

# Отчёт первого этапа разработки приложения БД.

## Системный анализ предметной области

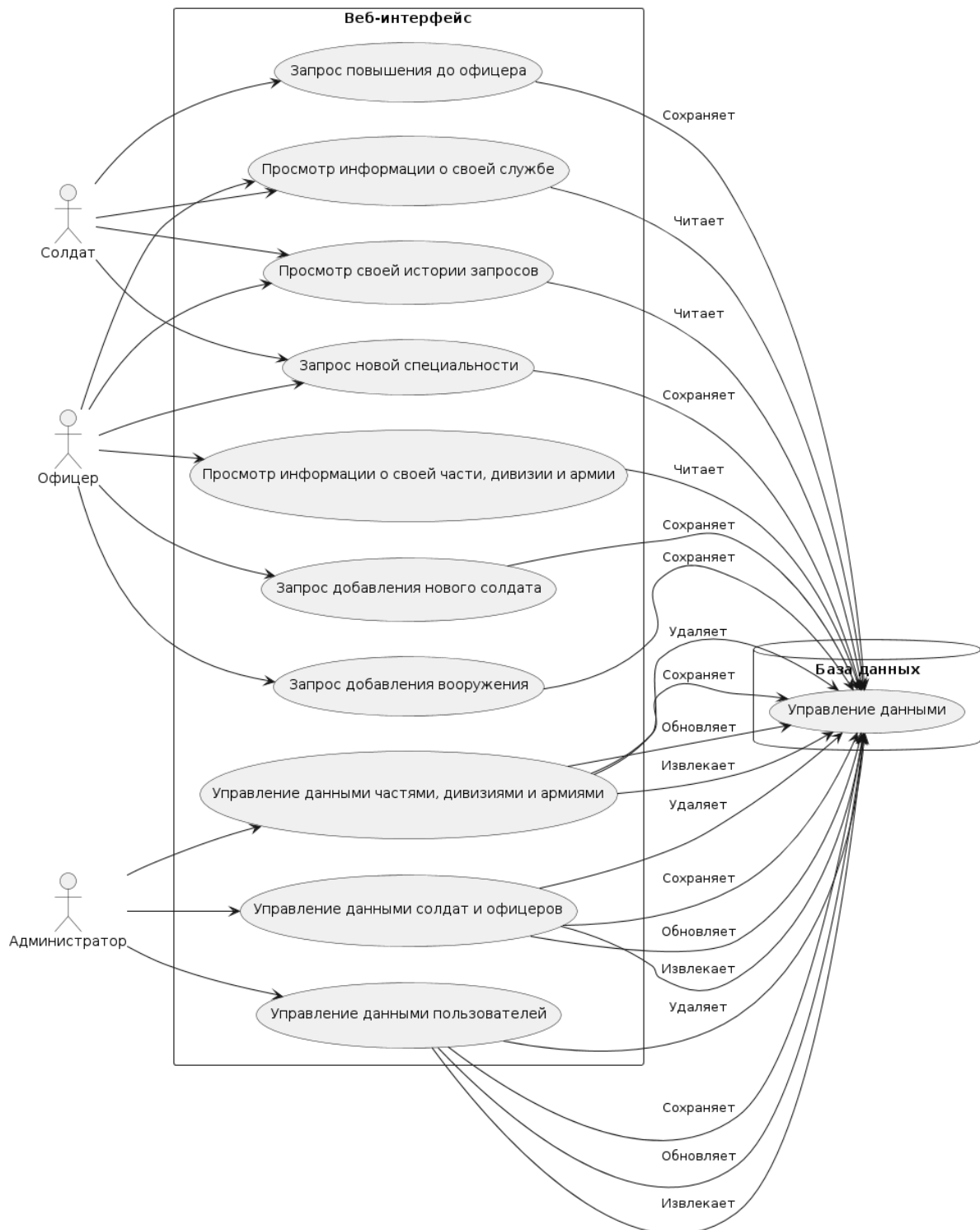
В качестве проекта был выбран вариант “информационная система военного округа” (4.7)

1. Назначение приложения: Создание информационной системы для управления военными частями и их данными. Пользователями будут высокопоставленные военные чины, администраторы, ответственные за ведение базы данных военных частей, а также рядовые военнослужащие.
2. Бизнес-процессы системы:
  - Управление структурой военных частей, дивизий и армий.
  - Назначение и учет военнослужащих.
  - Управление вооружением.
  - Учет и планирование инфраструктуры военных частей.
  - Анализ частей по ее специалистам, дивизий по их местоположение и другое.
3. Бизнес-процессы по типам пользователей:
  - Высокопоставленные военные чины: руководство и контроль над деятельностью военных подразделений, формирование запросов на изменение воинского состава и вооружения, анализ состояния частей.
  - Администраторы: ведение базы данных, внесение изменений в структуру подразделений, военнослужащих и их пользователей, обработка запросов от других пользователей.
  - Рядовые военнослужащие: получение информации о своей службе, отправление запросов для ее актуализации.
4. Информационные потоки и запросы:
  - Ввод данных о структуре военных частей, персонале и вооружении.
  - Запросы на выборку данных о частях, дивизиях, армиях, персонале и вооружении для анализа и принятия управленческих решений.
  - Получение персональной информации о своей службе.
5. Основные информационные объекты:
  - Военные части, дивизии, армии.
  - Персонал (офицерский и рядовой состав).
  - Вооружение.
  - Места дислокации.
  - Военные специальности.

