

**Даты редактирований:**

1. 24.11.2022 .

2. 01.12.2022 .

3. 12.12.2022

4. 14.12.2022

5. 15.12.2022

**Работу выполнил:**

М.А.Меньщиков

**Курирующий преподаватель:**

К.А.Кулаков

[PDF](#)

## **ПЛАН ТЕСТИРОВАНИЯ**

## Содержание

<b>1. Объект тестирования</b>	<b>4</b>
1.1. Описание объекта тестирования	4
1.2. Рамки тестирования	4
1.3. Перечень функциональностей объекта тестирования (ФТ)	4
<b>2. Стратегия тестирования</b>	<b>6</b>
2.1. Структура и связи объекта тестирования	6
2.1.1. Архитектура	6
2.1.2. Интерфейс пользователя	16
2.2. Стратегия блочного тестирования (БТ)	30
2.3. Стратегия интеграционного тестирования (ИТ)	31
2.4. Стратегия аттестационного тестирования (АТ)	31
2.5. Стратегии специальных видов тестирования	31
2.5.1. Нагрузочное тестирование (НТ)	31
2.5.2. Тестирование безопасности (ТБ)	32
2.6. Переход между этапами тестирования	32
2.7. Критерии прохождения тестирования	32
2.8. Условия возобновления и приостановки выполнения тестов	32
<b>3. Детальный план тестирования (ПТ)</b>	<b>34</b>
3.1. Блочное тестирование (ПТ.БТ)	34
3.2. Интеграционное тестирование (ПТ.ИТ)	42
3.3. Аттестационное тестирование (ПТ.АТ)	53
3.4. Нагрузочное тестирование (ПТ.НТ)	55
3.5. Тестирование безопасности (ПТ.ТБ)	56
<b>4. Трассируемость функциональных требований в тестах (ТФТ)</b>	<b>58</b>
4.1. Блочное тестирование (ТФТ.БТ)	58
4.2. Интеграционное тестирование (ТФТ.ИТ)	59

4.3. Аттестационное тестирование (ТФТ.АТ)	60
4.4. Нагрузочное тестирование (ТФТ.НТ)	60
4.5. Тестирование безопасности (ТФТ.ТБ)	61
<b>5. Журнал тестирования (ЖТ)</b>	<b>62</b>
5.1. Блочное тестирование (ЖТ.БТ)	62
5.2. Интеграционное тестирование (ЖТ.ИТ)	63
5.3. Аттестационное тестирование (ЖТ.АТ)	64
5.4. Нагрузочное тестирование (ЖТ.НТ)	64
5.5. Тестирование безопасности (ЖТ.ТБ)	65
<b>6. Журнал найденных ошибок (ЖО)</b>	<b>66</b>
6.1. Блочное тестирование (ЖО.БТ)	66
6.2. Интеграционное тестирование (ЖО.ИТ)	67
6.3. Аттестационное тестирование (ЖО.АТ)	67
6.4. Нагрузочное тестирование (ЖО.НТ)	67
6.5. Тестирование безопасности (ЖО.ТБ)	67
<b>7. Результаты</b>	<b>69</b>
<b>8. Приложения</b>	<b>70</b>
8.1. Косвенные входные данные авторизация/регистрация	70
8.2. Косвенные входные данные Цифровой профиль	70
8.3. Косвенные входные данные Чат-группы	70
8.4. Косвенные входные данные Личный профиль	71
8.5. Косвенные входные данные Сессионные ключи	71

# 1. Объект тестирования

## 1.1. Описание объекта тестирования

Система представляет из себя Web-сервис для поиска людей со схожими интересами на основании списка личных достижений (портфолио). Пользователь регистрируется в системе, вносит информацию о своих профессиональных достижениях и алгоритм автоматически проанализирует и определит специфику его интересов. В конечном итоге система добавит пользователя в тематическую чат-группу к другим пользователям, близких по списку профессиональных навыков для возможности обмена сообщениями.

## 1.2. Рамки тестирования

По плану будет протестирован функционал, связанный с регистрацией/авторизацией пользователя в системе, алгоритм кластеризации пользователей на основании их текстовых описаний и механизм распределения пользователей по чат-группам.

## 1.3. Перечень функциональностей объекта тестирования (ФТ)

Тестируемая часть Web-системы обладает следующими функциональными требованиями:

1. Регистрация и вход (**ФТ1**)
  - 1.1. Прохождение этапа регистрации для взаимодействия с системой; (**ФТ1.1**)
  - 1.2. Авторизация в системе для продолжения взаимодействия. (**ФТ1.2**)
2. Профиль пользователя (**ФТ2**)
  - 2.1. Просмотр и изменение информации личного профиля: имени, текстового описания; (**ФТ2.1**)
  - 2.2. Просмотр профилей других пользователей. (**ФТ2.2**)
3. Чат-группы (**ФТ3**)
  - 3.1. Кластеризация пользователей по текстовому описанию их профиля; (**ФТ3.1**)
  - 3.2. Автоматическое создание тематических чат-групп; (**ФТ3.2**)
  - 3.3. Автоматическое распределение пользователей по тематическим чат-группам; (**ФТ3.3**)
  - 3.4. Просмотр содержания тех чат-групп, в которые пользователь был автоматически добавлен; (**ФТ3.4**)
  - 3.5. Просмотр списка пользователей, состоящих в данной чат-группе; (**ФТ3.5**)

3.6. Возможность перехода в профиль пользователя, находящегося в данной чат-группе. **(ФТ3.6)**

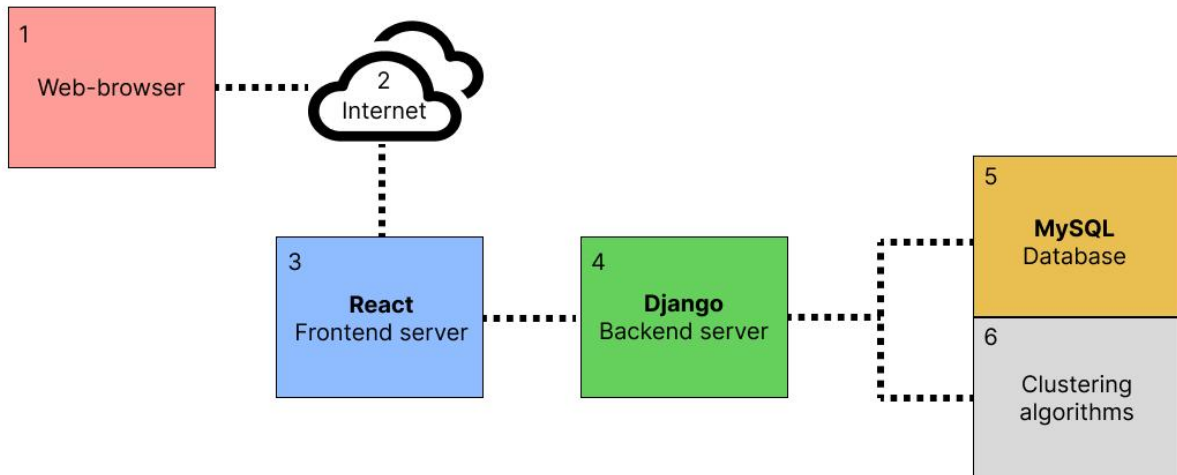
На функциональные требования накладываются следующие функциональные ограничения:

1. Пользователь идентифицируется по логину и паролю; **(ФТ1)**
2. В пароле и логине пользователя не должно быть пробельных и специальных символов; **(ФТ1)**
3. Длина логина должна быть не менее 4 символов и не более 18 символов; **(ФТ1)**
4. Длина пароля должна быть не менее 8 символов и не более 32 символов; **(ФТ1)**
5. Доступные символы для составления логина и пароля: [a-z0-9\_-]; **(ФТ1)**
6. Для просмотра списка доступных чат-групп необходима авторизация в системе; **(ФТ1)**
7. Для просмотра содержания доступных чат-групп необходима авторизация в системе; **(ФТ1)**
8. Для изменения данных своего профиля необходима авторизация в системе; **(ФТ1)**
9. Имя пользователя в его профиле не должно быть пустым; **(ФТ2)**
10. Доступные символы для создания имени пользователя в профиле: [a-яA-Яa-zA-Z0-9\_]; **(ФТ2)**
11. Размер текстового описания в профиле пользователя не должен превышать 512 символов; **(ФТ2)**
12. Чат-группа не должна быть пустой (количество пользователей в чат-группе не равно нулю); **(ФТ3)**
13. Пользователь может состоять только в одной чат-группе. **(ФТ3)**

## 2. Стратегия тестирования

### 2.1. Структура и связи объекта тестирования

#### 2.1.1. Архитектура



#### Основные модули:

1. **Web-browser** - система взаимодействия с конечным пользователем посредством отображения html-страниц и ожидания ответных реакций;
2. **Internet** - сеть передачи данных между клиентским и серверным приложениями;
3. **React Frontend server** - система формирования html-страниц для отправки клиентскому приложению и обработки событий от конечного пользователя;
4. **Django Backend server** - API для обращения к данным, необходимым при формировании html-страниц для конечного пользователя;
5. **MySQL Database** - хранилище данных, необходимых для работы исходной системы;
6. **Clustering algorithms** - логика системы, реализующая алгоритм кластеризации пользователей со схожими интересами на основании их текстового описания.

#### Взаимодействие:

- Связь [1 – 2 – 3]:
  - [– >] передача клиентских запросов/событий;
  - [ < – ] передача результата обработки клиентского запроса.
- Связь [3 – 4]:

- [ - > ] передача запроса на получение/внесения необходимых данных из/в БД (json);
- [ < - ] передача результата обработки запроса (json).
- Связь [4 – 5]:
  - [ - > ] Сохранение данных, необходимых для корректной работы системы;
  - [ < - ] Сериализация данных для обработки поступивших запросов.
- Связь [4 – 6]:
  - [ - > ] использование реализованных алгоритмов для обработки поступивших запросов.

### Структура кода:

- **Django Backend server:**

### Структура API-сервера:

- **manage.py** - скрипт используется для инициализации сервера, работы с базами данных и для запуска отладчика.
- **settings.py** - содержит в себе все настройки сервера. Здесь мы регистрируем модули, задаём размещение статичных файлов, настройки базы данных и так далее.
- **urls.py** - Функция path принимает на вход URL-адрес и ставит в соответствие функцию/класс из модуля views.py. URL соотношения хранятся в переменной urlpatterns, которая является списком функций path().
- **models.py** - содержит структуры таблиц из базы данных для взаимодействия с ними с помощью встроенной ORM. Таблицы БД указаны в пункте “MySQL Database”.
- **views.py** - тут размещена «логика» работы модуля. Отсюда идут запросы к данным из моделей (таблиц базы данных) и запуск реализованных алгоритмов их(данных) обработки.

Будут разрешены следующие пути/команды, по которым будут обрабатываться запросы (HTTP-запросы, содержание в body json-объект):

1. **/register** - получение запросов на регистрацию новых пользователей в системе.
  - a. **Входной запрос:**
    - i. **(str) login** - новый логин пользователя;
    - ii. **(str) password** - новый пароль пользователя.
  - b. **Выходной ответ:**

- i. В случае **успешной** регистрации возвращается json-файл с заполненным полем **token** (сессионный токен для дальнейшего взаимодействия с системой);
    - ii. В случае **провала** регистрации возвращается json-файл с заполненным полем **error** ( флаг о провале: True ).
  - c. **Привязанное представление:** RegisterAPIView(APIView).
- 2. **/auth** - получение запроса на авторизацию пользователя в системе.
  - a. **Входной запрос:**
    - i. **(str) login** - логин пользователя;
    - ii. **(str) password** - пароль пользователя.
  - b. **Выходной ответ:**
    - i. В случае **успешной** авторизации возвращается json-файл с заполненным полем **(str) token** (сессионный токен для дальнейшего взаимодействия с системой);
    - ii. В случае **провала** авторизации возвращается json-файл с заполненным полем **(bool) error** ( флаг о провале: True ).
  - c. **Привязанное представление:** AuthorizeAPIView(APIView).
- 3. **/change** - получение запроса на изменение публичной информации пользователя в системе: имени и/или текстового описания.
  - a. **Входной запрос:**
    - i. список json-объектов, каждый из которых имеет следующую структуру:
      - 1. **(str) type** - тип изменённого текстового поля (**name/description**);
      - 2. **(str) data** - новые данные.
    - ii. **(str) token** - сессионный токен авторизованного пользователя.
  - b. **Выходной ответ:**
    - i. **(bool) error** - успех операции (провал: True; успех: False).
  - c. **Привязанное представление:** ChangeDataAPIView(APIView).
- 4. **/groups** - получение запроса на формирования списка, доступных для данного пользователя, чат-групп.
  - a. **Входной запрос:**
    - i. **(str) token** - сессионный токен авторизованного пользователя.
  - b. **Выходной ответ:**
    - i. В случае **успеха** возвращается список json-объектов, каждый из которых имеет следующую структуру:
      - 1. **(int) group\_num** - номер чат-группы.
    - ii. В случае **провала** получения данных возвращается json-файл с заполненным полем **(bool) error** ( флаг о провале: True ).
  - c. **Привязанное представление:** AvailableGroupsAPIView(APIView)



5. **/chat** - получение запроса на формирования содержания чат-группы с переданным идентификаторов.
- a. **Входной запрос:**
    - i. **(int) group\_num** - номер чат-группы.
    - ii. **(str) token** - сессионный токен авторизованного пользователя.
  - b. **Выходной ответ:**
    - i. В случае **успеха** возвращается список json-объектов, каждый из которых имеет следующую структуру:
      - 1. **(str) login** - логин пользователя, состоящего в данной чат-группе;
      - 2. **(str) name** - имя пользователя, состоящего в данной чат-группе.
    - ii. В случае **провала** получения данных возвращается json-файл с заполненным полем **(bool) error** ( флаг о провале: True ).
  - c. **Привязанное представление:** ChatGroupAPIView(APIView).
6. **/profile** - получение запроса на формирование публичного содержания личного профиля пользователя с переданным логином.
- a. **Входной запрос:**
    - i. **(int) login** - логин пользователя, содержание публичных данных которого нужно получить.
    - ii. **(str) token** - сессионный токен авторизованного пользователя.
  - b. **Выходной ответ:**
    - i. В случае **успеха** возвращается json-объектов, каждый из которых имеет следующую структуру:
      - 1. **(str) description** - текстовое описание личного профиля данного пользователя;
      - 2. **(str) name** - имя данного пользователя.
    - ii. В случае **провала** получения публичных данных профиля возвращается json-файл с заполненным полем **(bool) error** ( флаг о провале: True ).
  - c. **Привязанное представление:** ProfileAPIView(APIView).

Функционал привязанных представлений обработчиков:

- 1. Класс **RegisterAPIView:**
  - a. Функция **post** - получение запроса на обработку по привязанному пути.
    - i. **Входные данные:**
      - 1. **(HttpRequest)** - запрос по протоколу http.
    - ii. **Выходные данные:**
      - 1. **(HttpResponse)** - ответ по протоколу http.

