МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет компьютерных наук

Кафедра технологий обработки и защиты информации

*Техническое задание*

*на разработку мобильного приложения­*

*«Оценка качества продукции “QWality”»*

Исполнители

Р.Ю. Перцев

Д.С. Сушкова

В.А. Баранов

В.В. Лихачев

Д.В. Фролов

М.В. Бен Амор

Заказчик

В.С. Тарасов

Воронеж 2025

Содержание

[Определения, обозначения, сокращения 4](#_Toc194241434)

[1. Введение 5](#_Toc194241435)

[2. Общие положения 5](#_Toc194241436)

[2.1. Название приложения 5](#_Toc194241437)

[2.2. Разработчики и заказчик 5](#_Toc194241438)

[2.3. Перечень документов, на основании которых создается приложение 6](#_Toc194241439)

[2.4. План реализации проекта по детектированию дефектов солнечных панелей на производстве 6](#_Toc194241440)

[3. Цели и задачи. 8](#_Toc194241441)

[3.1. Цели 8](#_Toc194241442)

[3.2. Задачи 8](#_Toc194241443)

[3.3. Критерии успешности проекта 8](#_Toc194241444)

[4. Функциональные требования: 9](#_Toc194241445)

[5. Нефункциональные требования: 9](#_Toc194241446)

[6. Стек 10](#_Toc194241447)

[7. Оформление и вёрстка страниц 10](#_Toc194241448)

[8. Дизайн приложения 10](#_Toc194241449)

[9. User stories 26](#_Toc194241450)

[10. Use Case 27](#_Toc194241451)

[10.1. Сценарий: Администратор изменяет параметры ИИ 27](#_Toc194241452)

[10.2. Сценарий: Попытка изменения параметров ИИ пользователем без прав администратора 27](#_Toc194241453)

[10.3. Сценарий: Проверка качества товара 28](#_Toc194241454)

[10.3.1. Успешная проверка: 28](#_Toc194241455)

[10.3.2. Неуспешная проверка: 28](#_Toc194241456)

[10.4. Сценарий: Просмотр статистики 28](#_Toc194241457)

[10.5. Сценарий: Регистрация нового пользователя в системе 29](#_Toc194241458)

[10.6. Сценарий: Пользователь переходит с экрана регистрации на экран входа 30](#_Toc194241459)

[10.7. Сценарий: Пользователь восстанавливает пароль 30](#_Toc194241460)

[10.8. Сценарий: Пользователь выбирает уровень подписки 30](#_Toc194241461)

[10.9. Сценарий: Пользователь настраивает параметры приложения 31](#_Toc194241462)

[10.10. Сценарий: Пользователь использует экран FAQ 31](#_Toc194241463)

[10.11. Сценарий: Пользователь управляет профилем 32](#_Toc194241464)

[10.12. Сценарий: Пользователь добавляет метод оплаты 33](#_Toc194241465)

[10.13. Сценарий: Роли и права пользователей с системой уведомлений и логированием 33](#_Toc194241466)

[10.14. Сценарий: Управление камерами модератором 34](#_Toc194241467)

[10.15. Сценарий: Управление админ панелью администратором 34](#_Toc194241468)

[10.16. Сценарий: Управление админ панелью владельцем 35](#_Toc194241469)

[10.17. Сценарий: Пользователь добавляет камеру 35](#_Toc194241470)

[10.18. Сценарий: Изменение статуса камеры 35](#_Toc194241471)

[10.19. Сценарий: Удаление камеры 36](#_Toc194241472)

[10.20. Сценарий: Автоматическая чистка корзины 36](#_Toc194241473)

[10.21. Сценарий: Ручная чистка корзины 36](#_Toc194241474)

[11. Перечень основных функциональных блоков системы 38](#_Toc194241475)

[12. Ограничения проекта 38](#_Toc194241476)

[12.1. Поддерживаемые типы солнечных панелей 38](#_Toc194241477)

[12.2. Качество и формат данных 38](#_Toc194241478)

[12.3. Ограничения модели машинного обучения 39](#_Toc194241479)

[12.4. Требования к оборудованию 39](#_Toc194241480)

[12.5. Скорость обработки 40](#_Toc194241481)

[ПРИЛОЖЕНИЕ 41](#_Toc194241482)

Определения, обозначения, сокращения

В настоящем техническом задании применяют следующие термины с соответствующими определениями:

Электролюминесцентный спектр – метод визуализации, применяемый для выявления дефектов в структуре солнечных панелей.

Журнал логов (логирование) – автоматическая запись действий пользователей и работы системы для аудита и отладки.

