

# Visionary

## Minimum Viable Product (MVP)

Приложение для перерисовки фотографий под одну из заранее определённых стилистик

# Функциональные возможности



Обработка изображений  
с помощью нейросети



Лента работ  
пользователей

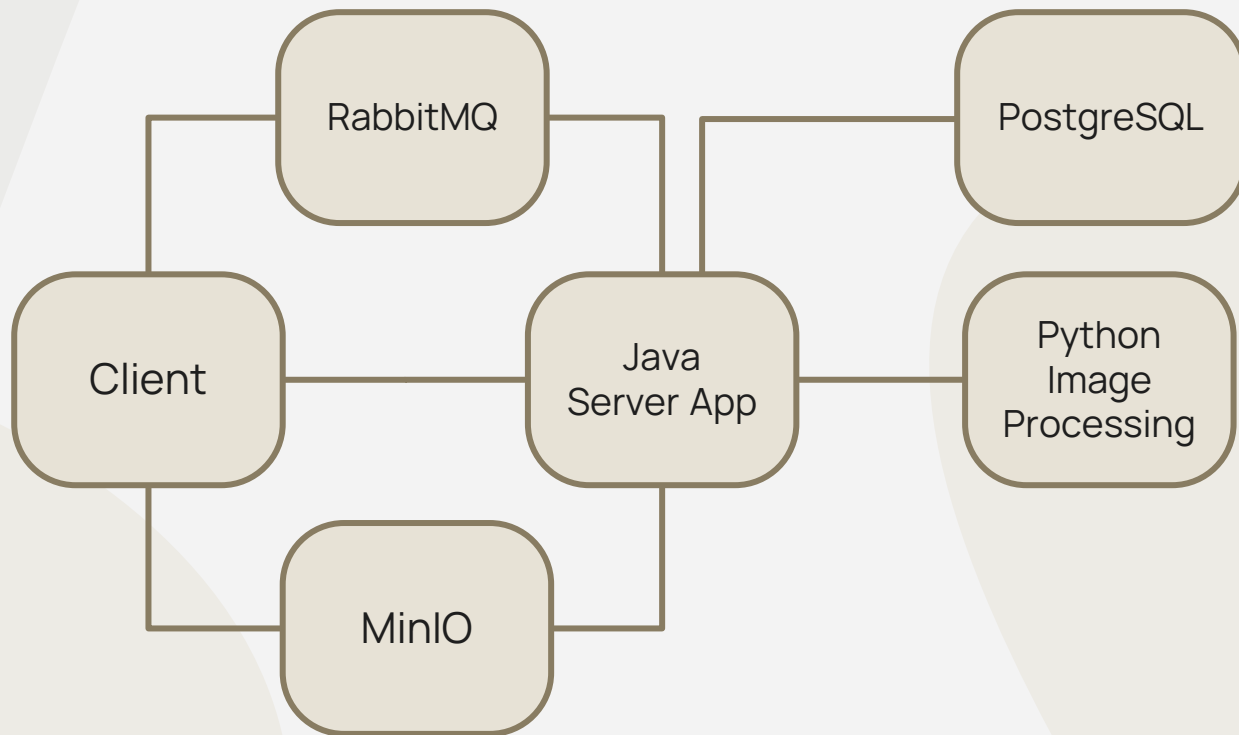


Комментирование  
работ



Отслеживание  
обновлений  
пользователей

# Архитектура проекта



# Стек технологий

Бэкенд:

- Java 19
- Spring
- MinIO
- PostgreSQL
- RabbitMQ

Обработка

изображений:

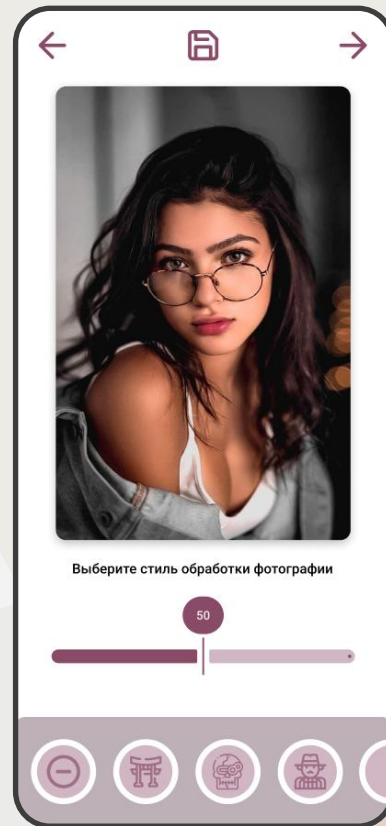
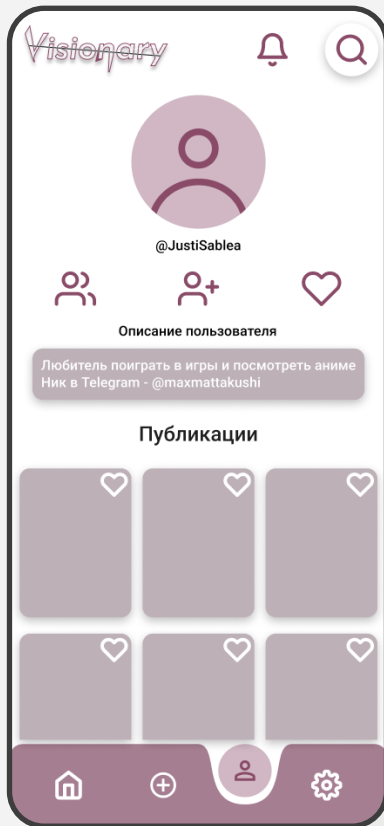
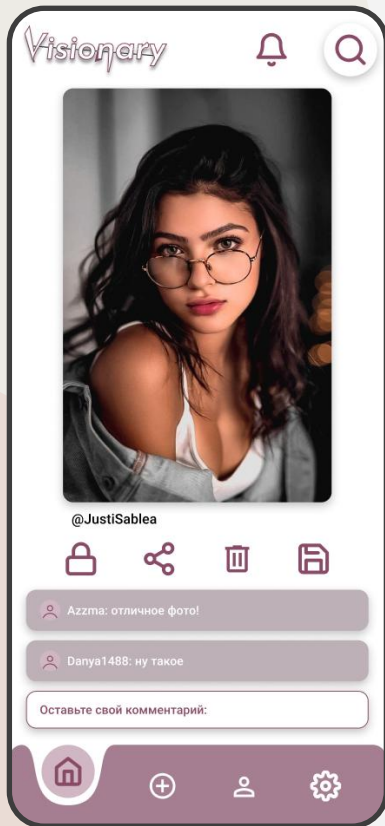
- Python 3.12
- FastAPI
- Diffusers

Клиент:

- Kotlin 1.9.23



# Интерфейс



# Интерфейс

- Экран профиля и редактирования с просьбой пройти регистрацию
- Переход к окну регистрации через настройки



Сценарий:  
Неавторизованный  
пользователь

# Интерфейс

- Проверка корректности введенных в поля регистрации данных
- Выход из аккаунта через экран настроек



Сценарий:  
Регистрация

# Интерфейс

## Возможности:

- Сделать фото и редактировать;
- Сохранить фото в галерею;
- Выбрать фото из галереи;
- Просмотреть выложенный пост.