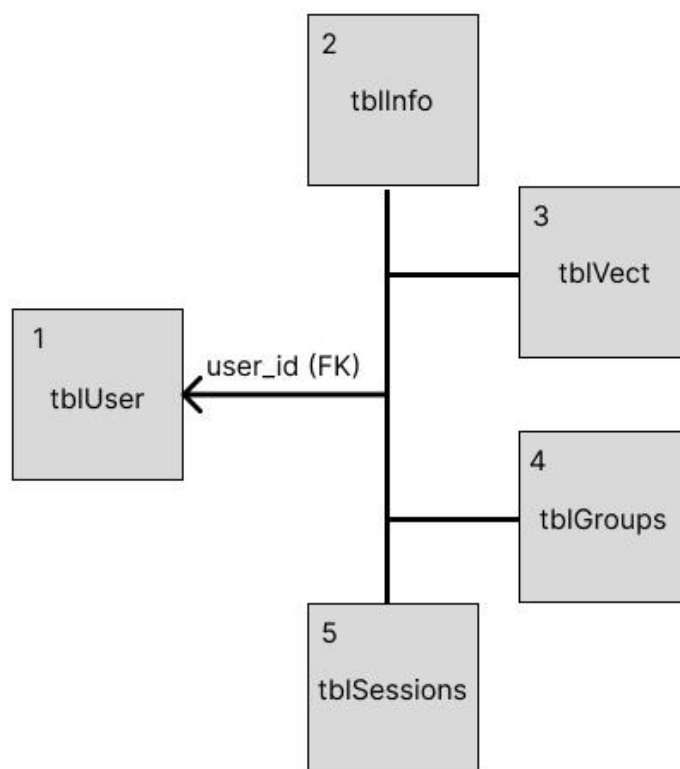
- b. Функция **validate\_reg** - проверка переданных данных для регистрации на наличие в БД.
    - i. **Входные данные:**
      - 1. (**str**) - логин;
      - 2. (**str**) - пароль.
    - ii. **Выходные данные:**
      - 1. (**bool**) - True, если валидация прошла успешно, иначе False.
  - c. Функция **add\_new\_user** - добавление информации о зарегистрированном пользователе в БД.
    - i. **Входные данные:**
      - 1. (**str**) - логин;
      - 2. (**str**) - пароль.
    - ii. **Выходные данные:**
      - 1. (**bool**) - True, если операция прошла успешно, иначе False.
2. Класс **AuthorizeAPIView:**
- a. Функция **post** - получение запроса на обработку по привязанному пути.
    - i. **Входные данные:**
      - 1. (**HttpRequest**) - запрос по протоколу http.
    - ii. **Выходные данные:**
      - 1. (**HttpResponse**) - ответ по протоколу http.
  - b. Функция **validate\_auth** - проверка переданных данных для авторизации на наличие в БД.
    - i. **Входные данные:**
      - 1. (**str**) - логин;
      - 2. (**str**) - пароль.
    - ii. **Выходные данные:**
      - 1. (**bool**) - True, если валидация прошла успешно, иначе False.
3. Класс **ChangeDataAPIView:**
- a. Функция **post** - получение запроса на обработку по привязанному пути.
    - i. **Входные данные:**
      - 1. (**HttpRequest**) - запрос по протоколу http.
    - ii. **Выходные данные:**
      - 1. (**HttpResponse**) - ответ по протоколу http.
  - b. Функция **make\_changes** - внесение изменений в публичную информацию пользователя по соответствующему идентификатору.
    - i. **Входные данные:**
      - 1. (**str**) - тип изменения;
      - 2. (**str**) - данные на замену;
      - 3. (**int**) - идентификатор пользователя.

- ii. **Выходные данные:**
      - 1. (**bool**) - True, если внесение изменений произведено успешно, иначе False.
  - c. Функция **recalc\_groups** - перераспределение пользователей по новым группам на основании изменённых данных.
    - i. **Входные данные:**
      - 1. Отсутствуют (идёт взаимодействие с БД).
    - ii. **Выходные данные:**
      - 1. (**bool**) - True, если перераспределение произведено успешно, иначе False.
- 4. Класс **AvailableGroupsAPIView:**
  - a. Функция **post** - получение запроса на обработку по привязанному пути.
    - i. **Входные данные:**
      - 1. (**HttpRequest**) - запрос по протоколу http.
    - ii. **Выходные данные:**
      - 1. (**HttpResponse**) - ответ по протоколу http.
  - b. Функция **prepare\_group\_list** - подготовка json-объекта со списком доступных чат-групп для пользователя с переданным идентификатором.
    - i. **Входные данные:**
      - 1. (**int**) - идентификатор пользователя.
    - ii. **Выходные данные:**
      - 1. (**list[int]**) - список идентификаторов чат-групп.
- 5. Класс **ChatGroupAPIView:**
  - a. Функция **post** - получение запроса на обработку по привязанному пути.
    - i. **Входные данные:**
      - 1. (**HttpRequest**) - запрос по протоколу http.
    - ii. **Выходные данные:**
      - 1. (**HttpResponse**) - ответ по протоколу http.
  - b. Функция **prepare\_chatgroup\_info** - подготовка json-объекта с информацией по чат-группе с переданным идентификатором.
    - i. **Входные данные:**
      - 1. (**int**) - идентификатор чат-группы.
    - ii. **Выходные данные:**
      - 1. (**dict[str, str]**) - логин пользователя, имя пользователя в случае **успеха**.
      - 2. (**bool**) - False в случае **провала** операции.
- 6. Класс **ProfileAPIView:**
  - a. Функция **post** - получение запроса на обработку по привязанному пути.
    - i. **Входные данные:**
      - 1. (**HttpRequest**) - запрос по протоколу http.

- ii. **Выходные данные:**
  - 1. (**HttpResponse**) - ответ по протоколу http.
- b. Функция **prepare\_profile\_info** - подготовка json-объекта с публичной информацией профиля пользователя с переданным логином.
  - i. **Входные данные:**
    - 1. (**str**) - логин пользователя.
  - ii. **Выходные данные:**
    - 1. (**tuple[str, str]**) - имя пользователя в системе и текстовое описание пользователя в случае **успеха**;
    - 2. (**bool**) - False в случае **провала** операции.
- 7. Функция **create\_session\_token** - создание сессионного токена для данного пользователя.
  - a. **Входные данные:**
    - i. (**int**) - идентификатор пользователя в БД.
  - b. **Выходные данные:**
    - i. (**str**) - сессионный токен в случае **успеха**;
    - ii. (**bool**) - False в случае **провала** создания токена.
- 8. Функция **validate\_token** - верификация сессионного токена.
  - a. **Входные данные:**
    - i. (**str**) - сессионный токен.
  - b. **Выходные данные:**
    - i. (**bool**) - True, если верификация пройдена, иначе False.

- **MySQL Database:**

База данных имеет следующую структуру таблиц:



1. **tblUser** - приватная информация пользователя в системе.

Поле	Тип	Ключ	Описание
user_id	int	PK	Идентификатор пользователя
login	string		логин пользователя
password	string		пароль пользователя

2. **tblInfo** - информация личного профиля пользователя в системе.

Поле	Тип	Ключ	Описание
info_id	int	PK	Идентификатор публичных данных пользователя
user_id	int	FK	идентификатор пользователя

name	string		Имя пользователя в системе
description	string		Текстовое описание пользователя

3. **tblVect** - цифровой представления пользовательского профиля в системе.

Поле	Тип	Ключ	Описание
vect_id	int	PK	Идентификатор векторного представления пользователя в системе
user_id	int	FK	идентификатор пользователя
vector	string		векторное представление текстового описания пользователя в системе

4. **tblGroups** - распределение пользователей по группам.

Поле	Тип	Ключ	Описание
groups_id	int	PK	Идентификатор
user_id	int	FK	идентификатор пользователя
group_label	int		номер группы

5. **tblSessions** - авторизированные пользователи в системе.

Поле	Тип	Ключ	Описание
session_id	int	PK	Идентификатор сессии

user_id	int	FK	идентификатор пользователя
token	string		сессионный токен
created	date		дата создания токена

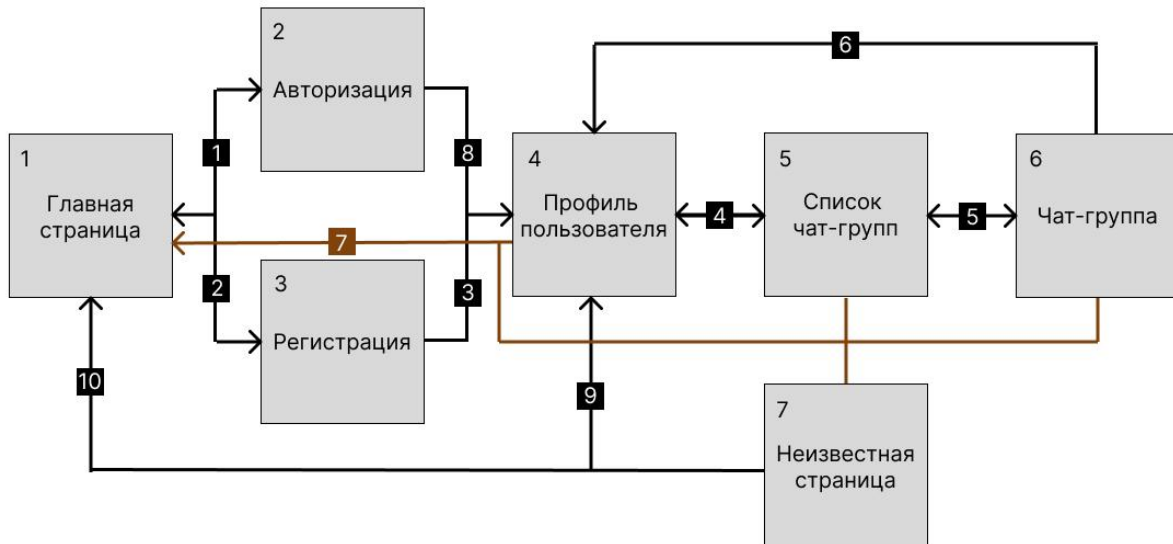
- **Clustering algorithms:**

Модуль содержит следующие функции:

1. **vectorize\_data** - перевод текстовых данных в их векторное представление. Для этих целей используется предобученная модель BERT от DeepPavlov ( <https://huggingface.co/DeepPavlov/rubert-base-cased> ). Текстовое представление токенизируется, разбивается на группы по 768 токенов, каждая последовательность токенов переводится в вектор. Полученные векторные представления фрагментов текста усредняются для получения результирующего вектора.
  - a. **Входные данные:**
    - i. **(str)** текстовое описание пользователя;
  - b. **Выходные данные:**
    - i. **(list[float])** вектор размерности 768.
  
2. **cluster\_users** - кластеризация пользователей на основании векторных представлений текстовых описаний их личного профиля. Используется библиотечная реализация кластерного алгоритма: DBSCAN ( <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html> ). Пользователи, которые были кластеризованы как “выбросы” не добавляются ни в одну из существующих групп.
  - a. **Входные данные:**
    - i. **(list[list[float]])** список векторных представлений пользовательских профилей;
    - ii. **(list[int])** список идентификаторов пользователей, соответствующих передаваемым векторным представлениям.
  - b. **Выходные данные:**
    - i. **(dict[int, int])** словарь пар, где ключу соответствует идентификатор пользователя, а значению соответствует номер группы в случае **успеха**.
    - ii. **(bool)** - False в случае **провала** операции.

## 2.1.2. Интерфейс пользователя

### Карта пользовательского интерфейса:



### Страницы:

1. **Главная страница** - Стартовая страница системы для перехода к регистрации и авторизации;
2. **Авторизация** - страница с формой авторизации в системе;
3. **Регистрация** - страница с формой регистрации в системе;
4. **Профиль пользователя** - страница с личной информацией пользователя в системе: именем и текстовым описанием пользователя;
5. **Список чат-групп** - страница со списком чат-групп, в которые был автоматически добавлен пользователь на основании его текстового описания в личном профиле;
6. **Чат-группа** - страница со списком пользователей, добавленных в данную группу;
7. **Неизвестная страница** - страница-заглушка для отображения при переходе по несуществующим путям в системе.

### Переходы:

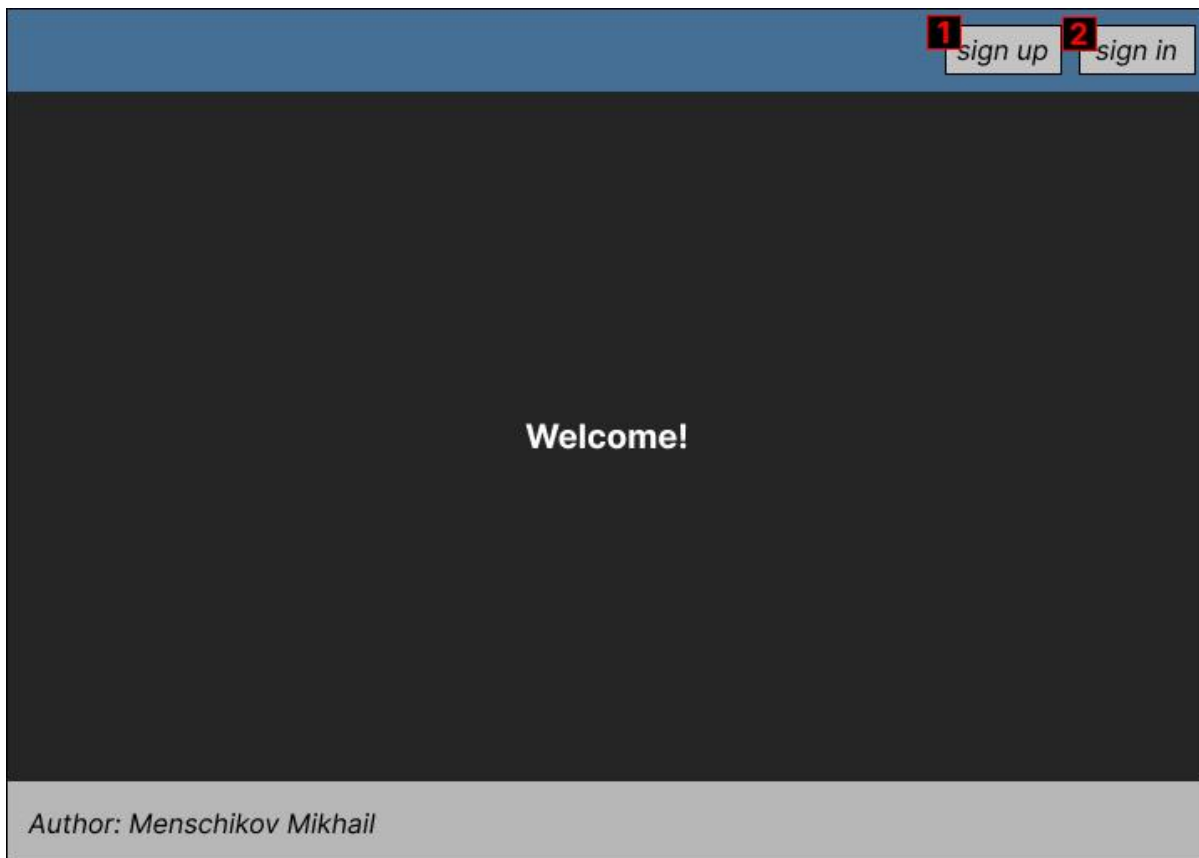
1. Переход с главной страницы системы на форму авторизации пользователя и обратно;



2. Переход с главной страницы системы на форму регистрации пользователя и обратно;
3. Переход в личный профиль пользователя после успешной регистрации;
4. Переход из личного профиля пользователя к списку доступных чат-групп и обратно;
5. Переход со страницы-списка доступных чат-групп к содержанию конкретной чат-группы и обратно;
6. Переход со страницы с содержанием конкретной чат-группы к профилю пользователя, состоящего в данной чат-группе;
7. Выход пользователя из системы и переход на главную страницу в состоянии неавторизованного пользователя;
8. Переход в личный профиль пользователя после успешной авторизации в системе;
9. Возвращение в личный профиль пользователя после попадания на неизвестную страницу в системе (в состоянии авторизованного пользователя);
10. Возвращение на главную страницу после попадания на неизвестную страницу в системе (в состоянии неавторизованного пользователя).