6. Ограничения исходной задачей:

- Необходимость соблюдения конфиденциальности и безопасности данных.
- Точность и актуальность информации о структуре частей и их составе.
- Необходимость предоставления доступа к определенным данным только определенным пользователям в соответствии с их полномочиями.
- Возможная необходимость интеграции с другими информационными системами (например, системами управления боевой техникой).

# Диаграмма прецедентов



Здесь представлены основные действия для каждой из трёх ролей. Роли здесь следующие:

1. Administrator - администратор БД
2. Officer - офицер
3. User - рядовой военнослужащий

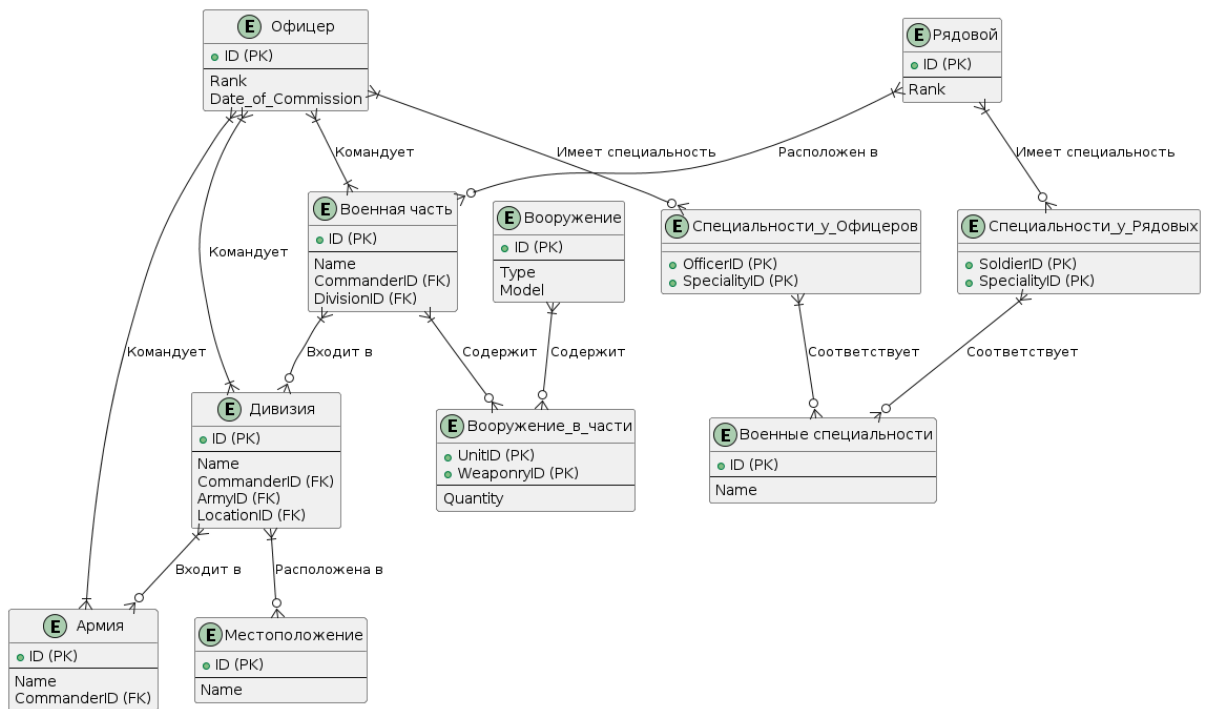
Взаимодействие происходит через веб-интерфейс приложения. Каждая из ролей имеет доступ к определённой её части. Стоит дополнить, что администратор также может взаимодействовать с БД напрямую через консоль. Это будет указано в контекстной диаграмме.

## Контекстная диаграмма



Прим.: здесь User объединяет роли рядового и офицера.