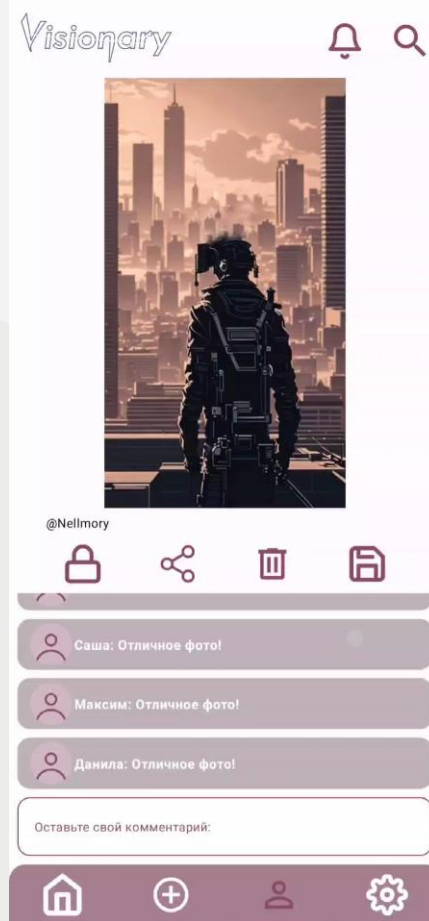


## Сценарий: Редактирование фото



# Интерфейс

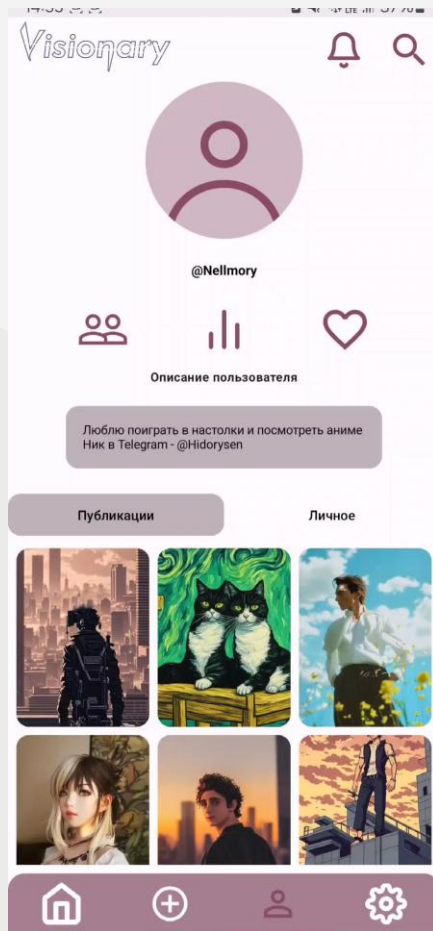
- Возможность оставить и просмотреть комментарии
- Интерфейс с инструментами для редактирования, удаления, сохранения в галерею поста и возможностью поделиться ссылкой



Сценарий:  
Комментирование и  
просмотр поста

# Интерфейс

- Всплывающие окна с информацией о подписках



Сценарий:  
Просмотр условий  
подписок

# API проекта

## image Операции, связанные с изображениями

**GET****/api/v1/image/{uuid}** Получить данные изображения по uuid**PUT****/api/v1/image/{uuid}** Обновить уровень доступа к изображению**DELETE****/api/v1/image/{uuid}** Удалить сохраненное изображение**POST****/api/v1/image/save** Сохранение обработанного изображения на сервере**GET****/api/v1/user/{userId}/images** Получить изображения другого пользователя**GET****/api/v1/user/images** Получить изображения текущего пользователя

# API проекта

## user Операции, связанные с пользователями

**GET****/api/v1/user/{userId}** Получить данные пользователя**PUT****/api/v1/user/{userId}** Обновить данные пользователя**DELETE****/api/v1/user/{userId}** Удалить аккаунт пользователя**GET****/api/v1/user/{userId}/images** Получить изображения другого пользователя**GET****/api/v1/user/images** Получить изображения текущего пользователя**DELETE****/api/v1/user/{userId}/avatar** Удалить аватар пользователя