#### **Прототипы интерфейса:**

##### **1. Главная страница**

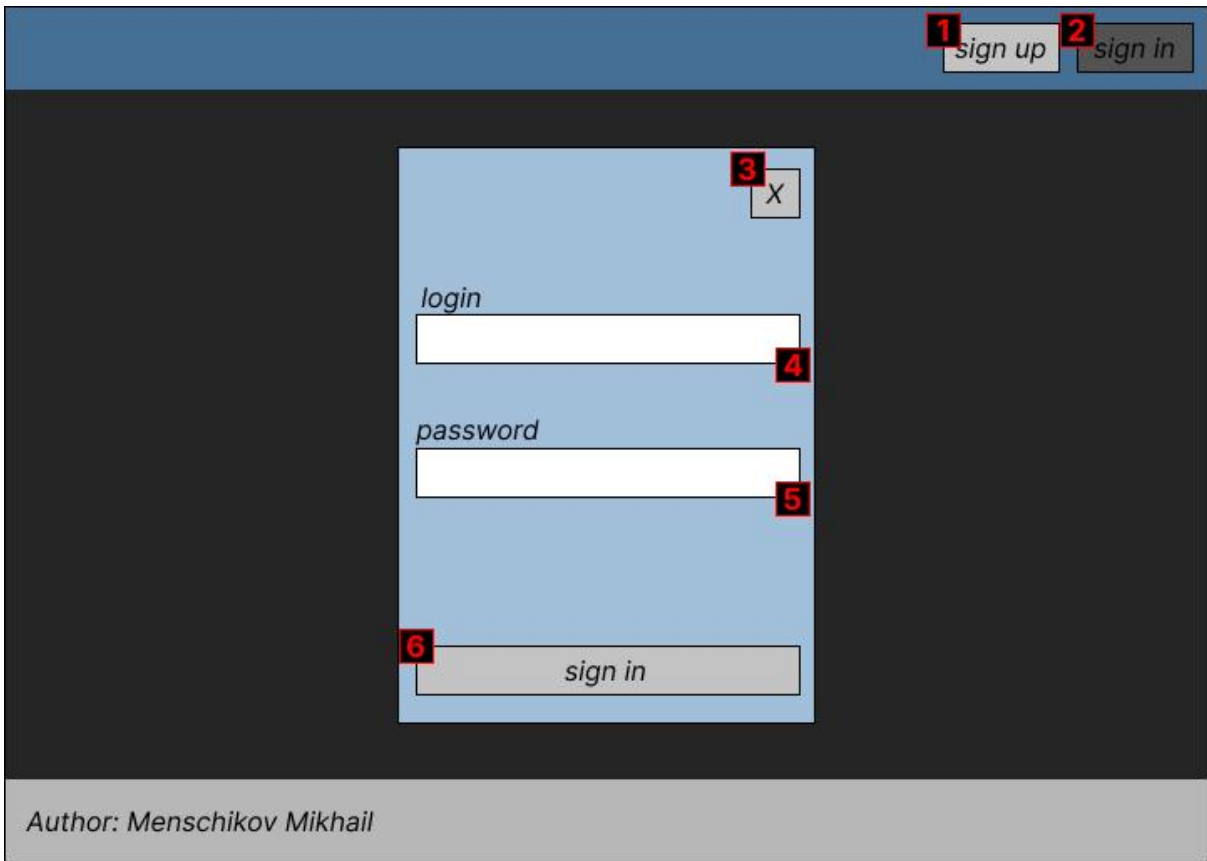


**Пр. интерфейса 1.** Главная страница web-системы.

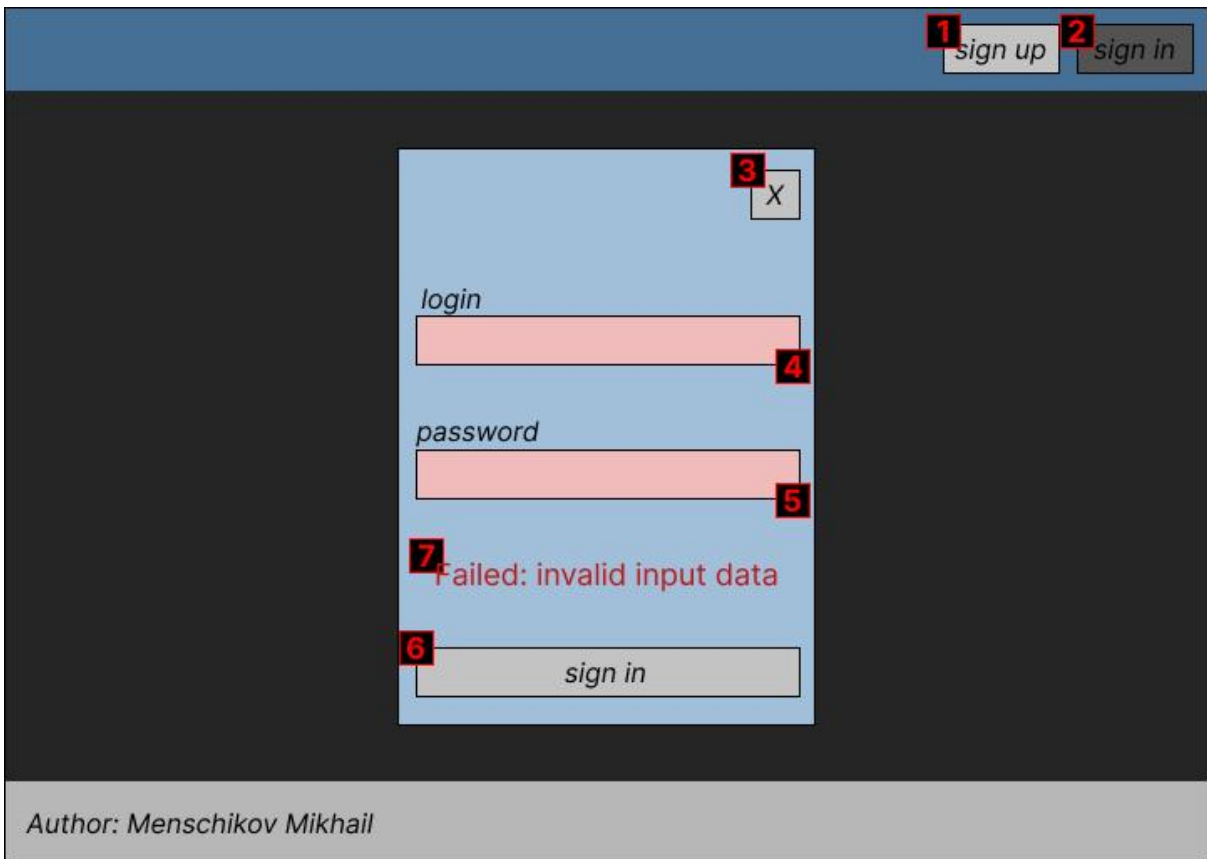
Описание:

1. Кнопка перехода на форму регистрации в системе;
2. Кнопка перехода на форму авторизации в системе.

**2. Авторизация**



Пр. интерфейса 2. Страница авторизации в системе.



### Пр. интерфейса 3. Страница неудачной попытки авторизации в системе.

Описание:

1. Кнопка перехода на форму регистрации в системе;
2. Неактивная кнопка перехода на форму авторизации в системе;
3. Кнопка отмены авторизации;
4. Поле ввода логина;
5. Поле ввода пароля;
6. Кнопка подтверждения авторизации в системе;
7. Сообщение о неудачной попытке авторизации в системе.

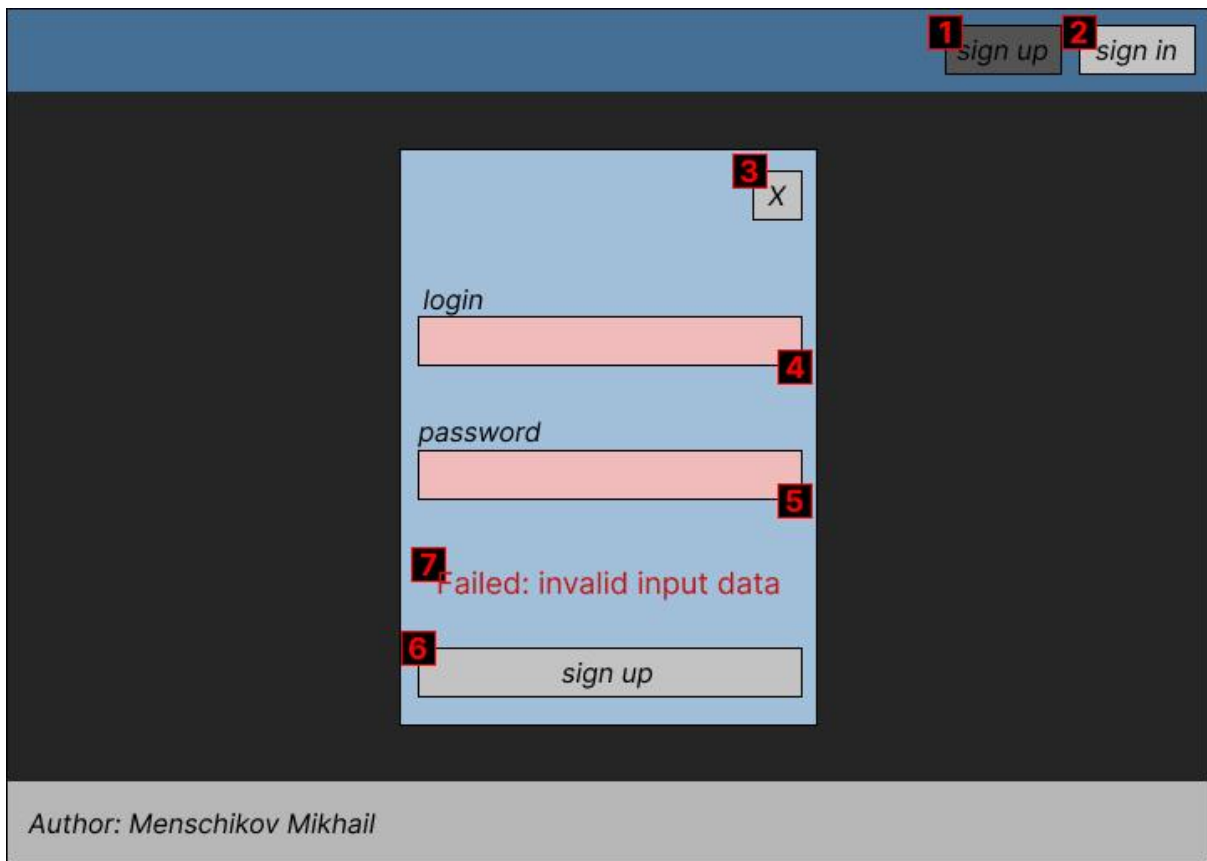
### 3. Регистрация

The image shows a registration form with the following elements and annotations:

- 1**: A "sign up" button in the top right corner of the page.
- 2**: A "sign in" button in the top right corner of the page.
- 3**: A close button (X) in the top right corner of the form.
- 4**: A text input field labeled "login".
- 5**: A text input field labeled "password".
- 6**: A "sign up" button at the bottom of the form.

Author: Menschikov Mikhail

### Пр. интерфейса 4. Страница регистрации в системе.

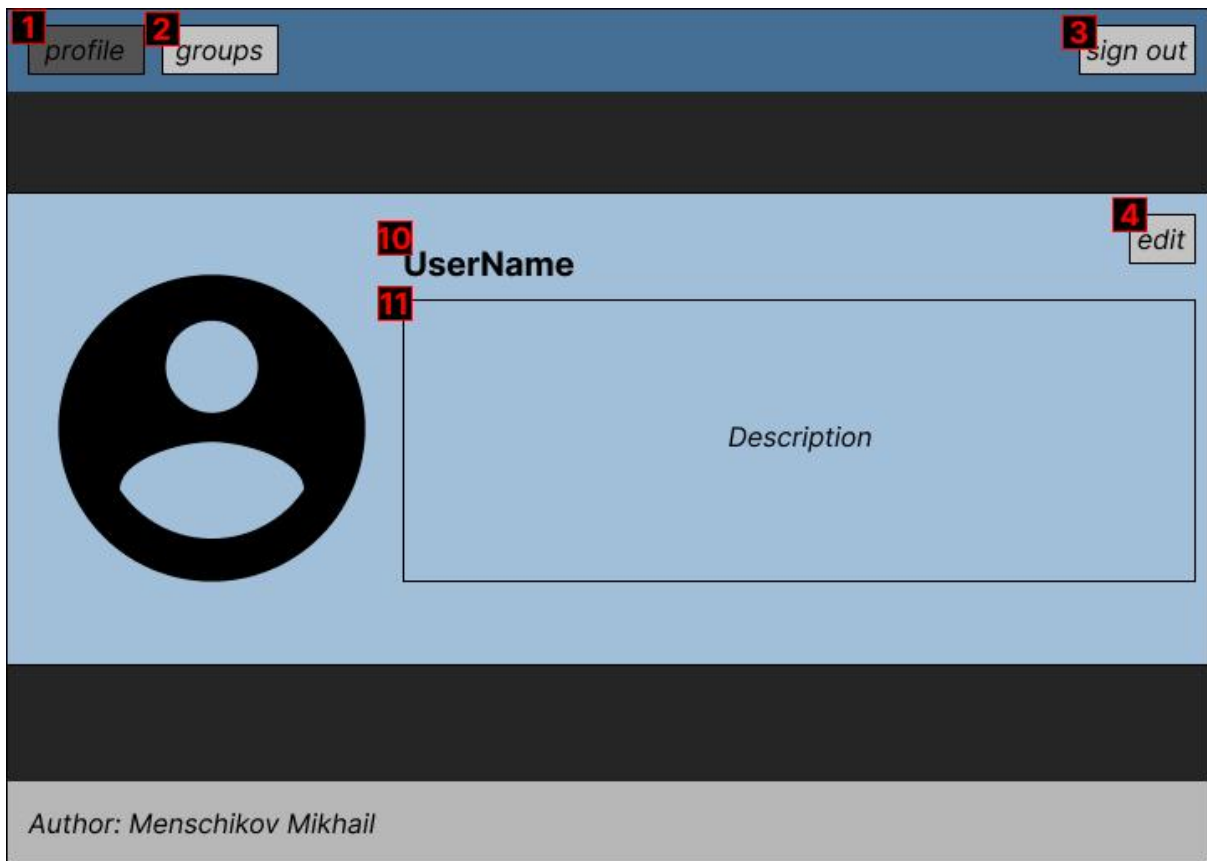


**Пр. интерфейса 5.** Страница неудачной попытки регистрации в системе.

Описание:

1. Неактивная кнопка перехода на форму регистрации в системе;
2. Кнопка перехода на форму авторизации в системе;
3. Кнопка отмены регистрации;
4. Поле ввода логина;
5. Поле ввода пароля;
6. Кнопка подтверждения регистрации в системе;
7. Сообщение о неудачной попытке регистрации в системе.

