Отчёт о тестировании проекта Visionary

SCRUM-113 «Получение, обновление, удаление изображений»

Цель фичи:  
 Реализовать CRUD-эндпоинты для изображений с поддержкой листинга, пагинации и фильтрации по protection.

1. GET single image by UUID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Public-image | GET /api/v1/image/{publicUuid} | 200 + ImageResponse | 200 | ✅ |
| Private-image владельцем | GET /api/v1/image/{privateUuid} + JWT (owner) | 200 + ImageResponse | 200 | ✅ |
| Private-image чужим | GET /api/v1/image/{privateUuid} + JWT (other) | 403 ACCESS\_FORBIDDEN | 403 | ✅ |
| Несуществующий UUID | GET /api/v1/image/{randomUuid} | 404 IMAGE\_NOT\_FOUND | 404 | ✅ |
| Без токена (private) | GET /api/v1/image/{privateUuid} без JWT | 401 Unauthorized | 401 | ✅ |

2. PUT update protection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Владелец меняет protection | PUT /api/v1/image/{uuid} + JWT(owner) | 200 + обновлённый ImageResponse | 200 | ✅ |
| Чужой пытается | PUT /api/v1/image/{uuid} + JWT(other) | 403 ACCESS\_FORBIDDEN | 403 | ✅ |
| Без токена | PUT /api/v1/image/{uuid} без JWT | 401 Unauthorized | 401 | ✅ |

3. DELETE image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Владелец удаляет | DELETE /api/v1/image/{uuid} + JWT(owner) | 200; затем GET → 404 IMAGE\_NOT\_FOUND | 200→404 | ✅ |
| Админ удаляет чужое | DELETE /api/v1/image/{uuid} + JWT(admin) | 200 | 200 | ✅ |
| Чужой без прав | DELETE /api/v1/image/{uuid} + JWT(other) | 403 ACCESS\_FORBIDDEN | 403 | ✅ |
| Без токена | DELETE /api/v1/image/{uuid} без JWT | 401 Unauthorized | 401 | ✅ |

4. GET image list (current user)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Валидный: size/page + protection + JWT | GET /api/v1/user/images?size=...&protection=... + JWT | 200 + массив изображений | 200 | ✅ |
| invalid protection | ...?protection=invalid + JWT | 400 INVALID\_REQUEST | 400 | ✅ |
| Пользователь удалён | для удалённого ID + JWT | 410 USER\_DELETED | 410 | ✅ |
| Без токена | GET ... без JWT | 401 Unauthorized | 401 | ✅ |

5. GET image list (other user)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Активный чужой + JWT | GET /api/v1/user/{otherId}/images?… + JWT | 200 + только public | 200 | ✅ |
| Неактивный чужой | для удалённого ID + JWT | 410 USER\_DELETED | 410 | ✅ |
| Без токена | без JWT | 401 Unauthorized | 401 | ✅ |

SCRUM-98 «Регистрация и авторизация»

Цель фичи:  
 Полный end-to-end поток регистрации, логина и ротации JWT (access + refresh).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Регистрация нового пользователя | POST /api/v1/auth/register + валидный | 200 + AuthResponse (tokens) | 200 | ✅ |
| Регистрация: дубликат email | POST register + существующий email | 409 EMAIL\_ALREADY\_IN\_USE | 409 | ✅ |
| Регистрация: дубликат nickname | POST register + существующий nickname | 409 NICKNAME\_ALREADY\_IN\_USE | 409 | ✅ |
| Логин корректный | POST /api/v1/auth/login + валидные creds | 200 + AuthResponse | 200 | ✅ |
| Логин неверный пароль | POST login + wrong password | 401 INVALID\_CREDENTIALS | 401 | ✅ |
| Refresh валидного токена | POST /api/v1/auth/refresh + valid token | 200 + новые токены | 200 | ✅ |
| Refresh просроченного токена | POST refresh + expired token | 401 EXPIRED\_REFRESH\_TOKEN | 401 | ✅ |
| Refresh несуществующего токена | POST refresh + random token | 401 INVALID\_REFRESH\_TOKEN | 401 | ✅ |

SCRUM-112 «Сохранение изображений на сервере»

