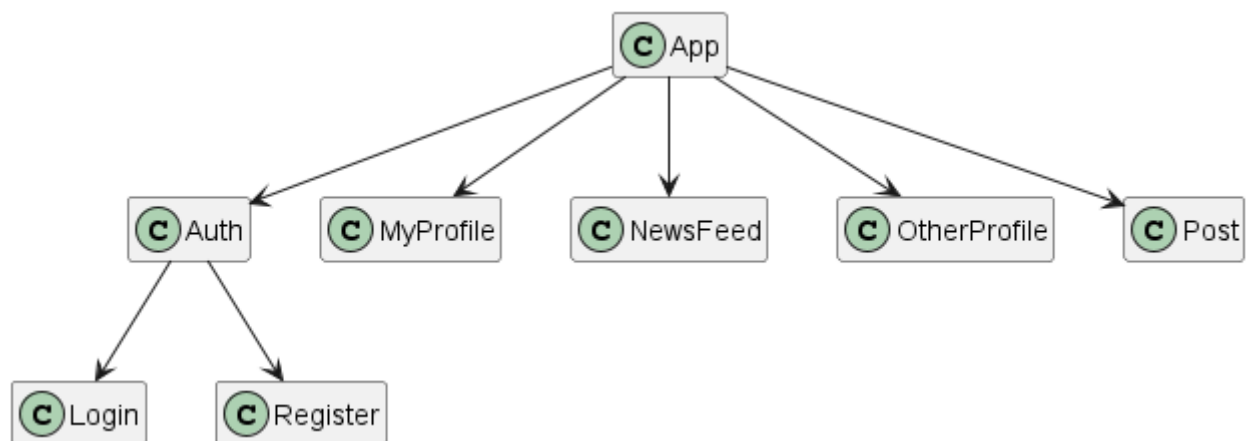
Back: Java, Spring Data/Security/MVC/Core/Boot, Maven