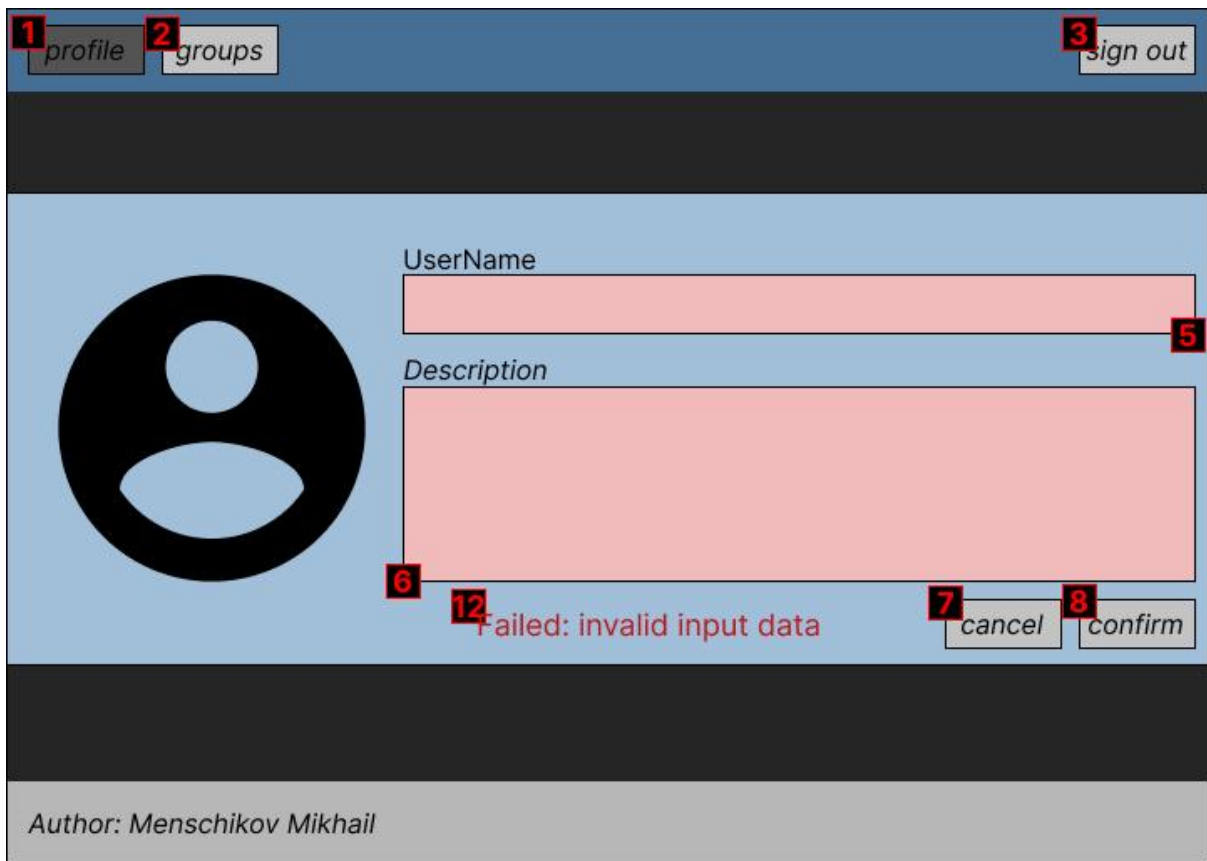
**4. Профиль пользователя**



Пр. интерфейса 6. Личный профиль зарегистрированного пользователя.

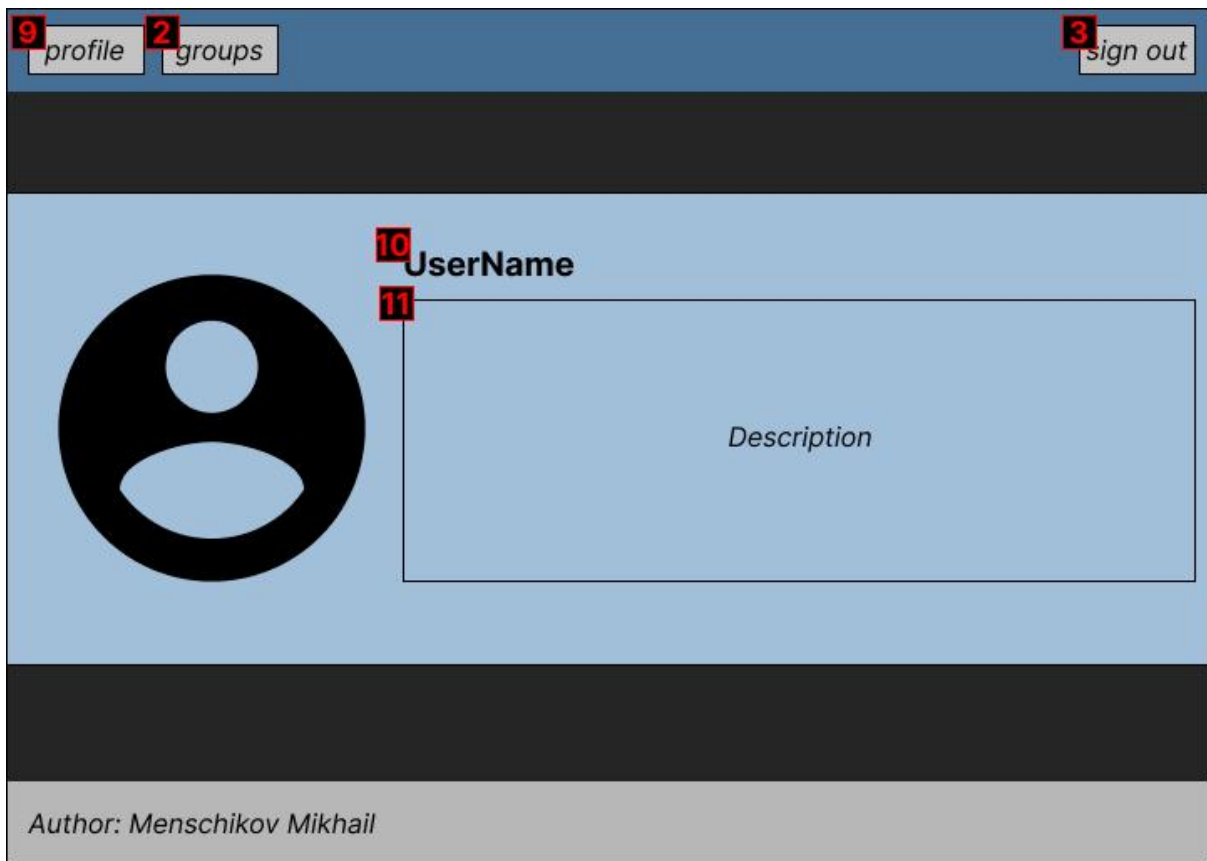
The image shows a web interface for editing a user profile. At the top, there is a navigation bar with three items: 'profile' (highlighted with a red box and number 1), 'groups' (highlighted with a red box and number 2), and 'sign out' (highlighted with a red box and number 3). Below this is a dark blue header. The main content area has a light blue background. On the left is a large black circular placeholder for a profile picture. To the right of the picture are two text input fields: 'UserName' (highlighted with a red box and number 5) and 'Description' (highlighted with a red box and number 6). At the bottom right of the main area are two buttons: 'cancel' (highlighted with a red box and number 7) and 'confirm' (highlighted with a red box and number 8). A dark blue footer contains the text 'Author: Menschikov Mikhail'.

**Пр. интерфейса 7.** Личный профиль зарегистрированного пользователя в состоянии редактирования данных.



**Пр. интерфейса 8.** Страница личного профиля пользователя после неудачной попытки изменения персональных данных.





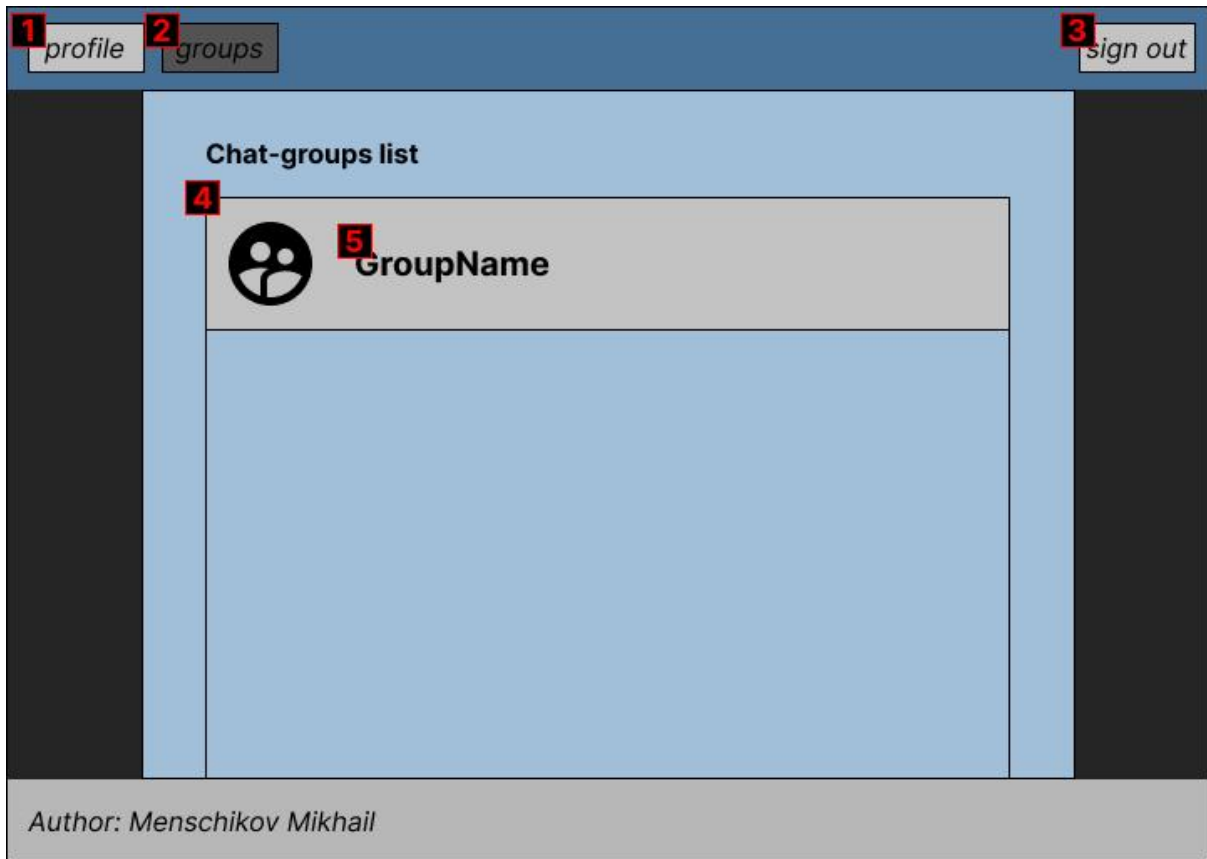
**Пр. интерфейса 9.** Личный профиль при другого пользователя в системе при просмотре.

Описание:

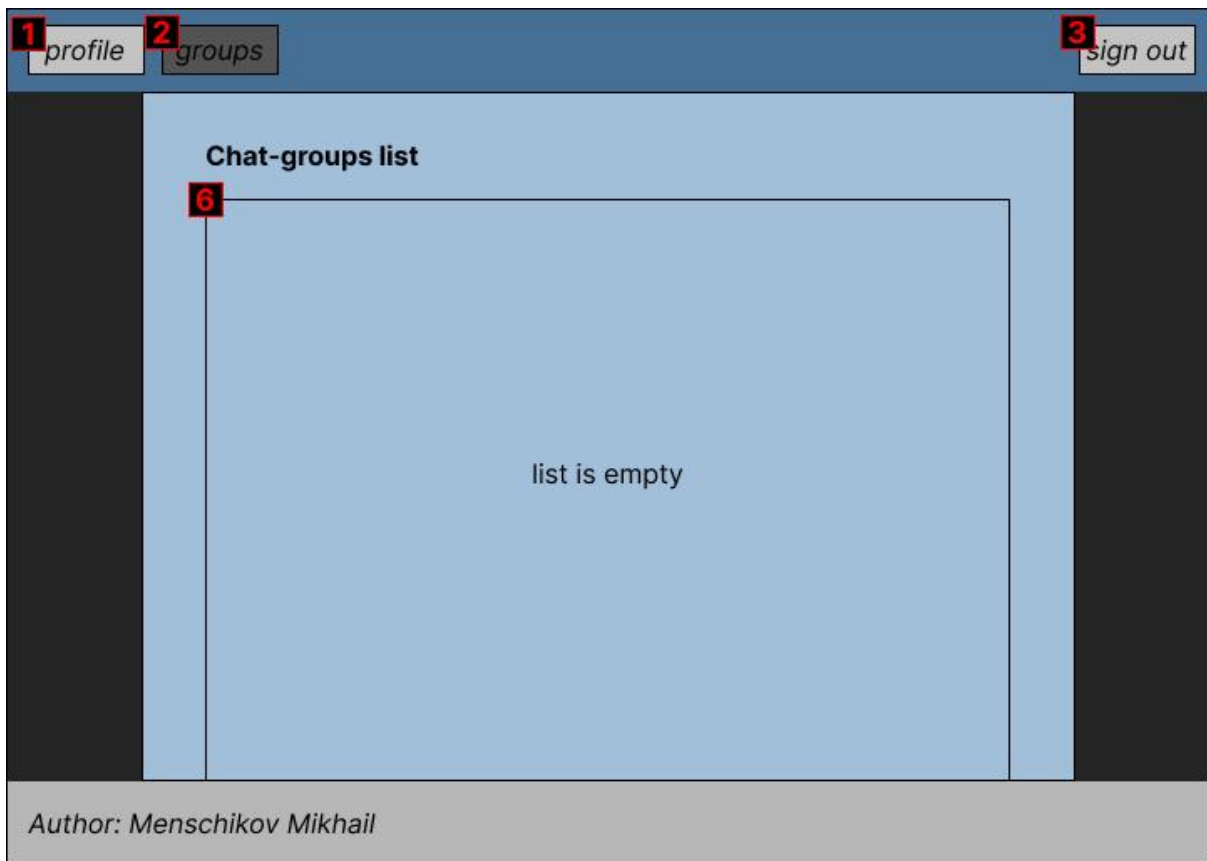
1. Неактивная кнопка перехода в личный профиль пользователя;
2. Кнопка перехода на страницу списка доступных чат-групп для данного пользователя;
3. Кнопка перехода в состояние неавторизованного пользователя;
4. Кнопка перехода в состояние редактирование данных личного профиля;
5. Поле редактирования имени пользователя в системе;
6. Поле редактирования текстового описания пользователя в системе;
7. Кнопка отмены изменения данных личного профиля пользователя;
8. Кнопка подтверждения изменения данных личного профиля пользователя;
9. Кнопка перехода в личный профиль пользователя;
10. Имя пользователя в системе;
11. Текстовое описание пользователя в системе;

12. Сообщение о неудачной попытке изменения персональных данных личного профиля.