## Диаграмма “сущность-связь”



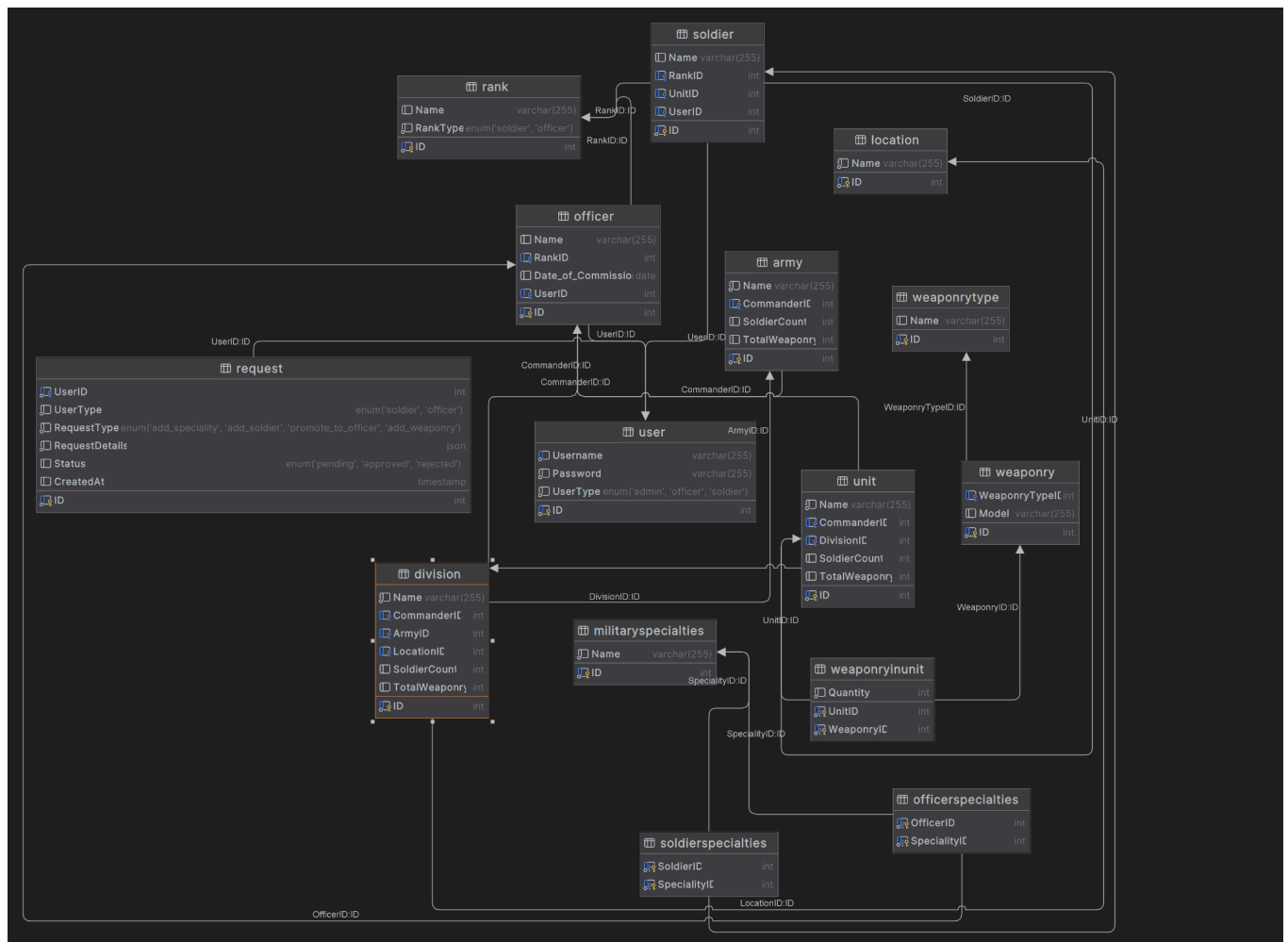
Было учтено, что:

1. У военнослужащих может быть несколько специальностей
2. Один и тот же тип вооружения может быть в нескольких частях. Также указывается их количество в каждой части
3. У дивизии есть своё местоположение
4. Офицер может быть командиром как части, так дивизии или армии

Также есть допущения и упрощения:

1. Отсутствуют некоторые формирования (роты, бригады и др.)
2. Отдельная часть может иметь специфическое местоположение
3. У многих типов вооружений могут быть атрибуты, которых нет у других

# Даталогическая схема



Ссылка на изображение:

<https://drive.google.com/file/d/1fTdWPkAryDAWruxTOfhUO0V6LCAH1C0b/view?usp=sharing>

Здесь представлены все таблицы БД с их конкретными атрибутами и связями.

## Приложение БД

### Архитектура приложения

Архитектура приложения представлена ниже:



Пользователь при помощи html-интерфейса работает с сайтом, браузер отправляет запрос на веб-сервер, а тот при необходимости запрашивает данные из БД.

## Структура проекта

```
db_app/
├── templates/
│   ├── login.html
│   ├── admin_dashboard.html
│   ├── soldier_page.html
│   └── ...
├── setup_db/
│   ├── migrations/
│   │   ├── migration_0001.sql
│   │   └── migration_0002.sql
│   ├── create_db.sql
│   ├── fill_tables.sql
│   └── ...
├── css/
│   └── styles.css
└── images/
```

```
|  └─ background.jpg
|
|  └─ requirements.txt
|  └─ config.json
|  └─ app.py
|  └─ setup_database.py
```

Пояснение:

- templates - папка с html-шаблонами для отображения страниц сайта
- setup\_db - папка, в которой находятся все необходимые sql-скрипты для создания базы данных, таблиц и заполнения их данными
- css - папка с файлами стилей для настройки отображения страниц сайта
- images - папка с изображениями. Применяются в процессе задания стилей
- requirements.txt - файл со всеми зависимостями в рамках Python
- config.json - файл конфигурации приложения. Там задаются параметры подключения к БД и адрес, где будет работать сайт
- app.py - файл с кодом веб-приложения
- setup\_database.py - скрипт, который создаёт базу данных и заполняет ее данными

Подробное описание частей проекта будет изложено далее

## Триггеры БД

Основная суть триггеров есть автоматическое выполнение необходимых действий при каком-либо событии. В данной системе благодаря триггерам поддерживаются актуальные значения в таких важных атрибутах, как количество солдат и вооружения. Они позволяют не пересчитывать заново эти значения при каждом запросе, что повышает производительность. Триггеры и их описание представлены ниже:

1. IncreaseSoldierCount
  - Событие: AFTER INSERT ON Soldier
  - Описание: Этот триггер увеличивает счётчик солдат в таблицах Unit, Division и Army после добавления нового солдата. Сначала обновляется количество солдат в части (Unit), затем в дивизии (Division), к которой относится часть, и, наконец, в армии (Army), к которой относится дивизия.
2. DecreaseSoldierCount
  - Событие: AFTER DELETE ON Soldier
  - Описание: Этот триггер уменьшает счётчик солдат в таблицах Unit, Division и Army после удаления солдата. Он выполняет обратные действия по сравнению с триггером IncreaseSoldierCount, уменьшая счётчики сначала в части, затем в дивизии и, наконец, в армии.
3. AfterInsertWeaponry
  - Событие: AFTER INSERT ON WeaponryInUnit
  - Описание: Этот триггер обновляет количество вооружения в таблицах Unit, Division и Army после добавления новой записи в таблицу WeaponryInUnit. Он сначала обновляет количество вооружения в части, затем в дивизии, к которой относится часть, и в армии, к которой относится дивизия.