Остальное: Postgres, Docker, Nginx, Github, Github Projects

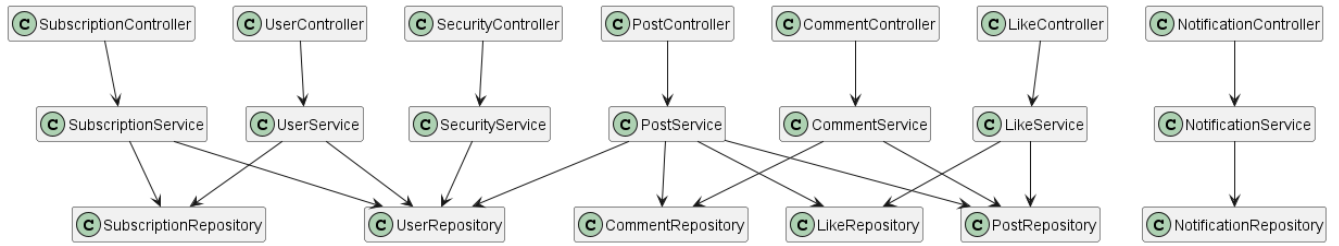
## Chapter 3. Общая схема



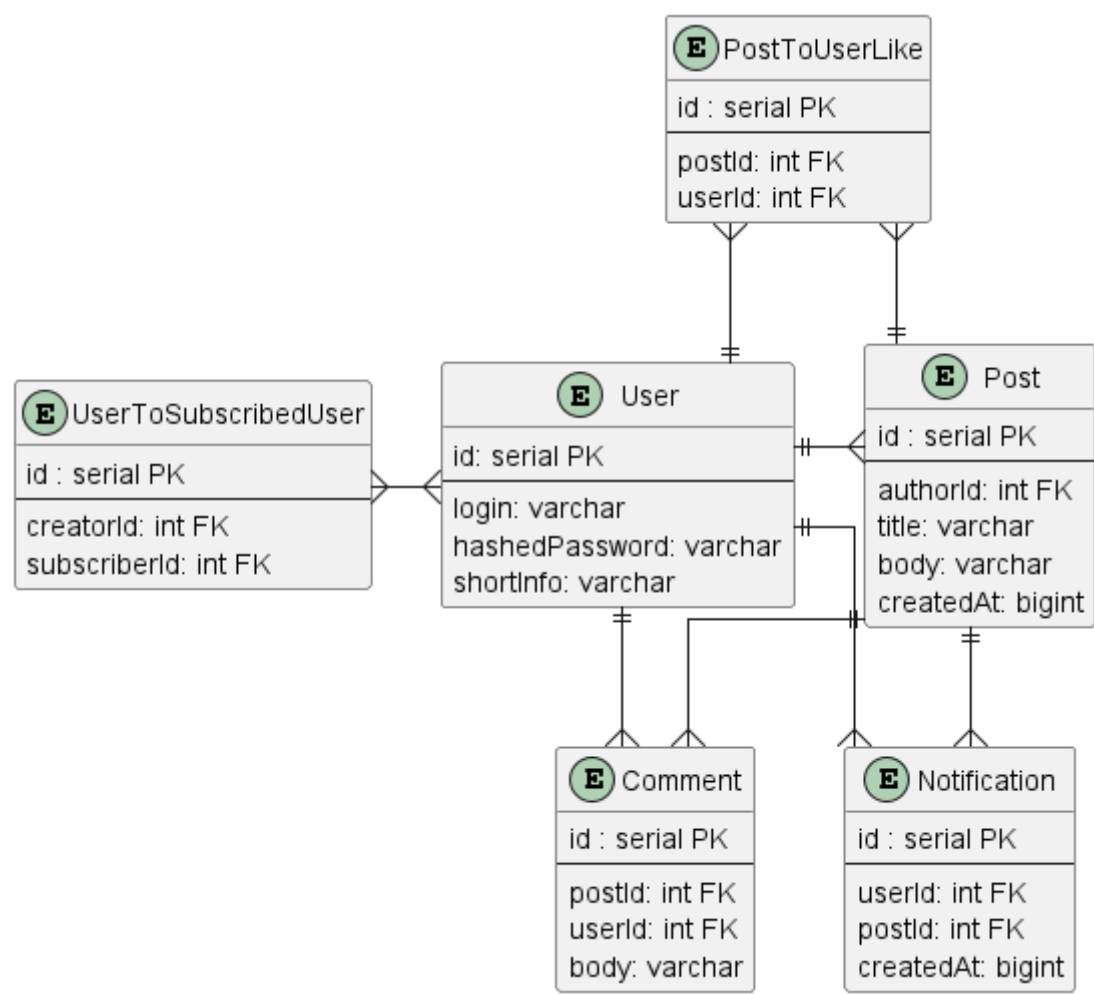
## Chapter 4. Схема классов front



## Chapter 5. Схема классов back



# Chapter 6. Схемы бд



## 6.1. User

Таблица профилей пользователей. Содержит логин, хешированный пароль и краткую информацию пользователя

Название поля	Тип	Ограничения
id	serial	PK - unique, not null
login	varchar	unique, not null, длина не меньше 8 и не больше 50 символов
hashedPassword	varchar	not null
shortInfo	varchar	not null, длина не меньше 1 символа и не больше 250 символов

## 6.2. Post

Таблица постов. Содержит ссылку на автора поста, название, тело поста и таймстемп

создания поста в utc миллисекундах

Название поля	Тип	Ограничения	Комментарий
id	serial	PK - unique, not null	-
authorId	int	FK, not null	-
title	varchar	not null, длина не меньше 1 символа и не больше 120 символов	-
body	varchar	not null, длина не меньше 1 символа и не больше 600 символов	-
createdAt	bigint	not null	таймстемп utc в миллисекундах

## 6.3. Comment

Таблица комментариев - содержит текст комментария, ссылку на пост, под которым оставлен комментарий, и ссылку на автора, который оставил комментарий

Название поля	Тип	Ограничения
id	serial	PK - unique, not null
userId	int	FK, not null
postId	int	FK, not null
body	varchar	not null, длина не меньше 1 символа и не больше 600 символов

## 6.4. Notification

Таблица уведомлений. Содержит ссылку на пользователя, которому предназначается уведомление, ссылку на пост, который вызвал создание уведомления, и таймстемп создания уведомления в utc миллисекундах. Уведомления, для которых  $createdAt + 30 \text{ дней} < \text{текущий utc таймстемп}$ , удаляются.