CI/CD (Continuous Integration/Continuous Deployment) – методология автоматизированной сборки, тестирования и развертывания кода (реализована через GitHub Actions и Docker).

API (Application Programming Interface) – интерфейс программирования приложений, позволяющий взаимодействовать между клиентской и серверной частями системы.

UI Kit – набор дизайн-компонентов интерфейса в Figma.

GPU (Graphics Processing Unit) – графический процессор, используемый для ускорения обработки изображений.

CUDA – технология NVIDIA для параллельных вычислений на GPU.

RAM (Random Access Memory) – оперативная память сервера.

VRAM (Video RAM) – видеопамять графической карты.

FAQ (Frequently Asked Questions) – раздел с ответами на часто задаваемые вопросы.

1. Введение

Солнечная энергетика играет огромную роль в развитии возобновляемых источников энергии. Однако со временем солнечные панели неизбежно изнашиваются: на них появляются микротрещины, участки перегрева, загрязнения и другие повреждения. Эти дефекты снижают эффективность работы панелей и в конечном итоге могут вывести их из строя. Поэтому особенно важно создать умную, автоматизированную систему, которая сможет вовремя выявлять такие проблемы с помощью технологий искусственного интеллекта.

1. Общие положения
   * 1. Название приложения

Полное наименование продукта: «приложение для обнаружения дефектов на солнечных панелях QWality»

Условное обозначение продукта: «QWality»

* + 1. Разработчики и заказчик

Заказчик: старший преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчик: 2 команда группы 4

Состав команды разработчика:

* Перцев Роман Юрьевич
* Лихачев Валерий Валерьевич
* Сушкова Дарья Сергеевна
* Бен Амор Мохамед Вассим
* Фролов Данила Валерьевич
* Баранов Виталий Алексеевич
  + 1. Перечень документов, на основании которых создается приложение

Данное ПО будет создаваться на основании следующих документов:

* **Техническое задание** – определяет требования к функционалу и архитектуре приложения;
* федерального закона "О персональных данных" от 27.07.2006 N 152-ФЗ;
* федерального закона «Об информации, информационных технологиях и о защите информации» от 27.07.2006 г. № 149-ФЗ.
  + 1. План реализации проекта по детектированию дефектов солнечных панелей на производстве

Сбор необходимой информации, постановка целей и задач системы (25.02.25 – 04.03.25)

* Сделан UI Kit;
* Написана первая версия ТЗ (планируется доработка);
* Описана ER-диаграмма;
* Определен стек технологий;

Анализ предметной области, анализ конкурентов и построение структуры требований (04.03.25 – 14.03.25)

* Закончен ЧБ дизайн экранов в Figma;
* Закончена схема API;
* Закончен брендбук;
* Описаны 80% диаграмм для Mobile-версии;

Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД (14.03.25 – 21.03.25)

* Полностью завершено написание ТЗ;
* Созданы диаграммы State Chart по экранам;
* Сделано цветное исполнение экранов в Figma;
* Сделаны экраны Web-версии;
* Записана презентация 1-го этапа;

Разработка рабочего проекта (написание программного кода, отладка и корректировка) (21.03.25 – 01.05.25)

* Разработка бэкенда для обработки изображений и работы с нейросетевой моделью;
* Разработка фронтенда;
* Интеграция с базой данных и реализация API;
* Реализация функционала логирования, отчетности и ролевого доступа;
* Первичная отладка, исправление багов;

Проведение тестирования программного обеспечения (01.05.25 – 01.06.25)

* Тестирование работы модели на реальных изображениях солнечных панелей;
* Проверка точности детекции дефектов;
* Тестирование API и пользовательского интерфейса на корректность работы;
* Нагрузочное тестирование для оценки производительности системы;
* Финальные исправления по результатам тестирования;
* Сдача итогового проекта (начало июня);
* Подготовка финальной версии системы;
* Составление документации;
* Демонстрация работы системы, проверка соответствия требованиям;

1. Цели и задачи.
   * 1. Цели

* **Автоматизация контроля качества** – разработка приложения для мониторинга и анализа качества продукции.
* **Повышение прозрачности** – хранение и анализ данных о проверках качества.
* Полный доступ к информации
  + 1. Задачи
* Разработка мобильного приложения, которое будет предоставлять доступ к информации о браке (процентное соотношение брака к нормальному товару, отображение участков производства с наивысшим количеством брака, доступ к фотографиям с бракованным товаром).
* Создание системы отчетности для мониторинга качества продукции.
* Обеспечение ролевого доступа для разных категорий пользователей.
* Разработка искусственного интеллекта, который будет проводить оценку качества товара.
* Разработка веб приложения.
  + 1. Критерии успешности проекта
* Функциональность – все заявленные функции работают стабильно и корректно.
* Скорость работы – задержка, не превышающая заявленной, при вводе и обработке информации.
* Кроссплатформенность – поддержка разных типов устройств
* Безопасность данных – защита информации и разграничение доступа.