#### 4. AfterUpdateWeaponry

- Событие: AFTER UPDATE ON WeaponryInUnit
- Описание: Этот триггер обновляет количество вооружения в таблицах Unit, Division и Army после обновления записи в таблице WeaponryInUnit. Он вычисляет разницу между старым и новым количеством вооружения и обновляет соответствующие счётчики в части, дивизии и армии.

#### 5. OnUpdateSoldierUnit

- Событие: BEFORE UPDATE ON Soldier
- Описание: Этот триггер обрабатывает изменения в поле UnitID солдата. Он корректирует количество солдат в старой и новой части (Unit), а также обновляет количество солдат в соответствующих дивизиях (Division) и армиях (Army). Триггер проверяет, изменились ли дивизия и армия, и если да, то обновляет соответствующие счётчики.

## Хранимые процедуры БД

Хранимые процедуры используются в данном случае для выноса больших и используемых в нескольких местах кода запросов в отдельные объекты в виде процедур, что, в том числе, делает код приложения более читаемым. Самые примечательные процедуры описаны ниже:

#### 1. Promote\_Soldier

- Описание: Эта процедура повышает солдата до офицера. Эта процедура состоит из нескольких запросов, поэтому они исполняются внутри отдельной транзакции, дабы в случае ошибки в процедуре база данных осталась в состоянии до этой процедуры.
- Параметры:
  1. p\_soldier\_id INT — ID солдата, который будет повышен.
  2. p\_rank\_id INT — ID ранга, который будет присвоен новому офицеру.
- Шаги:
  1. Добавляет солдата в таблицу Officer с заданным рангом и текущей датой назначения (Date\_of\_Commission).
  2. Переносит все специальности солдата из SoldierSpecialties в OfficerSpecialties.
  3. Меняет тип пользователя на officer в таблице User.
  4. Удаляет солдата из таблицы Soldier

#### 2. Request\_Add\_Weaponry

- Описание: Эта процедура создаёт запрос на добавление нового вооружения в часть.
- Параметры:
  1. p\_user\_id INT — ID пользователя, создающего запрос.
  2. p\_weaponry\_id INT — ID вооружения.

3. p\_unit\_id INT — ID военной части.
  4. p\_quantity INT — Количество вооружения.
- Шаги:
    1. Вставляет запись в таблицу Request с типом запроса request\_add\_weaponry и статусом pending.
3. CheckOfficerCommandingSoldier
- Описание: процедура CheckOfficerCommandingSoldier предназначена для проверки, является ли указанный пользователь командиром солдата или находится ли он в командной цепочке солдата. Процедура проверяет, командует ли пользователь (офицер) данной военной частью, дивизией или армией, к которой относится указанный солдат.
  - Параметры:
    1. soldier\_id INT: Идентификатор солдата, о котором запрашивается информация.
    2. user\_id INT: Идентификатор пользователя (офицера), который проверяется на предмет командования солдатом.
  - Шаги:
    - Выбор данных солдата и его командиров:
      - s.Name AS SoldierName: Имя солдата.
      - r.Name AS RankName: Название ранга солдата.
      - u.Name AS UnitName: Название части, к которой приписан солдат.
      - o1.Name AS UnitCommander: Имя командира части.
    - Соединение таблиц:
      - JOIN Rank r ON s.RankID = r.ID: Присоединение таблицы Rank для получения информации о ранге солдата.
      - JOIN Unit u ON s.UnitID = u.ID: Присоединение таблицы Unit для получения информации о части солдата.
      - JOIN Division d ON d.ID = u.DivisionID: Присоединение таблицы Division для получения информации о дивизии, к которой относится часть.
      - JOIN Army a ON a.ID = d.ArmyID: Присоединение таблицы Army для получения информации об армии, к которой относится дивизия.
      - JOIN Officer o1 ON u.CommanderID = o1.ID: Присоединение таблицы Officer для получения информации о командире части.
      - JOIN Officer o2 ON d.CommanderID = o2.ID: Присоединение таблицы Officer для получения информации о командире дивизии.
      - JOIN Officer o3 ON a.CommanderID = o3.ID: Присоединение таблицы Officer для получения информации о командире армии.
    - Условие WHERE:
      - s.ID = soldier\_id: Условие выбора данных о солдате по его ID.
      - (o1.UserID = user\_id OR o2.UserID = user\_id OR o3.UserID = user\_id): Условие проверки, является ли пользователь командиром части, дивизии или армии, к которой относится солдат.