## 5. Список чат-групп



Пр. интерфейса 10. Страница с пустым списком доступных чат-групп.

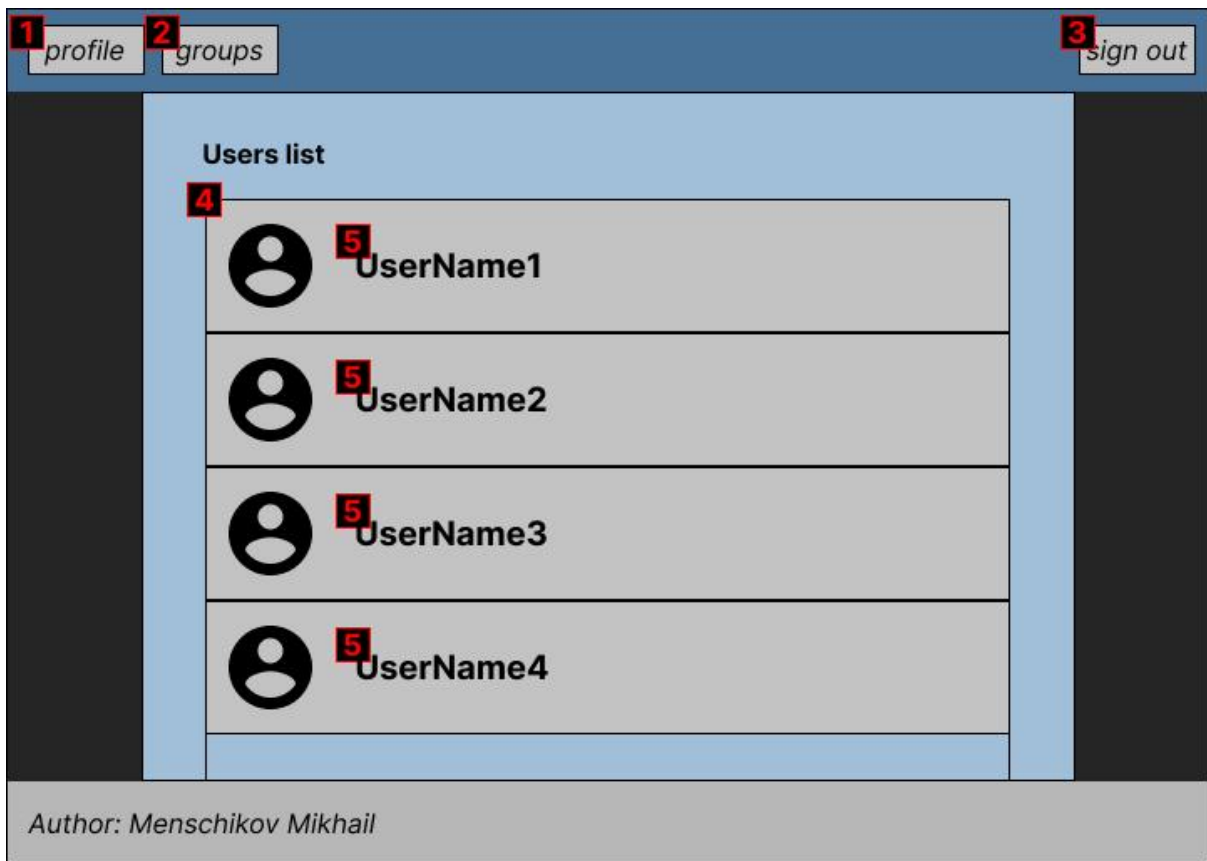


**Пр. интерфейса 11.** Страница с пустым списком доступных чат-групп.

Описание:

1. Кнопка перехода в личный профиль пользователя;
2. Неактивная кнопка перехода на страницу списка доступных чат-групп для данного пользователя;
3. Кнопка перехода в состояние неавторизованного пользователя;
4. Список активных плиток перехода в соответствующие доступные чат-группы;
5. Название доступной чат-группы;
6. Пустой список доступных чат-групп.

## 6. Чат-группа

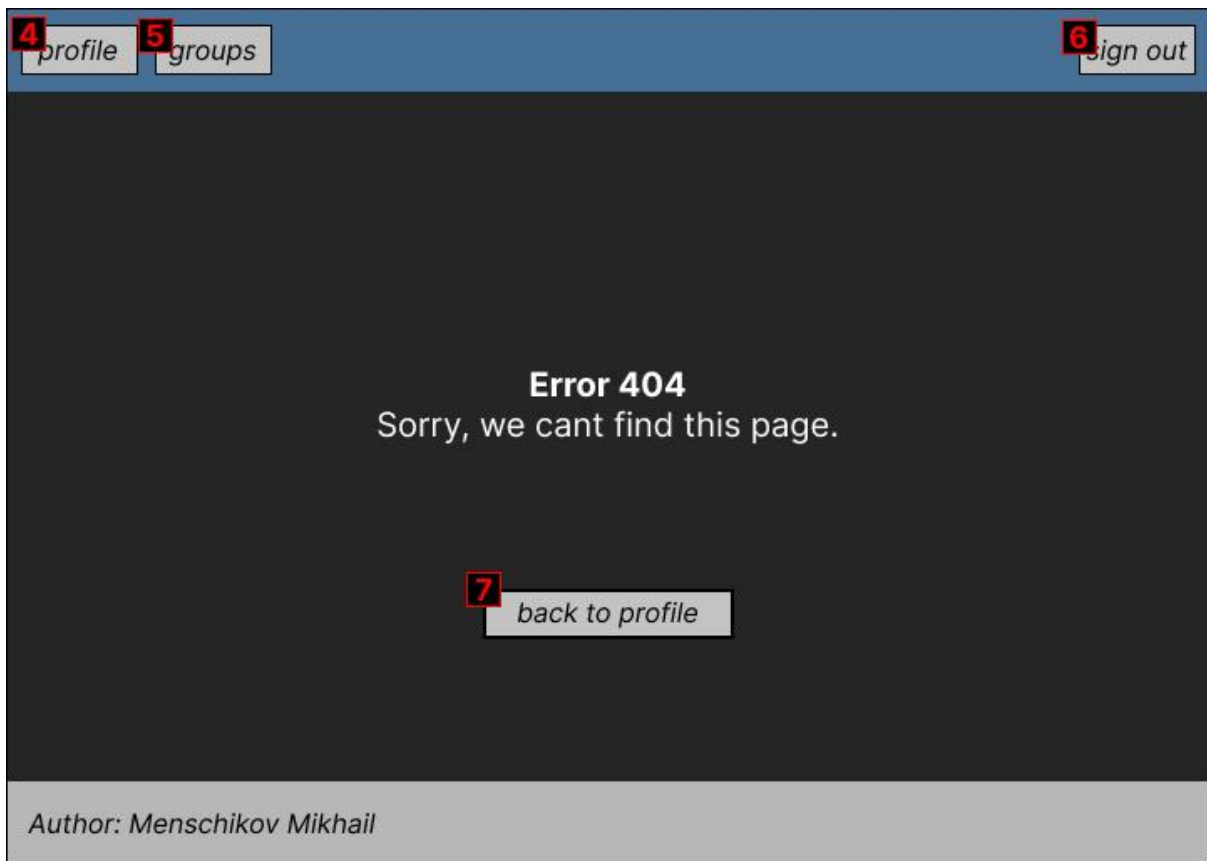


**Пр. интерфейса 12.** Страница со списком пользователей, состоящих в данной чат-группе.

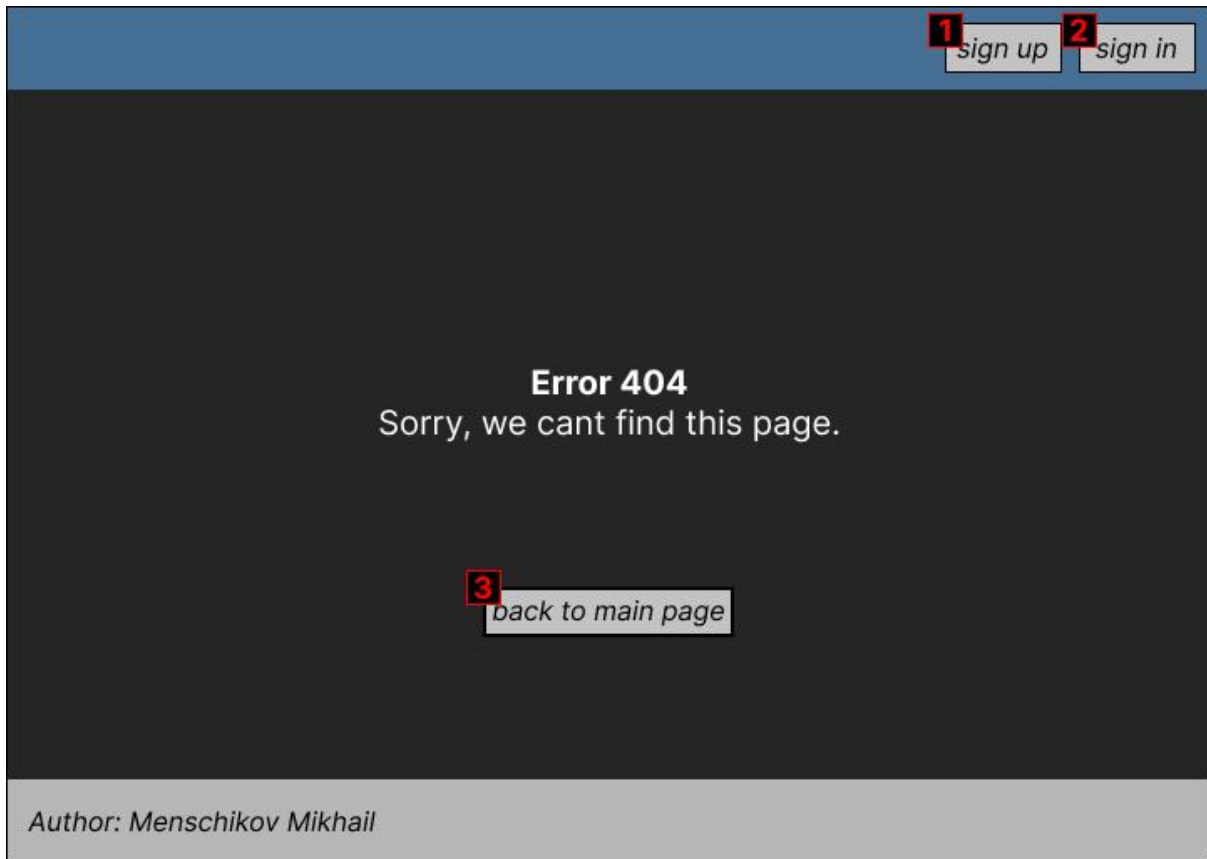
Описание:

1. Кнопка перехода в личный профиль пользователя;
2. Кнопка перехода на страницу списка доступных чат-групп для данного пользователя;
3. Кнопка перехода в состояние неавторизованного пользователя;
4. Список пользователей, состоящих в данной чат-группе;
5. Имя данного пользователя в системе.

## 7. Неизвестная страница



**Пр. интерфейса 13.** Ненайденная страница для авторизованного пользователя.



**Пр. интерфейса 14.** Ненайденная страница для неавторизованного пользователя.

Описание:

1. Кнопка перехода на форму регистрации в системе;
2. Кнопка перехода на форму авторизации в системе;
3. Кнопка перехода на главную страницу системы;
4. Кнопка перехода в личный профиль пользователя;
5. Кнопка перехода на страницу списка доступных чат-групп для данного пользователя;
6. Кнопка перехода в состояние неавторизованного пользователя;
7. Кнопка перехода в личный профиль пользователя.

## 2.2. Стратегия блочного тестирования (БТ)

Блочными тестами будет покрыт функционал модулей 4, 6 высокоуровневой архитектуры: **“Django Backend server”** , **“Clustering**

**algorithms**". Тестирование проводится статическими методами: в уме (на основании спроектированной части архитектуры системы и знаний работы используемых программных инструментов).

Будет протестирован следующий функционал:

- Перевод в векторное представление пустого текстового поля (функция **vectorize\_data**);
- Попытка выполнения кластеризации пользователей с недозаполненными входными параметрами (функция **cluster\_users**).

### 2.3. Стратегия интеграционного тестирования (ИТ)

Интеграционное тестирование будет проведено для интерфейса модуля №4 высокоуровневой архитектуры системы: "**Django backend server**". Тестирование проводится статическими методами: в уме (на основании спроектированной части архитектуры системы и знаний работы используемых программных инструментов).

Будут протестированы следующие моменты:

- Обращение к API по несуществующему пути;
- Попытка получения списка доступных чат-групп другого пользователя (класс **AvailableGroupsAPIView**);
- Попытка изменение содержания публичных данных другого пользователя (класс **ChangeDataAPIView**).

### 2.4. Стратегия аттестационного тестирования (АТ)

Аттестационное тестирование будет проведено для прототипа пользовательского интерфейса. Тестирование проводится статическими методами: в уме (на основании спроектированной части архитектуры системы и знаний работы используемых программных инструментов).

Будут протестированы следующие моменты:

- Просмотр содержания личного профиля другого пользователя;
- Переход по прямой ссылке на главную страницу web-системы, будучи авторизованным пользователем.