# API проекта

## auth Операции, связанные с авторизацией, регистрацией пользователя



**POST** `/api/v1/auth/register` Регистрация нового пользователя по никнейму



**POST** `/api/v1/auth/refresh` Обновление access-токена



**POST** `/api/v1/auth/login` Аутентификация уже существующего пользователя



## processing the processing API

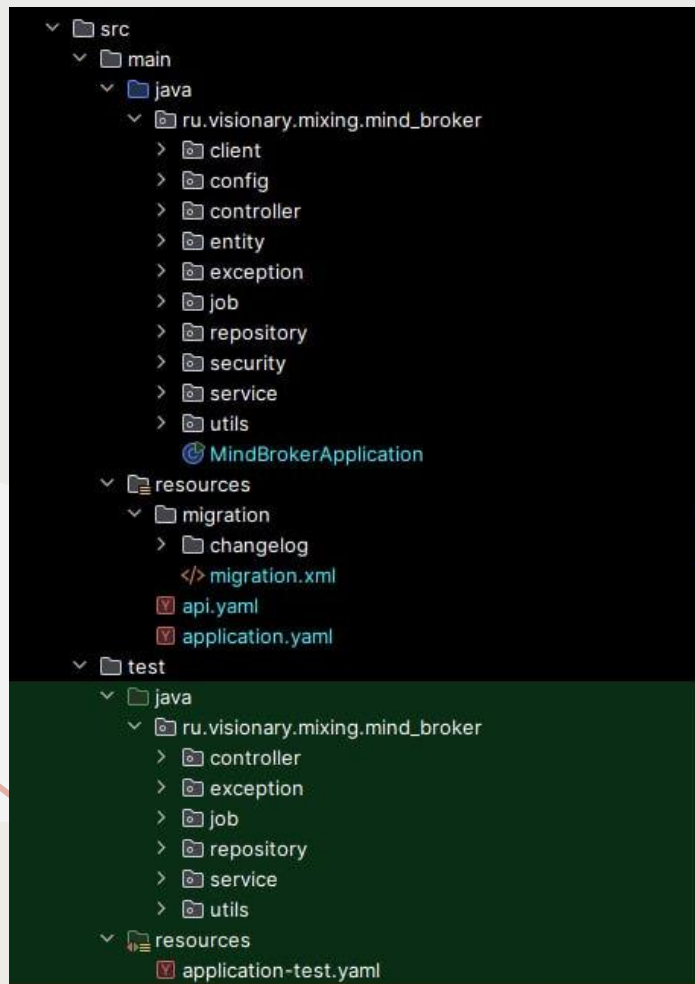


**POST** `/api/v1/processing` Обработка изображения нейросетью



**GET** `/api/v1/styles` Получить все существующие стили

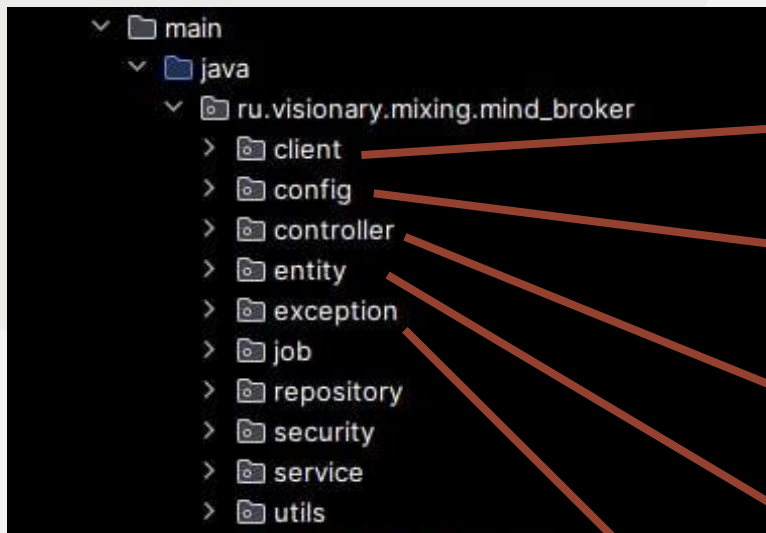




# Структура кода



# Структура кода



CLIENT: http-запросы в другие сервисы

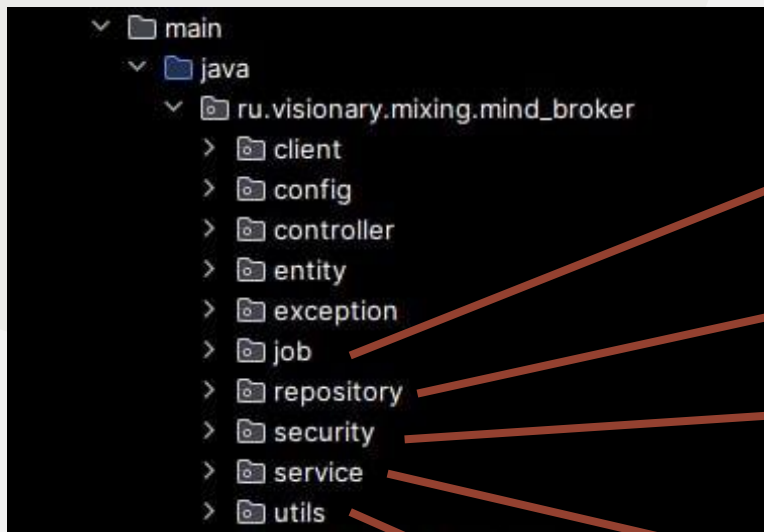
CONFIG: конфигурация отдельных элементов сервиса

CONTROLLER: контроллеры, принимающие входящие http-запросы

ENTITY: сущности, отражающие хранящиеся в БД данные

EXCEPTION: логика обработки ошибок

# Структура кода



JOB: логика, которая выполняется раз в какое то время

REPOSITORY: работа с базой данных

SECURITY: логика, связанная с защитой, авторизацией, аутентификацией

SERVICE: слой обработки запросов, поступающих на контроллеры

UTILS: вспомогательная логика



# Тестирование

Coverage Coverage x				
⏏ ⬆ ⬇ ⬅ ⬇ ⬇				
Element ^	Class, %	Method, %	Line, %	Branch, %
✓ [icon] ru.visionary.mixing	74% (56/75)	53% (174/323)	46% (564/12...	25% (91/360)