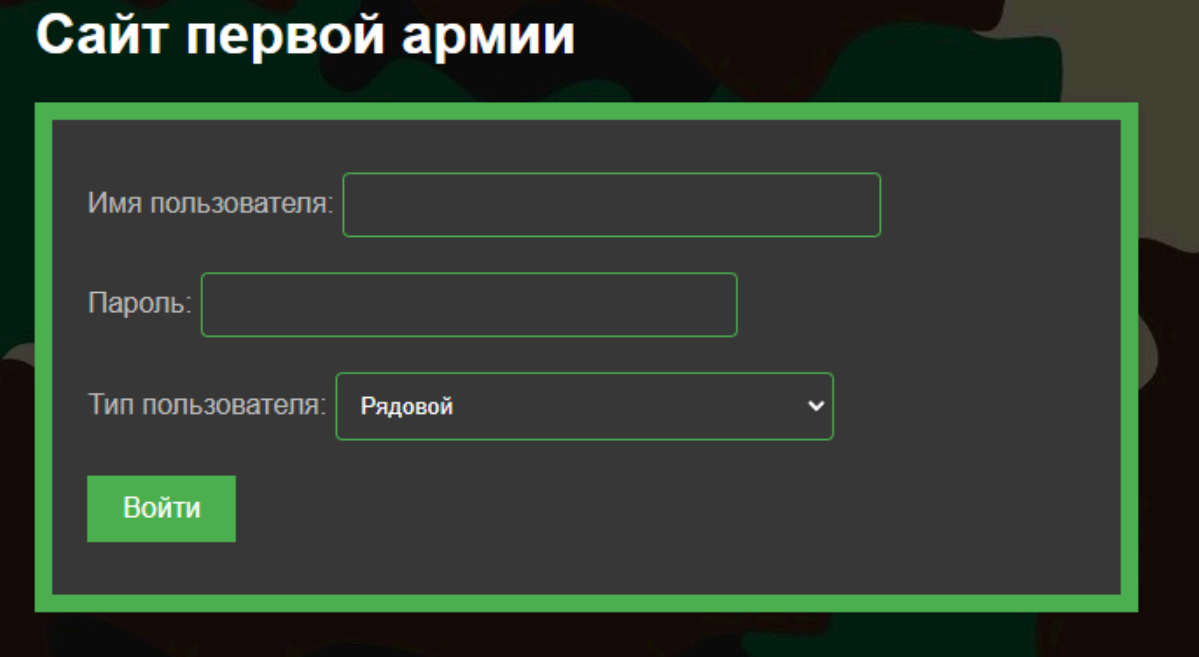
## Интерфейс приложения

В виду архитектуры приложения БД, пользователь не имеет прямого доступа к базе данных. Веб-приложение подключается к БД, имея к ней полный доступ. Пользователь же взаимодействует через браузер с приложением, которое решает: каким функционалом он может пользоваться - в зависимости от роли. Доступ контролируется при помощи пользовательских сессий. Данные о самих пользователях хранятся в таблице User: логин, хэш пароля и тип пользователя. Более высокоуровневая система ролей исключает риск повреждения базы данных (Тимофей, привет).

Также акцент был сделан на то, чтобы разрабатываемый сайт был структурно похож на те, которые встречаются в обычной жизни. Именно поэтому инициализация базы данных происходит посредством запуска отдельного скрипта на узле, где она будет находиться, а не в начальном окне сайта (оконного приложения).

Далее описана структура сайта:

1. Авторизация. Необходимо ввести логин, пароль и тип. При успешной авторизации происходит переадресация на главную страницу того типа пользователя, которому он соответствует.



The image shows a login interface for a website titled "Сайт первой армии" (Website of the first army). The interface is set against a dark background with a camouflage pattern. It features a central light gray box with a green border containing the login fields. The fields are labeled "Имя пользователя:" (Username:), "Пароль:" (Password:), and "Тип пользователя:" (User type:). The "Тип пользователя:" field is a dropdown menu currently showing "Рядовой" (Private) with a downward arrow. Below the fields is a green button labeled "Войти" (Login).

2. Интерфейс администратора
  - 2.1. Админ панель. Здесь представлена главная страница администратора, где есть ссылки для управления солдатами, офицерами, частями, армиями, пользователями, а также для рассмотрения запросов от других пользователей. Кроме того, есть возможность сменить пароль и выйти из аккаунта - этот функционал есть у всех типов пользователей.

# Админ-панель

- Просмотр солдат
- Просмотр офицеров
- Просмотр пользователей
- Добавить солдата
- Добавить офицера
- Добавить пользователя
- Просмотр частей
- Просмотр дивизий
- Просмотр армий
- Просмотр запросов

Сменить пароль  
Выйти

2.2. На этой странице происходит просмотр сведений о солдатах. Также есть возможность редактировать и удалять их. Страницы просмотра офицеров и пользователей схожи с этой.

Солдаты					
ID	Имя	Ранг	Часть	Пользователь	Действия
1	Смирнов Алексей Иванович	Рядовой	1-я часть	soldier1	Редактировать Удалить
2	Кузнецов Дмитрий Александрович	Ефрейтор	1-я часть	soldier2	Редактировать Удалить
3	Иванов Андрей Петрович	Сержант	1-я часть	soldier3	Редактировать Удалить
4	Соколов Сергей Алексеевич	Младший сержант	1-я часть	soldier4	Редактировать Удалить
5	Попов Александр Владимирович	Старший сержант	2-я часть	soldier5	Редактировать Удалить
6	Алексеев Денис Васильевич	Старшина	2-я часть	soldier6	Редактировать Удалить
7	Лобов Артем Викторович	Рядовой	2-я часть	soldier7	Редактировать Удалить
8	Семенов Владимир Андреевич	Ефрейтор	2-я часть	soldier8	Редактировать Удалить

2.2.1. Редактирование солдата. Данная страница представляет из себя форму, где можно изменить данные о солдате: имя, ранг, часть и пользователя. Форма отправляется в веб-приложение, где уже идет необходимое взаимодействие с базой данных. Формы редактирования офицера и пользователя аналогичны

### Редактировать солдата

Имя:

Смирнов Алексей Иванович

Ранг:

Рядовой

Часть:

1-я часть

Пользователь:

soldier1

Сохранить

- 2.3. Добавление пользователя. Страница представляет из себя форму для заполнения имени пользователя, его пароля и типа.

### Добавить пользователя

Имя пользователя:

Пароль:

Тип пользователя: Рядовой ▼

[Назад](#)

- 2.4. Страница просмотра частей. Здесь в виде таблицы отображается вся информация о частях военного округа. Для дивизий и армий всё аналогично.