Цель фичи:  
 POST /api/v1/image/save — загрузка multipart, сохранение в MinIO и БД.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Валидный JPEG | POST save + JPEG < maxSize + JWT | 200 + SaveImageResponse | 200 | ✅ |
| Пустой файл | POST save + файл size=0 | 400 EMPTY\_FILE | 400 | ✅ |
| Неподдерж. формат | POST save + .png или .txt | 415 FILE\_FORMAT\_NOT\_SUPPORTED | 415 | ✅ |
| Файл > maxSize | POST save + >100 MB | 413 FILE\_TOO\_LARGE | 413 | ✅ |
| Без токена | POST save без JWT | 401 Unauthorized | 401 | ✅ |
| Без file part | POST save без части file | 400 INVALID\_REQUEST | 400 | ✅ |

SCRUM-111 «Обновление / Удаление профиля пользователя»

Цель фичи:  
 PUT /api/v1/user/{id}, DELETE /api/v1/user/{id}, DELETE /api/v1/user/{id}/avatar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Update own profile | PUT /api/v1/user/{id} + валидные данные + JWT | 200 OK | 200 | ✅ |
| Update чужого профиля | PUT /api/v1/user/{otherId} + JWT | 403 Forbidden | 403 | ✅ |
| Update неактивного пользователя | PUT /api/v1/user/{deletedId} + JWT | 401 USER\_DELETED | 401 | ✅ |
| Delete own profile | DELETE /api/v1/user/{id} + JWT | 200 OK; при повторном GET → 401 | 200→401 | ✅ |
| Delete чужого профиля | DELETE /api/v1/user/{otherId} + JWT | 403 Forbidden | 403 | ✅ |
| Delete own avatar | DELETE /api/v1/user/{id}/avatar + JWT | 200 OK; avatarUuid → null | 200 | ✅ |
| Delete чужого avatar | DELETE /api/v1/user/{otherId}/avatar + JWT | 403 Forbidden | 403 | ✅ |

SCRUM-105 «Получение информации о пользователе»

Цель фичи:  
 GET /api/v1/user/{userId}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| Успешное получение | GET /api/v1/user/1 + JWT | 200 OK + UserResponse | 200 | ✅ |
| Пользователь не найден | GET /api/v1/user/999 + JWT | 404 USER\_NOT\_FOUND | 404 | ✅ |
| Пользователь удалён | GET /api/v1/user/{deletedId} + JWT | 410 USER\_DELETED | 410 | ✅ |
| Без токена | GET /api/v1/user/1 без JWT | 401 Unauthorized | 401 | ✅ |

SCRUM-103 «Получение изображений пользователя»

Цель фичи:  
 Эндпоинты для просмотра own/other user images с фильтром по protection и пагинацией.

Тесты дублируют пункты из SCRUM-113 (листинг own и чужих изображений) и GET single image.

SCRUM-102 «Обработка изображений через нейросеть»

Цель фичи:

Предоставить список стилей.

Асинхронная обработка задания через RabbitMQ и Megamind.

Надёжное сохранение задач и результатов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Запрос | Ожидание | Факт | Статус |
| GET /styles + JWT | GET /api/v1/styles + JWT | 200 + StylesResponse | 200 | ✅ |
| GET /styles без JWT | GET /api/v1/styles без JWT | 401 UNAUTHORIZED | 401 | ✅ |
| POST /process успешный | POST /api/v1/process?style=...&strength=... + JPG + JWT | 200 + ProcessingImageResponse { uuid } | 200 | ✅ |
| POST /process без JWT | без Authorization | 401 UNAUTHORIZED | 401 | ✅ |
| POST /process unknown style | style=foo + JPG + JWT | 404 STYLE\_NOT\_FOUND | 404 | ✅ |
| POST /process пустой файл | MultipartFile size=0 + JWT | 400 EMPTY\_FILE | 400 | ✅ |
| POST /process неподдерж. формат | .pdf или .txt + JWT | 415 FILE\_FORMAT\_NOT\_SUPPORTED | 415 | ✅ |
| POST /process strength вне [0,1] | strength=-0.1 или 1.1 + JPG + JWT | 400 INVALID\_REQUEST | 400 | ✅ |
| Megamind → 500 | MegamindClient throws HTTP 500 | 502 MEGAMIND\_ERROR | 502 | ✅ |
| cron-job updateStyles | ручной вызов updateStyles() | Репозиторий обновлён | — | ✅ |
| Listener: timeout | message старее threshold | Статус → CANCELED | — | ✅ |
| Listener: normal processing | Megamind OK | PROCESSING → COMPLETED | — | ✅ |
| Listener: Megamind error | MegamindClient throws ServiceException | Статус → FAILED | — | ✅ |