### 2.5. Стратегии специальных видов тестирования

#### 2.5.1. Нагрузочное тестирование (НТ)

Нагрузочному тестированию будет подвергаться только модуль №6 высокоуровневой архитектуры системы: "**Clustering algorithms**". Тестирование проводится статическими методами: в уме (на основании спроектированной

части архитектуры системы и знаний работы используемых программных инструментов).

Будут протестированы следующие характеристики:

- Скорость перевода текстовых данных в векторное представление в зависимости от их(текстовых данных) объёма;
- Скорость кластеризации пользователей по группам в зависимости от их(пользователей) объёма.

### 2.5.2. Тестирование безопасности (ТБ)

Тестированию безопасности будет подвергаться прототип пользовательского интерфейса. Тестирование проводится статическими методами: в уме (на основании спроектированной части архитектуры системы и знаний работы используемых программных инструментов).

Будут протестированы следующие моменты:

- Попытка перехода по прямой ссылке на страницу личного профиля без авторизации в системе;
- Взаимодействие с системой в авторизованном состоянии через несколько браузерных окон и последовательный выход;
- попытка авторизации в системе с неверным логином/паролем.

## 2.6. Переход между этапами тестирования

Блочное тестирование проводится параллельно с написанием программного кода системы. Интеграционное тестирование начинается по завершении блочного тестирования. Тестирование безопасности и производительности системы может производиться параллельно после завершения интеграционного тестирования. Аттестационное тестирование проводится после завершения специальных видов тестирования.

## 2.7. Критерии прохождения тестирования

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. Тестирование прошло успешно, если количество пройденных тестов составляет не менее 85% от общего числа, и не было обнаружено критических ошибок.

## 2.8. Условия возобновления и приостановки выполнения тестов

Тестирование должно быть приостановлено, если количество непройденных тестов составляет более 15% от общего количества,



выполненных на текущий момент, или обнаружены критические ошибки, сильно влияющие на функциональные возможности системы.

Возобновление тестирования возможно только после исправления ошибок, обнаруженных в ходе предыдущего тестирования. Повторное тестирование должно быть выполнено с самого начала, начиная с блочных тестов.

### 3. Детальный план тестирования (ПТ)

#### 3.1. Блочное тестирование (ПТ.БТ)

№ ПТ.БТ	№ ТЕСТА	№ ФТ	1	ПТ.БТ1.1	ФТ3.1
Цель теста			проверить поведение функции в случае векторизации пустого текстового поля		
Тип тестирования			негативный		
Объект тестирования			функция <b>vectorize_data</b>		
Входные данные			<ul style="list-style-type: none"> <li>текстовые данные: ""</li> </ul>		
Косвенные входный данные			инициализированная предобученная модель векторизации данных: BERT от DeepPavlov		
Ожидаемый результат			функция завершает выполнения без выпадающих исключений		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	2	ПТ.БТ2.1	ФТ3.1
Цель теста			проверить поведение функции в случае передачи неравного количества парных входных аргументов		
Тип тестирования			негативный		
Объект тестирования			функция <b>cluster_users</b>		
Входные данные			<ul style="list-style-type: none"> <li>количество векторных представлений пользовательских профилей: <b>10</b>;</li> <li>количество идентификаторов пользователей, соответствующих передаваемым векторным представлениям: <b>8</b>.</li> </ul>		
Косвенные входный данные			отсутствуют		
Ожидаемый результат			False		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	3	ПТ.БТ3.1	ФТ1.1
Цель теста			проверить поведение функции в случае передачи на вход пустого логина/пароля		
Тип тестирования			негативный		
Объект тестирования			функция <code>validate_reg</code>		
Входные данные			<ul style="list-style-type: none"> <li>логин: "";</li> <li>пароль: "".</li> </ul>		
Косвенные входный данные			отсутствуют		
Ожидаемый результат			False		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	4	ПТ.БТ3.2	ФТ1.1
Цель теста			проверить поведение функции в случае наличия данного логина/пароля в БД		
Тип тестирования			позитивный		
Объект тестирования			функция <code>validate_reg</code>		
Входные данные			<ul style="list-style-type: none"> <li>логин: "qwerty";</li> <li>пароль: "0123456".</li> </ul>		
Косвенные входный данные			используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные авторизация/регистрация"		
Ожидаемый результат			False		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	5	ПТ.БТ3.3	ФТ1.1
Цель теста			проверить поведение функции в случае отсутствия данного логина/пароля в БД		
Тип тестирования			позитивный		

Объект тестирования	функция <code>validate_reg</code>
Входные данные	<ul style="list-style-type: none"> <li>логин: "qwertyzxcv";</li> <li>пароль: "0123456789".</li> </ul>
Косвенные входный данные	используется мок-объект имитирующий поведение ORM для доступа к данным БД. Содержание мок-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные авторизация/регистрация"
Ожидаемый результат	True

№ ПТ.БТ	№ ТЕСТА	№ ФТ	6	ПТ.БТ3.4	ФТ1.1
Цель теста			проверить поведение функции в случае наличия в логине/пароле недопустимых символов		
Тип тестирования			негативный		
Объект тестирования			функция <code>validate_reg</code>		
Входные данные			<ul style="list-style-type: none"> <li>логин: "qwertyzxcv";</li> <li>пароль: "0123?456789".</li> </ul>		
Косвенные входный данные			используется мок-объект имитирующий поведение ORM для доступа к данным БД. Содержание мок-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные авторизация/регистрация"		
Ожидаемый результат			False		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	7	ПТ.БТ4.1	ФТ1.2
Цель теста			проверить поведение функции в случае отсутствия данного логина/пароля в БД		
Тип тестирования			позитивный		
Объект тестирования			функция <code>validate_auth</code>		

Входные данные	<ul style="list-style-type: none"> <li>логин: "qwertyzxcv"</li> <li>пароль: "qwertyzxcv"</li> </ul>
Косвенные входный данные	используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные авторизация/регистрация"
Ожидаемый результат	False

№ ПТ.БТ	№ ТЕСТА	№ ФТ	8	ПТ.БТ4.2	ФТ1.2
Цель теста			проверить поведение функции в случае наличия данного логина/пароля в БД		
Тип тестирования			позитивный		
Объект тестирования			функция <code>validate_auth</code>		
Входные данные			<ul style="list-style-type: none"> <li>логин: "qwerty"</li> <li>пароль: "0123456"</li> </ul>		
Косвенные входный данные			используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные авторизация/регистрация"		
Ожидаемый результат			True		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	9	ПТ.БТ5.1	ФТ2.1
Цель теста			проверить поведение функции в случае подачи на вход несуществующего поля с данными (тип изменения)		
Тип тестирования			негативный		
Объект тестирования			функция <code>make_changes</code>		

Входные данные	<ul style="list-style-type: none"> <li>тип изменения: "image";</li> <li>данные на замену: "fake image text";</li> <li>идентификатор пользователя: 2.</li> </ul>
Косвенные входный данные	используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Личный профиль"
Ожидаемый результат	False

№ ПТ.БТ	№ ТЕСТА	№ ФТ	10	ПТ.БТ5.2	ФТ2.1
Цель теста			проверить поведение функции в случае подачи на вход несуществующего идентификатора пользователя		
Тип тестирования			негативный		
Объект тестирования			функция make_changes		
Входные данные			<ul style="list-style-type: none"> <li>тип изменения: "name";</li> <li>данные на замену: "new name";</li> <li>идентификатор пользователя: -1.</li> </ul>		
Косвенные входный данные			используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Личный профиль"		
Ожидаемый результат			False		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	11	ПТ.БТ6.1	ФТ3.1, ФТ3.2, ФТ3.2
Цель теста			проверить поведение функции в случае успешного перераспределения пользователей по группам		
Тип тестирования			ПОЗИТИВНЫЙ		

Объект тестирования	функция <b>recalc_groups</b>
Входные данные	отсутствуют
Косвенные входные данные	используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Цифровой профиль"
Ожидаемый результат	True

№ ПТ.БТ	№ ТЕСТА	№ ФТ	12	ПТ.БТ7.1	ФТ3.4
Цель теста			проверить поведение функции в случае наличия доступных чат-групп для данного пользователя		
Тип тестирования			позитивный		
Объект тестирования			функция <b>prepare_group_list</b>		
Входные данные			<ul style="list-style-type: none"> <li>идентификатор пользователя: 2</li> </ul>		
Косвенные входные данные			используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Чат-группы"		
Ожидаемый результат			[ 1 ]		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	13	ПТ.БТ7.2	ФТ3.4
Цель теста			проверить поведение функции в случае подачи на вход идентификатора несуществующего пользователя		
Тип тестирования			негативный		
Объект тестирования			функция <b>prepare_group_list</b>		
Входные данные			<ul style="list-style-type: none"> <li>идентификатор пользователя: -2</li> </ul>		

<b>Косвенные входный данные</b>	используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Чат-группы"
<b>Ожидаемый результат</b>	пустой список

№ ПТ.БТ	№ ТЕСТА	№ ФТ	14	ПТ.БТ8.1	ФТ3.5
<b>Цель теста</b>			проверить поведение функции в случае подачи на вход идентификатора несуществующей чат-группы		
<b>Тип тестирования</b>			негативный		
<b>Объект тестирования</b>			функция <code>prepare_chatgroup_info</code>		
<b>Входные данные</b>			<ul style="list-style-type: none"> <li>идентификатор чат-группы: -2</li> </ul>		
<b>Косвенные входный данные</b>			используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Чат-группы"		
<b>Ожидаемый результат</b>			False		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	15	ПТ.БТ9.1	ФТ3.6
<b>Цель теста</b>			проверить поведение функции в случае подачи на вход логина несуществующего пользователя		
<b>Тип тестирования</b>			негативный		
<b>Объект тестирования</b>			функция <code>prepare_profile_info</code>		
<b>Входные данные</b>			<ul style="list-style-type: none"> <li>логин пользователя: "not_existed_login"</li> </ul>		
<b>Косвенные входный данные</b>			используется mock-объект имитирующий поведение ORM для доступа к данным БД. Содержание mock-объекта представлено в		



	разделе "Приложения" пункт "Косвенные входные данные Личный профиль"
Ожидаемый результат	False

№ ПТ.БТ	№ ТЕСТА	№ ФТ	16	ПТ.БТ9.2	ФТ3.6
Цель теста			проверить поведение функции в случае подачи на вход логина существующего пользователя		
Тип тестирования			позитивный		
Объект тестирования			функция <code>prepare_profile_info</code>		
Входные данные			<ul style="list-style-type: none"> <li>логин пользователя: "existed_login"</li> </ul>		
Косвенные входный данные			используется мок-объект имитирующий поведение ORM для доступа к данным БД. Содержание мок-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Личный профиль"		
Ожидаемый результат			("Anna", "i like ride a bike")		

№ ПТ.БТ	№ ТЕСТА	№ ФТ	17	ПТ.БТ10.1	ФТ2.1, ФТ2.2, ФТ3.4, ФТ3.5
Цель теста			проверить поведение функции в случае наличия необходимого сессионного токена в БД		
Тип тестирования			позитивный		
Объект тестирования			функция <code>validate_token</code>		
Входные данные			<ul style="list-style-type: none"> <li>сессионный токен: "test_session_key"</li> </ul>		
Косвенные входный данные			используется мок-объект имитирующий поведение ORM для доступа к данным БД. Содержание мок-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Сессионные ключи"		

Ожидаемый результат	True
---------------------	------

№ ПТ.БТ	№ ТЕСТА	№ ФТ	18	ПТ.БТ11.1	ФТ1.2, ФТ1.1
Цель теста			проверить поведение функции в случае подачи на вход несуществующего идентификатора пользователя		
Тип тестирования			негативный		
Объект тестирования			функция <code>create_session_token</code>		
Входные данные			<ul style="list-style-type: none"> <li>идентификатор пользователя: -1</li> </ul>		
Косвенные входный данные			используется мок-объект имитирующий поведение ORM для доступа к данным БД. Содержание мок-объекта представлено в разделе "Приложения" пункт "Косвенные входные данные Авторизация/Регистрация"		
Ожидаемый результат			False		

### 3.2. Интеграционное тестирование (ПТ.ИТ)

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	1	ПТ.ИТ1.1	ФТ1.1
Цель теста			Отправка пост запроса на Django API-сервер для регистрации нового пользователя в системе		
Тип тестирования			позитивный		
Объект тестирования			RegisterAPIView (путь "/register")		
Входные данные			json-объект со следующими полями: <ul style="list-style-type: none"> <li><b>login:</b> "new_user_login"</li> <li><b>password:</b> "439078431"</li> </ul>		
Косвенные входный данные			Таблица БД <code>tblUser</code> пустая		

Ожидаемый результат	<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>token:</b> *сгенерированный сессионный токен*</li> </ul>
---------------------	---

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	2	ПТ.ИТ1.2	ФТ1.1
Цель теста			Отправка пост запроса на Django API-сервер для регистрации нового пользователя с занятым логином/паролем		
Тип тестирования			ПОЗИТИВНЫЙ		
Объект тестирования			RegisterAPIView (путь "/register")		
Входные данные			<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul>		
Косвенные входный данные			<p>Таблица БД <b>tblUser</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• <b>user_id:</b> 1;</li> <li>• <b>login:</b> "new_user_login";</li> <li>• <b>password:</b> "439078431".</li> </ul>		
Ожидаемый результат			<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	3	ПТ.ИТ1.3	ФТ1.1
Цель теста			Отправка пост запроса на Django API-сервер без поля <b>login</b> в json-объекте		
Тип тестирования			НЕГАТИВНЫЙ		
Объект тестирования			RegisterAPIView (путь "/register")		
Входные данные			<p>json-объект со следующими полями:</p>		

	<ul style="list-style-type: none"> <li>• <b>password:</b> "439078431"</li> </ul>
<b>Косвенные входный данные</b>	Таблица БД <b>tblUser</b> пустая
<b>Ожидаемый результат</b>	json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	4	ПТ.ИТ1.4	ФТ1.1
<b>Цель теста</b>			Отправка пост запроса на Django API-сервер без json-объекта		
<b>Тип тестирования</b>			негативный		
<b>Объект тестирования</b>			RegisterAPIView (путь "/register")		
<b>Входные данные</b>			Отсутствуют		
<b>Косвенные входный данные</b>			Таблица БД <b>tblUser</b> пустая		
<b>Ожидаемый результат</b>			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	5	ПТ.ИТ2.1	ФТ1.2
<b>Цель теста</b>			Отправка пост запроса на Django API-сервер для авторизации в системе и получения сессионного токена		
<b>Тип тестирования</b>			позитивный		
<b>Объект тестирования</b>			AuthorizeAPIView (путь "/auth")		
<b>Входные данные</b>			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul>		
<b>Косвенные входный данные</b>			Таблица БД <b>tblUser</b> содержит запись:		