### Части

Номер	Название части	Название дивизии	Название армии	Командир части	Действия
1	1-я часть	Первая дивизия	Первая армия	Солдат Солдатыч	<a href="#">Редактировать</a>
2	2-я часть	Вторая дивизия	Первая армия	Смирнов С.С.	<a href="#">Редактировать</a>
3	3-я часть	Первая дивизия	Первая армия	Морозов М.М.	<a href="#">Редактировать</a>
4	4-я часть	Вторая дивизия	Первая армия	Николаев Н.Н.	<a href="#">Редактировать</a>

[Назад](#)

- 2.4.1. Страница редактирования части. Здесь администратор имеет возможность изменить данные о какой-либо части.

### Редактировать часть

Название:

Командир: Солдат Солдатыч ▼

Дивизия: Первая дивизия ▼

[Назад](#)

- 2.5. Страница просмотра запросов. Администратор может просмотреть данные о запросе: его тип и детали - и на основе этого принять решение: принять или отклонить.

### Запросы пользователей

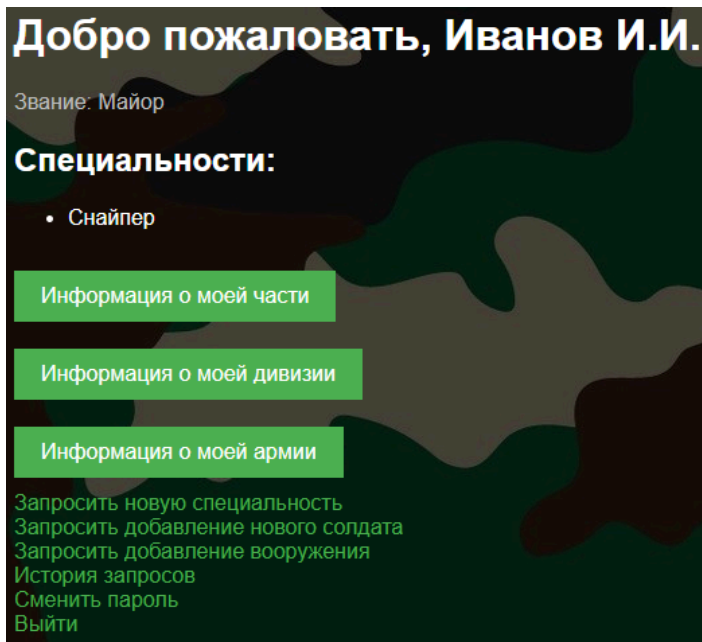
Имя пользователя	Тип пользователя	Тип запроса	Детали запроса	Действие
soldier1	soldier	add_specialty	{"specialty_id": "2"}	<input type="button" value="Одобрить"/> <input type="button" value="Отклонить"/>
officer1	officer	add_weaponry	{"unit_id": "4", "quantity": "4", "weapon_type": "1"}	<input type="button" value="Одобрить"/> <input type="button" value="Отклонить"/>

[Назад](#)

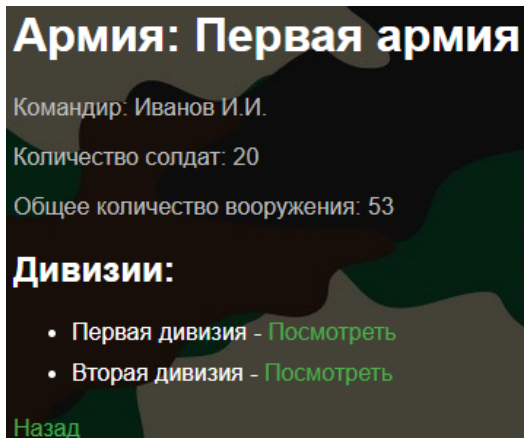
## 3. Интерфейс офицера

- 3.1. Главная страница. Здесь представлен интерфейс, обеспечивающий функционал офицера как пользователя приложения бд. Есть

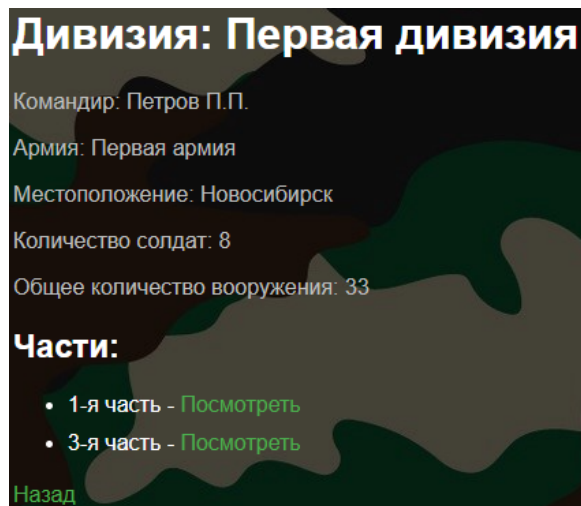
возможность посмотреть основную информацию о себе, своей части, дивизии или армии. Также есть возможность запросить добавить в базу данных свою новую специальность, солдата и оружие в часть. Есть ссылка на историю отправленных запросов, которая есть и у солдата.