SCRUM-121 «Splash-screen»

Цель фичи:  
 При старте показывать Splash-screen с логотипом, затем авто-переход на LoginScreen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Действие | Ожидание | Факт | Статус |
| Появление Splash-screen | Холодный старт приложения | Логотип на белом фоне | ✅ | ✅ |
| Длительность ≈1,25 с | Измерить время | ≈1,25 сек | 1,27 сек | ✅ |
| Авто-переход на LoginScreen | По таймеру | Переход на экран входа | ✅ | ✅ |
| Back-кнопка на Splash | Нажать «Back» | Закрытие/сворачивание приложения | ✅ | ✅ |
| Поворот экрана во время Splash | Повернуть устройство | Логотип остаётся центрированным, таймер идёт | ✅ | ✅ |

SCRUM-86 «Экран настроек»

Цель фичи:  
 Экран «Настройки» с набором пунктов (Subscription, Pro/Premium, Notify, Account, Change Password, Logout, Delete Account, Personalization, Change Theme).

**Компонент:** SettingsItem + mock-диалоги.

|  |  |  |  |
| --- | --- | --- | --- |
| Пункт меню | Ожидаемое поведение | Факт | Статус |
| Подписка | Открытие proDialog | ✅ | ✅ |
| Premium-версия | Открытие premiumDialog | ✅ | ✅ |
| Уведомления | Переключение тумблера | ✅ | ✅ |
| Аккаунт | Без действий (mock) | ✅ | ✅ |
| Сменить пароль | Появление поля ввода и скрытие по check | ✅ | ✅ |
| Выйти из аккаунта | Mock-диалог подтверждения | ✅ | ✅ |
| Удалить аккаунт | Mock-диалог удаления | ✅ | ✅ |
| Персонализация | Открытие mock-диалога | ✅ | ✅ |
| Сменить тему | Переключение UI-цвета (mock) | ✅ | ✅ |
| Скролл списка | Корректная прокрутка | ✅ | ✅ |
| Ландшафтная ориентация | Адаптивность, кликабельность без обрезки текста | ✅ | ✅ |

SCRUM-82 «Экраны профиля»

Фича:

MyProfileScreen и OtherUserProfileScreen (заглушки).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элемент | Действие | Ожидание | Факт | Статус |
| Аватар (свой/чужой) | Тап | Без падений | ✅ | ✅ |
| Поле описания | Тап | Фокус, ввод запрещён | ✅ | ✅ |
| Иконки статистики | Тап | Без ошибок | ✅ | ✅ |
| Кнопки «Публикации/Личное» | Тап | Переключение табов | ✅ | ✅ |
| Элемент грид-изображения | Тап | Без падений | ✅ | ✅ |

SCRUM-123 «Камера и галерея»

Фича:

Подключить выбор/съёмку изображения (CreatePictureScreen).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элемент | Действие | Ожидание | Факт | Статус |
| «Сделать фото» | Тап | Запрос разрешений → открытие камеры | ✅ | ✅ |
| «Выбрать из галереи» | Тап | Запрос разрешений → открытие галереи | ✅ | ✅ |
| Perms CAMERA / STORAGE | Согласие/отказ | Корректная обработка разрешений | ✅ | ✅ |
| Полученный URI | После операции | URI сохраняется в состоянии компонента | ✅ | ✅ |
| Отображение фото | На скрине | Показ выбранного изображения | ✅ | ✅ |

SCRUM-117 «Настроить метрики»

Фича:

Интеграция Spring Actuator + Prometheus.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoint | Метод | Ожидание | Факт | Статус |
| /actuator/prometheus | GET | 200 OK + Prometheus формат | ✅ | ✅ |
| /actuator/health | GET | 200 OK + статус UP | ✅ | ✅ |
| /actuator/metrics/http.server.requests | GET | 200 OK + HTTP метрики | ✅ | ✅ |
| /v3/api-docs | GET | 200 OK + JSON OpenAPI | ✅ | ✅ |