

Спроектировать и разработать систему регистрации и обработки пользовательских заявок.

Пользователь посредством системы может подавать заявки оператору на рассмотрение.

Оператор может просматривать пользовательские заявки и принимать или отклонять их. Администратор управляет правами доступа.

- Спроектировать и разработать back-приложение
- Спроектировать и разработать Базу данных (Liquibase)

Функции приложения

- Создать заявку (Заявка помимо прочих системных полей состоит из статуса и текстового обращения пользователя, номера телефона, имени)
- Отправить заявку оператору на рассмотрение
- Просмотреть список заявок с возможностью сортировки по дате создания в оба направления (как от самой старой к самой новой, так и наоборот) и пагинацией по 5 элементов, фильтрация по статусу
- Посмотреть заявку
- Принять заявку
- Отклонить заявку
- Просмотреть список пользователей
- Назначить права оператора
- Заявки могут дублироваться (это некоммерческий проект. важно увидеть только то, как вы реализуете функционал)

В системе предусмотрены **3 роли**:

- Пользователь
- Оператор
- Администратор

У пользователя системы может быть одновременно несколько ролей, например, «Оператор» и «Администратор».

У заявки пользователя предусмотрено **4 статуса**:

- черновик
- отправлено
- принято
- отклонено

Пользователь может

- логиниться в систему (выдача jwt и access токенов)
- выходить из системы /logout
- создавать заявки

- создавать черновики
- просматривать созданные им заявки с возможностью сортировки по дате создания в оба направления (как от самой старой к самой новой, так и наоборот) и пагинацией по 5 элементов
- редактировать созданные им заявки в статусе «черновик»
- отправлять заявки на рассмотрение оператору

Пользователь НЕ может:

- редактировать отправленные на рассмотрение заявки
- видеть заявки других пользователей
- принимать заявки
- отклонять заявки
- назначать права
- смотреть список пользователей

Оператор может

- логиниться в систему (выдача jwt и access токенов)
- выходить из системы /logout
- смотреть все отправленные на рассмотрение заявки с возможностью сортировки по дате создания в оба направления (как от самой старой к самой новой, так и наоборот) и пагинацией по 5 элементов. Должна быть фильтрация по имени. Просматривать отправленные заявки только конкретного пользователя по его имени/части имени (у пользователя, соответственно, должно быть поле name)
- смотреть заявку по id
- принимать заявки
- отклонять заявки

Оператор НЕ может

- создавать заявки
- просматривать заявки в статусе отличном от «отправлено»
- редактировать заявки
- назначать права

Администратор может

- логиниться в систему (выдача jwt и access токенов)
- выходить из системы /logout
- смотреть список пользователей
- смотреть заявки в статусе отправлено, принято, отклонено. Пагинация 5 элементов, сортировка по дате. Фильтрация по имени.
- назначать пользователям права оператора

Администратор НЕ может

- создавать заявки

- редактировать заявки
- принимать заявки
- отклонять заявки

Технические требования к приложению

- Java 11 или 17
 - Использовать архитектуру REST
 - Использовать Spring Boot
 - Использовать Spring Security
 - Использовать JPA
 - Использовать реляционную БД (PostgreSQL, MariaDB)
 - Миграцию осуществлять с помощью Liquibase
 - Создание пользователей и ролей не предусмотрено в этой системе.
- Подразумевается, что данные об учетных записях пользователей и роли уже есть в БД
- В описании ТЗ пропишите все креды (логин:пароль) всех созданных вами пользователей
 - Запуск с помощью встроенного сервера приложений Apache Tomcat в Spring Boot (Не Docker)
 - Приложение должно запускаться без каких-либо ошибок

Плюсом будет:

Проводить проверку номер телефона через сервис [Dadata](#). Не нужно делать для этого отдельный эндпоинт. Внимательно изучите АПИ dadata.

Отправлять запрос через **Spring Cloud Open Feign**, сохранять данные в формате: *код страны, код города, номер телефона* который придет в ответе. Номер телефона должен быть мобильный.