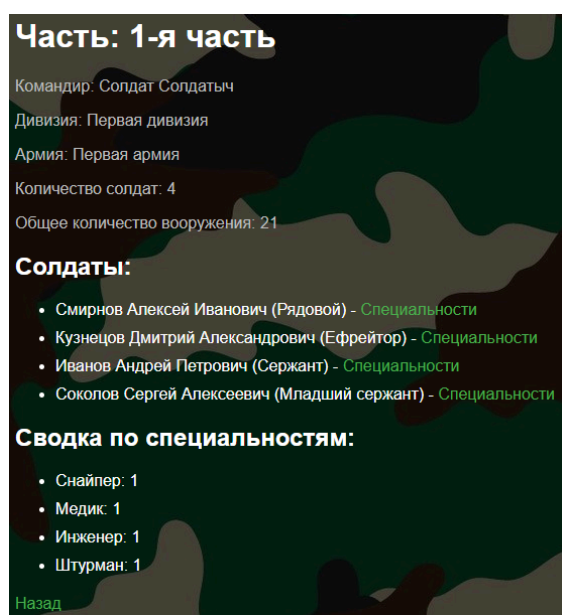
- 3.2. Страница просмотра армии. На этой страницу выводится вся информация об армии, которая непосредственно подчиняется этому офицеру. Если же ему не подчинена никакая армия, будет выведена ошибка. Аналогично с просмотром своей дивизии и части. Также есть ссылки, которые ведут на страницу просмотра дивизий, которые находятся в этой армии.



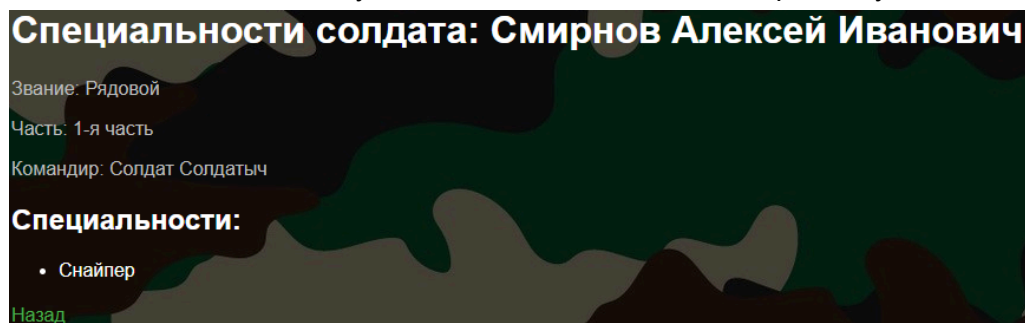
- 3.2.1. Страница просмотра дивизии. Отображаются сведения о дивизии. Примечательна здесь информация о ее местоположении. Есть возможность посмотреть части, которые находятся в составе этой дивизии.



- 3.2.2. Страница просмотра части. Здесь, помимо прочего, отображается список солдат в этой части и сводка по специальностям: какой специальностью сколько солдат обладает. Также здесь есть возможность просмотреть специальности каждого солдата из этой части.



- 3.2.3. Страница просмотра специальностей солдата. У офицера есть доступ только к тем солдатам, которые ему подчинены.



- 3.3. Страница запроса новой специальности. Здесь офицер выбирает специальность, которую он получил, чтобы зафиксировать ее в базе

данных. Администратор проверяет запрос и принимает решение. У солдата также есть доступ к этой форме.

### Запрос на добавление новой специальности

Выберите специальность:

Снайпер

▼

Отправить запрос

Назад

- 3.4. Страница запроса добавления солдата. Офицер может отправить этот запрос администратору, если сведения о новом солдате тому не поступили. Далее администратор вручную задает часть и пользователя новому солдату.

### Запрос на добавление нового солдата

Имя солдата:

Ранг солдата:

Рядовой

▼

Отправить запрос

- 3.5. Страница запрос добавления нового вооружения. При поступлении вооружения в часть офицер отправляет этот запрос, указывая всю необходимую информацию. Офицер может сформировать запрос только для той части, которая ему подчинена.

### Запрос на добавление нового вооружения

Тип вооружения:

АК-47

▼

Количество:

Часть:

1-я часть

▼

Отправить запрос

- 3.6. Страница истории запросов. Здесь офицер может просмотреть свои запросы: новые и уже рассмотренные. У солдата есть аналогичная страница.

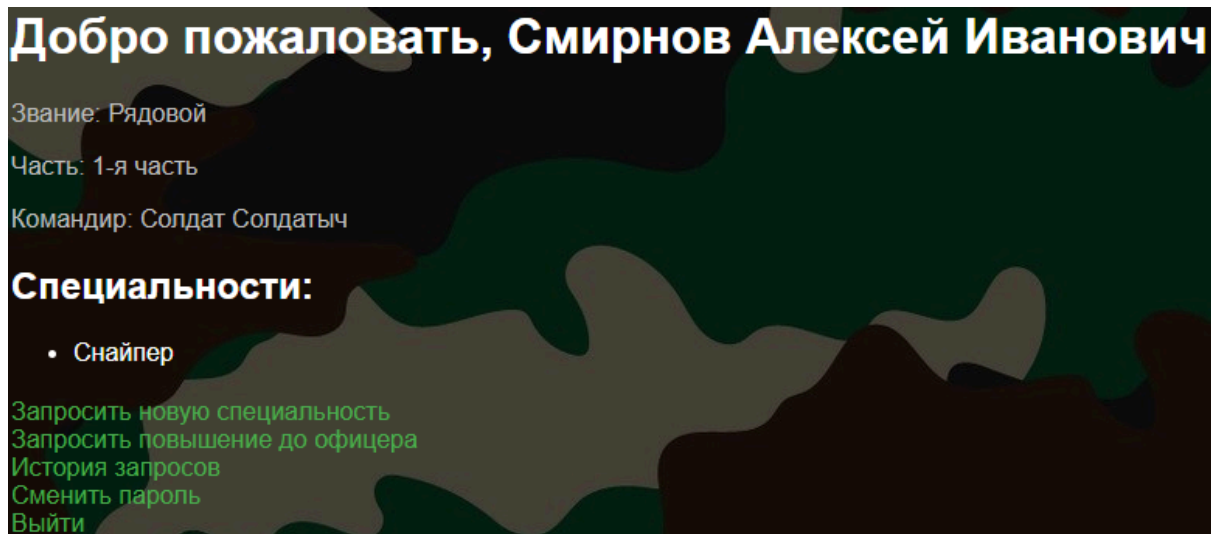
История ваших запросов			
Тип запроса	Детали запроса	Статус	Дата создания
add_weaponry	{"unit_id": "4", "quantity": "4", "weapon_type": "1"}	pending	2024-05-22 16:30:36