	<ul style="list-style-type: none"> <li>• <b>user_id:</b> 1;</li> <li>• <b>login:</b> "new_user_login";</li> <li>• <b>password:</b> "439078431"</li> </ul>
Ожидаемый результат	<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>token:</b> *сгенерированный сессионный токен*</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	6	ПТ.ИТ2.2	ФТ1.2
Цель теста			Отправка пост запроса на Django API-сервер для авторизации в системе без наличия необходимых записей в БД		
Тип тестирования			ПОЗИТИВНЫЙ		
Объект тестирования			AuthorizeAPIView (путь "/auth")		
Входные данные			<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul>		
Косвенные входный данные			Таблица БД <b>tblUser</b> пустая		
Ожидаемый результат			<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	7	ПТ.ИТ2.3	ФТ1.2
Цель теста			Отправка пост-запроса на Django API-сервер для авторизации без поля <b>login</b> в json-объекте		
Тип тестирования			НЕГАТИВНЫЙ		
Объект тестирования			AuthorizeAPIView (путь "/auth")		

<b>Входные данные</b>	json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>password:</b> "439078431"</li> </ul>
<b>Косвенные входный данные</b>	Таблица БД <b>tblUser</b> содержит запись: <ul style="list-style-type: none"> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul>
<b>Ожидаемый результат</b>	json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	8	ПТ.ИТ2.4	ФТ1.2
<b>Цель теста</b>			Отправка пост-запроса на Django API-сервер для авторизации без json-объекта		
<b>Тип тестирования</b>			негативный		
<b>Объект тестирования</b>			AuthorizeAPIView (путь "/auth")		
<b>Входные данные</b>			Отсутствуют		
<b>Косвенные входный данные</b>			Таблица БД <b>tblUser</b> содержит запись: <ul style="list-style-type: none"> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul>		
<b>Ожидаемый результат</b>			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	9	ПТ.ИТ3.1	ФТ2.1, ФТ3.1
<b>Цель теста</b>			Отправка пост-запроса на Django API-сервер для изменения текстового описания пользователя в системе		
<b>Тип тестирования</b>			позитивный		

<b>Объект тестирования</b>	ChangeDataAPIView (путь “/change”)
<b>Входные данные</b>	<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>type:</b> “description”</li> <li>• <b>data:</b> “new_description”</li> <li>• <b>token:</b> “user_session_key”</li> </ul>
<b>Косвенные входный данные</b>	<p>Таблица БД <b>tblUser</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• <b>user_id:</b> 1</li> <li>• <b>login:</b> “new_user_login”</li> <li>• <b>password:</b> “439078431”</li> </ul> <p>Таблица БД <b>tblSession</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• session_id: 1</li> <li>• user_id: 1</li> <li>• token: “user_session_key”</li> </ul> <p>Таблица БД <b>tblInfo</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• info_id: 1</li> <li>• user_id: 1</li> <li>• name: “new_user_login”</li> <li>• description: “”</li> </ul>
<b>Ожидаемый результат</b>	<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>error:</b> False</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	10	ПТ.ИТ3.2	ФТ2.1
<b>Цель теста</b>	Отправка пост-запроса на Django API-сервер без json-объекта				
<b>Тип тестирования</b>	негативный				
<b>Объект тестирования</b>	ChangeDataAPIView (путь “/change”)				
<b>Входные данные</b>	Отсутствует				
<b>Косвенные входный данные</b>	<p>Таблица БД <b>tblUser</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• <b>user_id:</b> 1</li> <li>• <b>login:</b> “new_user_login”</li> <li>• <b>password:</b> “439078431”</li> </ul>				

	<p>Таблица БД <b>tblSession</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● session_id: 1</li> <li>● user_id: 1</li> <li>● token: “user_session_key”</li> </ul> <p>Таблица БД <b>tblInfo</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● info_id: 1</li> <li>● user_id: 1</li> <li>● name: “new_user_login”</li> <li>● description: “”</li> </ul>
<b>Ожидаемый результат</b>	<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>● <b>error:</b> True</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	11	ПТ.ИТ3.3	ФТ2.1
<b>Цель теста</b>			Отправка пост-запроса на Django API-сервер с несуществующим значением поля <b>type</b>		
<b>Тип тестирования</b>			негативный		
<b>Объект тестирования</b>			ChangeDataAPIView (путь “/change”)		
<b>Входные данные</b>			<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>● <b>type:</b> “image”</li> <li>● <b>data:</b> “new_description”</li> <li>● <b>token:</b> “user_session_key”</li> </ul>		
<b>Косвенные входный данные</b>			<p>Таблица БД <b>tblUser</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● <b>user_id:</b> 1</li> <li>● <b>login:</b> “new_user_login”</li> <li>● <b>password:</b> “439078431”</li> </ul> <p>Таблица БД <b>tblSession</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● session_id: 1</li> <li>● user_id: 1</li> <li>● token: “user_session_key”</li> </ul> <p>Таблица БД <b>tblInfo</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● info_id: 1</li> <li>● user_id: 1</li> <li>● name: “new_user_login”</li> </ul>		



	<ul style="list-style-type: none"> <li>• description: ""</li> </ul>
Ожидаемый результат	json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	12	ПТ.ИТ3.4	ФТ2.1
Цель теста			Отправка пост-запроса на Django API-сервер без передачи сессионного токена		
Тип тестирования			негативный		
Объект тестирования			ChangeDataAPIView (путь "/change")		
Входные данные			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>type:</b> "image"</li> <li>• <b>data:</b> "new_description"</li> </ul>		
Косвенные входный данные			Таблица БД <b>tblUser</b> содержит запись: <ul style="list-style-type: none"> <li>• <b>user_id:</b> 1</li> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul> Таблица БД <b>tblSession</b> содержит запись: <ul style="list-style-type: none"> <li>• session_id: 1</li> <li>• user_id: 1</li> <li>• token: "user_session_key"</li> </ul> Таблица БД <b>tblInfo</b> содержит запись: <ul style="list-style-type: none"> <li>• info_id: 1</li> <li>• user_id: 1</li> <li>• name: "new_user_login"</li> <li>• description: ""</li> </ul>		
Ожидаемый результат			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	13	ПТ.ИТ4.1	ФТ3.3
---------	---------	------	----	----------	-------

Цель теста	Отправка пост-запроса на Django API-сервер для получения списка доступных для данного пользователя чат-групп
Тип тестирования	ПОЗИТИВНЫЙ
Объект тестирования	AvailableGroupsAPIView (путь "/groups")
Входные данные	json-объект со следующими полями: <ul style="list-style-type: none"> <li>● <b>token:</b> "user_session_key"</li> </ul>
Косвенные входный данные	<p>Таблица БД <b>tblUser</b> содержит записи:</p> <ul style="list-style-type: none"> <li>● <b>user_id:</b> 1</li> <li>● <b>login:</b> "new_user_login"</li> <li>● <b>password:</b> "439078431"</li> </ul> <p>Таблица БД <b>tblSession</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● session_id: 1</li> <li>● user_id: 1</li> <li>● token: "user_session_key"</li> </ul> <p>Таблица БД <b>tblInfo</b> содержит запись:</p> <ul style="list-style-type: none"> <li>● info_id: 1</li> <li>● user_id: 1</li> <li>● name: "new_user_login"</li> <li>● description: ""</li> </ul> <p>Таблица БД <b>tblGroups</b> пустая.</p>
Ожидаемый результат	пустой json-объект

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	14	ПТ.ИТ4.2	ФТ3.3
Цель теста	Отправка пост-запроса на Django API-сервер без передачи сессионного токена				
Тип тестирования	негативный				
Объект тестирования	AvailableGroupsAPIView (путь "/groups")				
Входные данные	Пустой json-объект				
Косвенные входный данные	Таблица БД <b>tblUser</b> содержит записи:				

	<ul style="list-style-type: none"> <li>• <b>user_id:</b> 1</li> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>password:</b> "439078431"</li> </ul> <p>Таблица БД <b>tblSession</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• session_id: 1</li> <li>• user_id: 1</li> <li>• token: "user_session_key"</li> </ul> <p>Таблица БД <b>tblInfo</b> содержит запись:</p> <ul style="list-style-type: none"> <li>• info_id: 1</li> <li>• user_id: 1</li> <li>• name: "new_user_login"</li> <li>• description: ""</li> </ul> <p>Таблица БД <b>tblGroups</b> пустая.</p>
<b>Ожидаемый результат</b>	<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	15	ПТ.ИТ5.1	ФТ3.4
<b>Цель теста</b>			Отправка пост-запроса на Django API-сервер для получения списка пользователей, принадлежащих чат-группе с переданным идентификатором		
<b>Тип тестирования</b>			ПОЗИТИВНЫЙ		
<b>Объект тестирования</b>			ChatGroupAPIView (путь "/chat")		
<b>Входные данные</b>			<p>json-объект со следующими полями:</p> <ul style="list-style-type: none"> <li>• <b>group_num:</b> 1</li> <li>• <b>token:</b> "user_session_key"</li> </ul>		
<b>Ожидаемый результат</b>			<p>json-объект:</p> <pre> {{login: "cur_user2",name: "cur_user2"}, {login: "cur_user3",name: "cur_user3"}, {login: "new_user_login",name: "new_user_login"}} </pre>		

	}
--	---

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	16	ПТ.ИТ5.2	ФТ3.4
Цель теста			Отправка пост-запроса на Django API-сервер с сессионным токеном пользователя, который не состоит в данной чат-группе		
Тип тестирования			негативный		
Объект тестирования			ChatGroupAPIView (путь "/chat")		
Входные данные			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>group_num:</b> 1</li> <li>• <b>token:</b> "user_session_key2"</li> </ul>		
Ожидаемый результат			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	17	ПТ.ИТ5.3	ФТ3.4
Цель теста			Отправка пост-запроса на Django API-сервер с несуществующим номером чат-группы		
Тип тестирования			негативный		
Объект тестирования			ChatGroupAPIView (путь "/chat")		
Входные данные			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>group_num:</b> -1</li> <li>• <b>token:</b> "user_session_key"</li> </ul>		
Ожидаемый результат			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	18	ПТ.ИТ6.1	ФТ2.2, ФТ3.6
Цель теста			Отправка пост-запроса на Django API-сервер для получения публичной информации пользователя с указанным логином		
Тип тестирования			позитивный		
Объект тестирования			ProfileAPIView (путь "/profile")		
Входные данные			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>login:</b> "new_user_login"</li> <li>• <b>token:</b> "user_session_key"</li> </ul>		
Ожидаемый результат			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>description:</b> ""</li> <li>• <b>name:</b> ""</li> </ul>		

№ ПТ.ИТ	№ ТЕСТА	№ ФТ	19	ПТ.ИТ6.2	ФТ2.2, ФТ3.6
Цель теста			Отправка пост-запроса на Django API-сервер без json-объекта		
Тип тестирования			негативный		
Объект тестирования			ProfileAPIView (путь "/profile")		
Входные данные			Отсутствуют		
Ожидаемый результат			json-объект со следующими полями: <ul style="list-style-type: none"> <li>• <b>error:</b> True</li> </ul>		

### 3.3. Аттестационное тестирование (ПТ.АТ)

№ ПТ.АТ	№ ТЕСТА	№ ФТ	1	ПТ.АТ1.1	ФТ2.1
Цель теста			Изменение публичных данных личного профиля.		
Тип тестирования			позитивный		
Сценарий действий пользователя и ответных реакций системы для проверки			<ol style="list-style-type: none"> <li>1. Пользователь находится в состоянии авторизованного на странице <b>Личного профиля</b>.</li> <li>2. Нажать на кнопку <b>“edit”</b>.</li> <li>3. Система переключит профиль в состояние редактирования.</li> <li>4. Ввести текст <b>“new_name”</b> в поле для редактирования имени. Нажать кнопку <b>“confirm”</b>.</li> <li>5. Система сохранит изменения публичной информации пользователя.</li> </ol>		

№ ПТ.АТ	№ ТЕСТА	№ ФТ	2	ПТ.АТ2.1	ФТ3.4, ФТ3.5
Цель теста			Просмотр содержания конкретной чат-группы.		
Тип тестирования			позитивный		
Сценарий действий пользователя и ответных реакций системы для проверки			<ol style="list-style-type: none"> <li>1. Пользователь находится в состоянии авторизованного. У данного пользователя есть одна доступная группа.</li> <li>2. Нажать на панель с название чат-группы</li> <li>3. Система отобразит список пользователей, состоящих в данной чат-группе</li> </ol>		

№ ПТ.АТ	№ ТЕСТА	№ ФТ	3	ПТ.АТ3.1	ФТ3.6
Цель теста			Переход на страницу другого пользователя		
Тип тестирования			позитивный		

<p><b>Сценарий действий пользователя и ответных реакций системы для проверки</b></p>	<ol style="list-style-type: none"> <li>1. Пользователь находится в состоянии авторизованного. пользователь находится на странице чат-группы, в которую он был добавлен.</li> <li>2. Пользователь нажимает на любую панель с именем другого пользователя</li> <li>3. Система отображает личный профиль данного пользователя.</li> </ol>
--	--

### 3.4. Нагрузочное тестирование (ПТ.НТ)

№ ПТ.НТ	№ ТЕСТА	№ ФТ	1	ПТ.НТ1.1	ФТ3.1
Цель теста			Оценить скорость перевода текстовых данных в векторное представление в зависимости от их(текстовых данных) объёма		
Тип тестирования			негативный		
Объект тестирования			Функция <b>vectorize_data</b>		
Входные данные			<p>Последовательно функции подаются текстовые строки размером:</p> <ul style="list-style-type: none"> <li>• 500 символов;</li> <li>• 1000 символов;</li> <li>• 1500 символов;</li> <li>• 2000 символов.</li> </ul> <p>Для каждой входной последовательности измеряется время работы функции.</p>		
Косвенные входный данные			инициализированная предобученная модель векторизации данных: BERT от DeepPavlov.		
Ожидаемый результат			время работы функции ограничено квадратичной функцией		

№ ПТ.НТ	№ ТЕСТА	№ ФТ	2	ПТ.НТ2.1	ФТ3.2, ФТ3.3
---------	---------	------	---	----------	--------------

Цель теста	Оценить скорость кластеризации пользователей по группам в зависимости от их(пользователей) объёма.
Тип тестирования	негативный
Объект тестирования	Функция <b>cluster_users</b>
Входные данные	Последовательно функции подаются выборки размером: <ul style="list-style-type: none"> <li>• 250 элементов;</li> <li>• 500 элементов;</li> <li>• 750 элементов;</li> <li>• 1000 элементов.</li> </ul> Для каждой входной выборки замеряется время работы функции.
Косвенные входный данные	Отсутствует
Ожидаемый результат	время работы функции ограничено квадратичной функцией