> [icon] generated	100% (15/...	43% (65/148)	23% (91/388)	0% (0/146)
✓ [icon] mind_broker	68% (41/60)	62% (109/175)	57% (473/817)	42% (91/214)
> [icon] client	50% (2/4)	40% (2/5)	18% (2/11)	0% (0/4)
> [icon] config	11% (1/9)	3% (1/26)	4% (2/46)	100% (0/0)
> [icon] controller	100% (4/4)	93% (14/15)	90% (20/22)	100% (0/0)
> [icon] entity	100% (9/9)	72% (18/25)	76% (23/30)	0% (0/2)
> [icon] exception	100% (3/3)	83% (10/12)	94% (36/38)	100% (0/0)
> [icon] job	100% (1/1)	100% (3/3)	85% (18/21)	42% (6/14)
> [icon] repository	100% (10/...	96% (30/31)	94% (143/151)	100% (0/0)
> [icon] security	0% (0/2)	0% (0/9)	0% (0/37)	0% (0/12)
> [icon] service	60% (9/15)	62% (28/45)	49% (216/440)	46% (76/164)
> [icon] utils	100% (2/2)	100% (3/3)	65% (13/20)	50% (9/18)
[icon] MindBrokerApplication	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)

# Текущий статус проекта

База данных,

Docker, PostgreSQL, MinIO

RabbitMQ, CI/CD

Авторизацию и регистрацию с JWT

Ключевые экраны

CRUD по основным сущностям

Автодеплой

Подключение системы сбора  
метрик

Связать остаток  
функционала

Завершить верстку экранов

Завершить настройку  
пайплайна нейронной сети

# Visionary

## Minimum Viable Product (MVP)

Приложение для перерисовки фотографий под одну из заранее определенных стилистик



Github



Reports