Назад

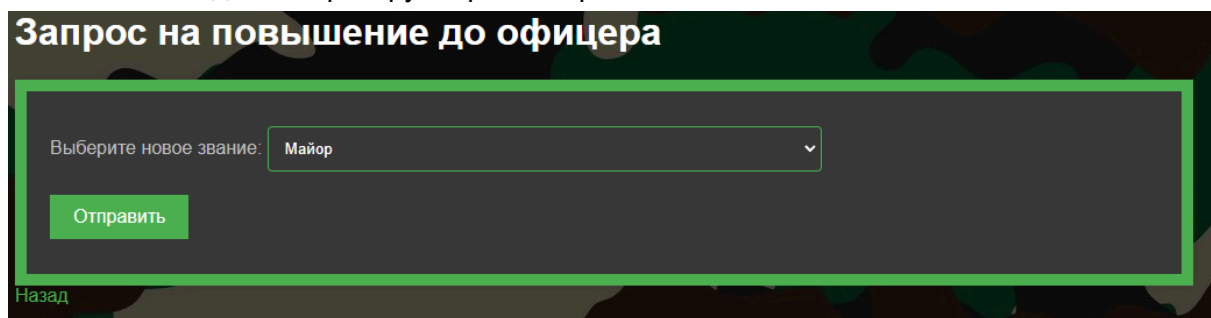


#### 4. Интерфейс солдата

- 4.1. Главная страница. На этой страница показываются сведения о службе данного солдата. Есть возможность запросить новую специальность и повышение до офицера.



- 4.2. Страница запроса новой специальности. Аналогична этой же странице у офицера.
- 4.3. Страница запроса повышение до офицера. Солдат выбирает офицерское звание, которое он получил, и отправляет запрос администратору на рассмотрение.



- 4.4. Страница истории запросов. Аналогична этой же странице у офицера. Таким образом, все предусмотренные бизнес-процессы покрыты удобным интерфейсом. Доступ пользователей ограничивается на стороне бекенда, что не допускает несанкционированный доступ, например, через консоль.

## Инструкция по запуску приложения

ВНИМАНИЕ: всё решение разрабатывалось под Linux, на Windows возникают ошибки в sql-скриптах (возможно, поможет выбор другого диалекта sql). Если у вас нет возможности запустить приложение на машине с ОС Linux, то это демонстрируется в скринкасте, где, помимо этого, показывается весь функционал сайта. Также вы можете сами протестировать сайт, перейдя по ссылке: <http://185.196.117.180:5002/login>

Для запуска приложения необходимо следующее ПО:

1. Python версии 3.6 или выше. Также необходимы следующие библиотеки:
  - a. Flask версии 3.0.3 и выше,
  - b. mysql-connector-python версии 8.4.0 и выше,
  - c. bcrypt версии 3.2.0 и выше.
2. MySQL сервер версии 5.7 или выше.

Инструкция по запуску веб-приложения:

1. Установка зависимостей. Убедитесь, что на вашем компьютере установлен Python. Установите все необходимые библиотеки. Можно воспользоваться командой `pip install -r requirements.txt`.
2. Запуск сервера базы данных. Убедитесь, что у вас установлен MySQL сервер. Запустите сервер базы данных, используя команду запуска, соответствующую вашей операционной системе.
3. Настройка файла `config.json`. В нём есть следующие значения:
  - a. `host` - адрес бд,
  - b. `user` - пользователь бд,
  - c. `password` - пароль для доступа к бд,
  - d. `database` - используемая база данных. Впишите туда "mysql", если создаете базу данных military впервые,
  - e. `secret_key` - ключ для функционирования сессий в Flask. Можно проигнорировать,
  - f. `app_host` - адрес, на котором запустится сайт,
  - g. `app_port` - порт, на котором запустится сайт.
4. Создание базы данных. Запустите скрипт `setup_database.py`. Он исполнит все sql-скрипты, необходимые для создания БД и заполнения её тестовыми данными.
5. Запуск веб-приложения. Перейдите в каталог с веб-приложением. Запустите веб-приложение, запустив главный файл приложения командой `python app.py`.
6. Доступ к приложению. После запуска приложения откройте ваш веб-браузер и перейдите по адресу `http://185.196.117.180:5000` (или другому адресу, если он был изменен).

Теперь вы можете использовать веб-приложение.

Для взаимодействия с сайтом вам обязательно потребуются данные пользователей. Они представлены ниже:

1. Администратор
  - a. Логин: admin
  - b. Пароль: admin
2. Солдат
  - a. Логин: soldier1,..., soldier20
  - b. Пароль: 1234
3. Офицер
  - a. Логин: officer1,..., officer7
  - b. Пароль: 1234