### 3.5. Тестирование безопасности (ПТ.ТБ)

№ ПТ.ТБ	№ ТЕСТА	№ ФТ	1	ПТ.ТБ1.1	ФТ1.2
Цель теста			попытка авторизации в системе с неверным логин/паролем.		
Тип тестирования			позитивный		
Косвенные данные			Таблица в БД с зарегистрированными пользователями ( <b>tblUser</b> ) пуста		
Сценарий действий пользователя и ответных реакций системы для проверки			<ol style="list-style-type: none"> <li>1. Перейти на <b>Главную страницу</b> системы в состоянии неавторизованного пользователя;</li> <li>2. Отображается <b>Главная страница</b>;</li> <li>3. Нажать на кнопку <b>“sign in”</b>;</li> <li>4. Отображается форма авторизации в системе</li> <li>5. Ввод логина <b>“login”</b> и пароля <b>“password”</b> в соответствующие поля ввода и нажать кнопку <b>“sign in”</b></li> <li>6. Отображается предупреждающее</li> </ol>		



	сообщение о неудачной попытке авторизации.
--	--

№ ПТ.ТБ	№ ТЕСТА	№ ФТ	2	ПТ.ТБ2.1	ФТ1.2
Цель теста			Взаимодействие с системой в авторизованном состоянии через несколько браузерных окон и последовательный выход		
Тип тестирования			негативный		
Косвенные данные			Таблица в БД с данными зарегистрированных пользователей ( <b>tblUser</b> ) содержит следующую запись: <ul style="list-style-type: none"> <li>• user_id: 1</li> <li>• login: qwerty</li> <li>• password: 123456</li> </ul>		
Сценарий действий пользователя и ответных реакций системы для проверки			<ol style="list-style-type: none"> <li>1. Открыть два окна браузера и перейти на <b>Главную страницу</b> системы в одном из них.</li> <li>2. Отображается <b>Главная страница</b> в первом окне браузера;</li> <li>3. Авторизоваться в системе с логином: qwerty и паролем: 123456.</li> <li>4. После успешной авторизации отображается личный профиль пользователя.</li> <li>5. Скопировать прямую ссылку на профиль и открыть его во втором окне браузера.</li> <li>6. Отображается личный профиль пользователя во втором окне.</li> <li>7. Нажать на кнопку <b>"sign out"</b> во втором окне браузера.</li> <li>8. Отобразится <b>Главная страница</b> системы и пользователь перейдёт в неавторизованное состояние.</li> <li>9. Нажать на кнопку перезагрузки страница в первом окне браузера.</li> <li>10. Отобразится <b>Главная страница</b> системы.</li> </ol>		

№ ПТ.ТБ	№ ТЕСТА	№ ФТ	3	ПТ.ТБ3.1	ФТ1.2
Цель теста			Попытка перехода по прямой ссылке на страницу личного профиля без авторизации в системе.		
Тип тестирования			негативный		
Сценарий действий пользователя и ответных реакций системы для проверки			<ol style="list-style-type: none"> <li>1. Пользователь находится в авторизованном состоянии на странице <b>Личного профиля</b>.</li> <li>2. Перейти по прямой ссылке на <b>Главную страницу</b> системы.</li> <li>3. Система перенаправит пользователя в его <b>Личный профиль</b>.</li> </ol>		

#### 4. Трассируемость функциональных требований в тестах (ТФТ)

Цвет	Вид теста
	позитивный
	негативный

##### 4.1. Блочное тестирование (ТФТ.БТ)

ТФТ.БТ	№ ФТ									
№ ТЕСТА	1.1	1.2	2.1	2.2	3.1	3.2	3.3	3.4	3.5	3.6
ПТ.БТ1.1										
ПТ.БТ2.1										
ПТ.БТ3.1										

ПТ.БТ3.2	■									
ПТ.БТ3.3	■									
ПТ.БТ3.4	■									
ПТ.БТ4.1		■								
ПТ.БТ4.2		■								
ПТ.БТ5.1			■							
ПТ.БТ5.2			■							
ПТ.БТ6.1					■	■	■			
ПТ.БТ7.1								■		
ПТ.БТ7.2								■		
ПТ.БТ8.1									■	
ПТ.БТ9.1										■
ПТ.БТ9.2										■
ПТ.БТ10.1			■	■				■	■	
ПТ.БТ11.1	■	■								

#### 4.2. Интеграционное тестирование (ТФТ.ИТ)

ТФТ.ИТ	№ ФТ									
№ ТЕСТА	1.1	1.2	2.1	2.2	3.1	3.2	3.3	3.4	3.5	3.6
ПТ.ИТ1.1	■									
ПТ.ИТ1.2	■									
ПТ.ИТ1.3	■									
ПТ.ИТ1.4	■									
ПТ.ИТ2.1		■								

ПТ.ИТ2.2		■								
ПТ.ИТ2.3		■								
ПТ.ИТ2.4		■								
ПТ.ИТ3.1			■		■					
ПТ.ИТ3.2			■							
ПТ.ИТ3.3			■							
ПТ.ИТ3.4			■							
ПТ.ИТ4.1							■			
ПТ.ИТ4.2							■			
ПТ.ИТ5.1								■		
ПТ.ИТ5.2								■		
ПТ.ИТ5.3								■		
ПТ.ИТ6.1				■						■
ПТ.ИТ6.2										■

#### 4.3. Аттестационное тестирование (ТФТ.АТ)

ТФТ.АТ	№ ФТ									
№ ТЕСТА	1.1	1.2	2.1	2.2	3.1	3.2	3.3	3.4	3.5	3.6
ПТ.АТ1.1			■							
ПТ.АТ2.1								■	■	
ПТ.АТ3.1										■

#### 4.4. Нагрузочное тестирование (ТФТ.НТ)

<b>ТФТ.НТ</b>	<b>№ ФТ</b>									
<b>№ ТЕСТА</b>	<b>1.1</b>	<b>1.2</b>	<b>2.1</b>	<b>2.2</b>	<b>3.1</b>	<b>3.2</b>	<b>3.3</b>	<b>3.4</b>	<b>3.5</b>	<b>3.6</b>
<b>ПТ.НТ1 .1</b>										
<b>ПТ.НТ2 .1</b>										

#### 4.5. Тестирование безопасности (ТФТ.ТБ)

<b>ТФТ.ТБ</b>	<b>№ ФТ</b>									
<b>№ ТЕСТА</b>	<b>1.1</b>	<b>1.2</b>	<b>2.1</b>	<b>2.2</b>	<b>3.1</b>	<b>3.2</b>	<b>3.3</b>	<b>3.4</b>	<b>3.5</b>	<b>3.6</b>
<b>ПТ.ТБ1. 1</b>										
<b>ПТ.ТБ2. 1</b>										
<b>ПТ.ТБ3. 1</b>										

## 5. Журнал тестирования (ЖТ)

### 5.1. Блочное тестирование (ЖТ.БТ)

№ ЖТ.БТ	№ ТЕСТА	Дата	Результат	№ отчёта об ошибке
1	ПТ.БТ1.1	15.12.2022	провален	ЖО.БТ1
2	ПТ.БТ2.1	15.12.2022	провален	ЖО.БТ2
3	ПТ.БТ3.1	15.12.2022	пройден	—
4	ПТ.БТ3.2	15.12.2022	пройден	—
5	ПТ.БТ3.3	15.12.2022	пройден	—
6	ПТ.БТ3.4	15.12.2022	пройден	—
7	ПТ.БТ4.1	15.12.2022	пройден	—
8	ПТ.БТ4.2	15.12.2022	пройден	—
9	ПТ.БТ5.1	15.12.2022	пройден	—
10	ПТ.БТ5.2	15.12.2022	пройден	—
11	ПТ.БТ6.1	15.12.2022	пройден	—
12	ПТ.БТ7.1	15.12.2022	пройден	—
13	ПТ.БТ7.2	15.12.2022	пройден	—
14	ПТ.БТ8.1	15.12.2022	пройден	—

15	ПТ.БТ9.1	15.12.2022	пройден	—
16	ПТ.БТ9.2	15.12.2022	пройден	—
17	ПТ.БТ10.1	15.12.2022	пройден	—
18	ПТ.БТ11.1	15.12.2022	пройден	—

## 5.2. Интеграционное тестирование (ЖТ.ИТ)

№ ЖТ.ИТ	№ ТЕСТА	Дата	Результат	№ отчёта об ошибке
1	ПТ.ИТ1.1	15.12.2022	пройден	—
2	ПТ.ИТ1.2	15.12.2022	пройден	—
3	ПТ.ИТ1.3	15.12.2022	пройден	—
4	ПТ.ИТ1.4	15.12.2022	пройден	—
5	ПТ.ИТ2.1	15.12.2022	пройден	—
6	ПТ.ИТ2.2	15.12.2022	пройден	—
7	ПТ.ИТ2.3	15.12.2022	пройден	—
8	ПТ.ИТ2.4	15.12.2022	пройден	—
9	ПТ.ИТ3.1	15.12.2022	пройден	—
10	ПТ.ИТ3.2	15.12.2022	пройден	—

11	ПТ.ИТ3.3	15.12.2022	пройден	—
12	ПТ.ИТ3.4	15.12.2022	пройден	—
13	ПТ.ИТ4.1	15.12.2022	пройден	—
14	ПТ.ИТ4.2	15.12.2022	пройден	—
15	ПТ.ИТ5.1	15.12.2022	пройден	—
16	ПТ.ИТ5.2	15.12.2022	пройден	—
17	ПТ.ИТ5.3	15.12.2022	пройден	—
18	ПТ.ИТ6.1	15.12.2022	пройден	—
19	ПТ.ИТ6.2	15.12.2022	пройден	—

### 5.3. Аттестационное тестирование (ЖТ.АТ)

№ ЖТ.АТ	№ ТЕСТА	Дата	Результат	№ отчёта об ошибке
1	ПТ.АТ1.1	15.12.2022	пройден	—
2	ПТ.АТ2.1	15.12.2022	пройден	—
3	ПТ.АТ3.1	15.12.2022	пройден	—

### 5.4. Нагрузочное тестирование (ЖТ.НТ)



№ ЖТ.НТ	№ ТЕСТА	Дата	Результат	№ отчёта об ошибке
1	ПТ.НТ1.1	15.12.2022	Пройден	—
2	ПТ.НТ2.1	15.12.2022	Пройден	—

### 5.5. Тестирование безопасности (ЖТ.ТБ)

№ ЖТ.ТБ	№ ТЕСТА	Дата	Результат	№ отчёта об ошибке
1	ПТ.ТБ1.1	15.12.2022	Пройден	—
2	ПТ.ТБ2.1	15.12.2022	Пройден	—
3	ПТ.ТБ3.1	15.12.2022	Провален	ЖО.ТБ1

## 6. Журнал найденных ошибок (ЖО)

### 6.1. Блочное тестирование (ЖО.БТ)

№ ЖО.БТ	№ ТЕСТА	1	ПТ.БТ1.1
Дата составления отчёта		15.12.2022	
Описание		проверить поведение функции в случае векторизации пустого текстового поля	
Ожидаемый результат		Завершение выполнения функции без выпадения исключений	
Фактический результат		Выпадение исключения RuntimeError: Expected tensor for argument #1 'indices' to have one of the following scalar types: Long, Int; but got torch.FloatTensor instead (while checking arguments for embedding)	
Рекомендации		Сделать проверку на пустое текстовое значение	
Статус		Решено	

№ ЖО.БТ	№ ТЕСТА	2	ПТ.БТ2.1
Дата составления отчёта		15.12.2022	
Описание		проверить поведение функции в случае передачи неравного количества парных входных аргументов	
Ожидаемый результат		False	
Фактический результат		Выпадение исключения IndexError	
Рекомендации		Сделать проверку на равное количество парных аргументов	

Статус	Решено
--------	--------

## 6.2. Интеграционное тестирование (ЖО.ИТ)

Отсутствует.

## 6.3. Аттестационное тестирование (ЖО.АТ)

Отсутствует.

## 6.4. Нагрузочное тестирование (ЖО.НТ)

Отсутствует.

## 6.5. Тестирование безопасности (ЖО.ТБ)

№ ЖО.ТБ	№ ТЕСТА	1	ПТ.ТБ3.1
Дата составления отчёта		15.12.2022	
Описание		Попытка перехода по прямой ссылке на страницу личного профиля без авторизации в системе.	
Ожидаемый сценарий		<ul style="list-style-type: none"> <li>4. Пользователь находится в авторизованном состоянии на странице <b>Личного профиля</b>.</li> <li>5. Перейти по прямой ссылке на <b>Главную страницу</b> системы.</li> <li>6. Система перенаправит пользователя в его <b>Личный профиль</b>.</li> </ul>	
Фактический результат		Система отображает <b>Главную страницу</b> системы, когда пользователь находится в авторизованном состоянии.	
Рекомендации		Добавить обработчик перенаправления в зависимости от состояния авторизации пользователя в системе	

<b>№ ЖО.ТБ</b>	<b>№ ТЕСТА</b>	<b>1</b>	<b>ПТ.ТБ3.1</b>
<b>Статус</b>		Не решено	

## 7. Результаты

### Покрывание кода тестами:

Расчет тестового покрытия относительно исполняемого кода системы проводится по формуле:

$$T_{cov} = \frac{L_{tc}}{L_{code}} \times 100\% , \text{ где}$$

$T_{cov}$  - тестовое покрытие,

$L_{tc}$  - количество строк кода, покрытых тестами,

$L_{code}$  - общее количество строк кода.

№ тестируемого модуля в архитектуре	Строк кода					
	Фактически	Покрыто тестированием				
		Блочное	Интеграционное	Аттестационное	Нагрузочное	Безопасность
3	564	—	—	35%	—	15%
4	231	65%	82%	51%	—	6%
6	57	32%	68%	—	68%	—

### Статистика тестирования:

Тип теста	Кол-во тестов в плане	Вид теста		Проведённые тесты			
		-	+	Успех	Провал	Не реализовано	Не выполнено
Блочные	18	10	8	16	2	0	0
Интеграционные	19	11	8	19	0	0	3
Аттестационные	3	0	3	3	0	0	3
Нагрузочные	2	2	0	2	0	0	0

Безопасность	3	2	1	2	1	0	0
--------------	---	---	---	---	---	---	---

### Выводы:

В рамках дисциплины “Верификация программного обеспечения” была протестирована часть web-сервиса “Поиска людей со схожими интересами на основании портфолио”.

## 8. Приложения

### 8.1. Косвенные входные данные авторизация/регистрация

user_id	login	password
2	“qwerty”	“0123456”

### 8.2. Косвенные входные данные Цифровой профиль

vect_id	user_id	vector
1	2	[1,1]
2	1	[-1,1]
3	3	[8, 40]
4	4	[7, 35]

### 8.3. Косвенные входные данные Чат-группы

group_id	user_id	group_num
1	2	1

2	1	1
3	3	2
4	4	2

#### 8.4. Косвенные входные данные Личный профиль

info_id	user_id	name	description
1	1	"Anna"	"i like ride a bike"
2	2	"Ben"	"I like to swim in ocean"

#### 8.5. Косвенные входные данные Сессионные ключи

session_id	user_id	key
1	1	"test_session_key"