Название поля	Тип	Ограничения	Комментарий
id	serial	PK - unique, not null	-
userId	int	FK, not null	-
postId	int	FK, not null	-
createdAt	bigint	not null	таймстемп utc в миллисекундах

## 6.5. PostToUserLike

Таблица отметок "нравится". Содержит ссылку на пост, на который поставлена отметка, и ссылку на пользователя, поставившего отметку. Содержит уникальный индекс на {userId, postId}

Название поля	Тип	Ограничения
id	serial	PK - unique, not null
userId	int	FK, not null
postId	int	FK, not null

## 6.6. UserToSubscribedUser

Таблица подписок. Содержит ссылку на пользователя, на которого подписались, и ссылку на пользователя, который подписался. Содержит уникальный индекс {creatorId, subscriberId}

Название поля	Тип	Ограничения	Комментарий
id	serial	PK - unique, not null	-
creatorId	int	FK, not null	идентификатор пользователя, на которого подписаны
subscriberId	int	FK, not null	идентификатор пользователя, который подписан



# Chapter 7. API

## 7.1. Security

### 7.1.1. Авторизация

POST /api/v1/security/auth

#### Входные данные

```
{
  "login": "string",
  "password": "string"
}
```

#### Выходные данные

##### Тело

```
{
  "token": "string"
}
```

#### Коды ошибок

Код	Причины
422	ошибка валидации
400	логин не найден, пароль не совпал

### 7.1.2. Регистрация

POST /api/v1/security/register

#### Входные данные

```
{
  "login": "string",
  "password": "string",
  "shortInfo": "string"
}
```

#### Выходные данные

## Тело

```
{
  "token": "string"
}
```

## Коды ошибок

Код	Причины
422	ошибка валидации
400	логин не уникален

# 7.2. Профиль пользователя

## 7.2.1. Получение своего профиля

GET /api/v1/user/me

## Входные данные

-

## Выходные данные

## Тело

```
{
  "login": "string",
  "shortInfo": "string",
  "subscriptions": [
    {
      "id": 1,
      "login": "string"
    }
  ]
}
```

## Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован

## 7.2.2. Получение чужого профиля

GET /api/v1/user/id/{id}

где {id} - идентификатор пользователя

#### Входные данные

-

#### Выходные данные

##### Тело

```
{
  "id": 1,
  "login": "string",
  "shortInfo": "string",
  "subscribed": true
}
```

#### Коды ошибок

Код	Причины
400	идентификатор запрашиваемого пользователя совпадает с идентификатором запрашивающего
401	запрашивающий пользователь неавторизован
404	пользователь не найден

### 7.2.3. Получение профилей по логину

GET /api/v1/user/login/{login}

где {login} - логин или часть логина пользователя. В ответ не включается профиль запрашивающего пользователя

#### Входные данные

-

#### Выходные данные

##### Тело

```
[
  {
    "id": 1,
    "login": "string",
    "shortInfo": "string",
    "subscribed": true
  }
]
```

```
}  
]
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован

### 7.2.4. Изменение своего профиля

PUT /api/v1/user

#### Входные данные

```
{  
  "shortInfo": "string",  
  "password": "string"  
}
```

#### Выходные данные

Тело

-

#### Коды ошибок

Код	Причины
422	ошибка валидации
401	запрашивающий пользователь неавторизован

## 7.3. Пост

### 7.3.1. Создание поста

POST /api/v1/post

#### Входные данные

```
{  
  "title": "string",  
  "body": "string"  
}
```

## Выходные данные

### Тело

-

### Коды ошибок

Код	Причины
422	ошибка валидации
401	запрашивающий пользователь неавторизован

## 7.3.2. Получение всех своих постов

GET /api/v1/post/filter/mine?page={page}&size={size}&sort={sort}

где {page}, {size}, {sort} - параметры пагинации

### Входные данные

-

### Выходные данные

#### Тело

```
[
  {
    "id": 1,
    "title": "string",
    "body": "string",
    "authorId": 1,
    "authorLogin": "string",
    "createdAt": 1111111111,
    "likeCounter": 1,
    "hitLike": true,
    "firstComments": [
      {
        "id": 1,
        "authorLogin": "string",
        "body": "string"
      }
    ]
  }
]
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован

### 7.3.3. Получение всех постов пользователя

GET /api/v1/post/user/{id}?page={page}&size={size}&sort={sort}

где {page}, {size}, {sort} - параметры пагинации, {id} - идентификатор пользователя

#### Входные данные

-

#### Выходные данные

#### Тело

```
[
  {
    "id": 1,
    "title": "string",
    "body": "string",
    "authorId": 1,
    "authorLogin": "string",
    "createdAt": 111111111,
    "likeCounter": 1,
    "hitLike": true
    "firstComments": [
      {
        "id": 1,
        "authorLogin": "string",
        "body": "string"
      }
    ]
  }
]
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован
404	пользователь не найден

### 7.3.4. Получение постов фида новостей

GET /api/v1/post/filter/feed

#### Входные данные

-

#### Выходные данные

##### Тело

```
[
  {
    "id": 1,
    "title": "string",
    "body": "string",
    "authorId": 1,
    "authorLogin": "string",
    "createdAt": 1111111111,
    "likeCounter": 1,
    "hitLike": true
    "firstComments": [
      {
        "id": 1,
        "authorLogin": "string",
        "body": "string"
      }
    ]
  }
]
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован

### 7.3.5. Получение поста

GET /api/v1/post/{id}

где {id} - идентификатор поста

#### Входные данные

-

## Выходные данные

### Тело

```
{
  "title": "string",
  "body": "string",
  "createdAt": 1111111111,
  "authorId": 1,
  "authorLogin": "string",
  "likeCounter": 1,
  "hitLike": true,
  "isAuthor": true
}
```

### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован
404	пост не найден

## 7.3.6. Изменение поста

PUT /api/v1/post/{id}

где {id} - идентификатор поста

### Входные данные

```
{
  "title": "string",
  "body": "string"
}
```

## Выходные данные

### Тело

-

### Коды ошибок

Код	Причины
422	ошибка валидации
401	запрашивающий пользователь неавторизован



Код	Причины
403	попытка изменить пост от пользователя, не являющегося автором поста
404	пост не найден

### 7.3.7. Удаление поста

**DELETE** /api/v1/post/{id}

где {id} - идентификатор поста

#### Входные данные

-

#### Выходные данные

##### Тело

-

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован
403	попытка удалить пост от пользователя, не являющегося автором поста
404	пост не найден

## 7.4. Комментарий

### 7.4.1. Получение комментариев для поста

**GET** /api/v1/post/{id}/comment?page={page}&size={size}&sort={sort}

где {page}, {size}, {sort} - параметры пагинации, {id} - идентификатор поста

#### Входные данные

-

#### Выходные данные

##### Тело

[

```
{
  "id": 1,
  "authorLogin": "string",
  "body": "string"
}
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован
404	пост не найден

### 7.4.2. Создание комментария

POST /api/v1/post/{id}/comment

где {id} - идентификатор поста

#### Входные данные

```
{
  "body": "string"
}
```

#### Выходные данные

Тело

-

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован
404	пост не найден

## 7.5. Подписка

### 7.5.1. Подписка/отписка на/от пользователя

POST /api/v1/subscription/{userId}

где userId - идентификатор пользователя, на которого нужно подписаться/от которого

нужно отписаться

### Входные данные

-

### Выходные данные

#### Тело

```
{  
  "subscribed": true  
}
```

#### Коды ошибок

Код	Причины
400	попытка подписи на самого себя
401	запрашивающий пользователь неавторизован
404	пользователь не найден

## 7.6. Отметка "Нравится"

### 7.6.1. Добавление/удаление отметки "Нравится"

POST /api/v1/post/{id}

где {id} - идентификатор поста

### Входные данные

-

### Выходные данные

#### Тело

```
{  
  "hitLike": true  
}
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован
404	пост не найден

## 7.7. Уведомление

### 7.7.1. Получение своих уведомлений

GET /api/v1/notification

#### Входные данные

-

#### Выходные данные

##### Тело

```
[
  {
    "id": 1,
    "postId": 1
  }
]
```

#### Коды ошибок

Код	Причины
401	запрашивающий пользователь неавторизован