1. Функциональные требования:

* Распознавание дефектов по классам.
* Генерация отчетов по запросу.
* Доступ к разным функциям в зависимости от роли.
* Интеграция с базами данных для хранения результатов анализа.
* Экспорт отчетов в форматы PDF, CSV.
* Поддержка форматов изображений JPEG, PNG
* Удаление истории логов.
* Логирование всех решений ИИ.

1. Нефункциональные требования:

* **Производительность – высокая скорость обработки изображений за счёт эффективного алгоритма.**
* **Доступность системы - 99,5%** это означает, что система не может простаивать более **3,65 часов** в месяц).
* **Работа в условиях производственного потока:** не менее 2 панелей в минуту при условии, что панели состоят из 60 ячеек.
* **Поддержка пиковых нагрузок: до 5 панелей в минуту**, масштабируемая через балансировку нагрузки, при условии, что панели состоят из 60 ячеек.
* **Совместимость -** поддерживает работу в популярных браузерах: Google Chrome, Mozilla Firefox, Yandex.
* Энергоэффективность - обработка изображений выполняется **на сервере**, а не на мобильном устройстве.
* Поддержка **развертывания в облаке и на локальных серверах**.
* Обеспечение безопасности данных и контроль доступа.
* Дизайн, соответствующий заявленному набору для пользовательского интерфейса.

1. Стек

* Бекенд - python, Flask
* ИИ - python, PyTorch
* Android - React Native
* CI/CD - github actions
* Контейнеризация – Docker
* Объектное хранилище - MinIO
* СУБД - postgreSQL

1. ****Оформление и вёрстка страниц****

* Приложение должно быть выдержано в едином стиле.
* Должно быть разработано название, которое используется в оформлении страниц.
* В дизайне применяется единая цветовая палитра и ограниченный набор шрифтов.
* Цвета должны быть контрастными, чтобы интерфейс был удобным и читаемым.
* Все страницы должны корректно отображаться на экранах разных размеров.

1. Дизайн приложения

При первом запуске приложения появляется экран с информацией про приложение и компанию. На экране есть 2 функциональные кнопки «Зарегистрироваться» и «Уже есть аккаунт? Войти» рисунок 1.

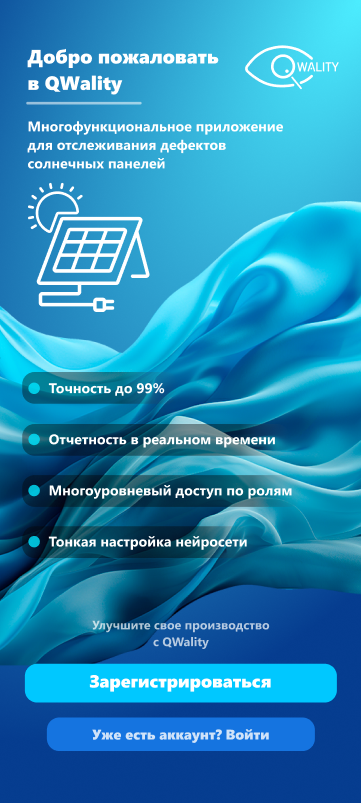


Рисунок 1 – приветственный экран

При нажатии на кнопку регистрации появляется экран регистрации рисунок 2.

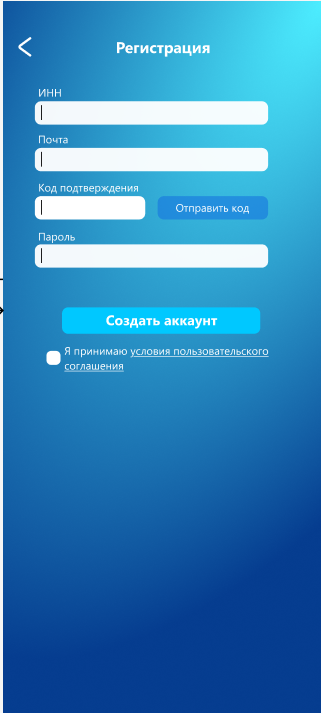


Рисунок 2 - экран регистрации

При нажатии на кнопку входа появляется экран входа рисунок 3.

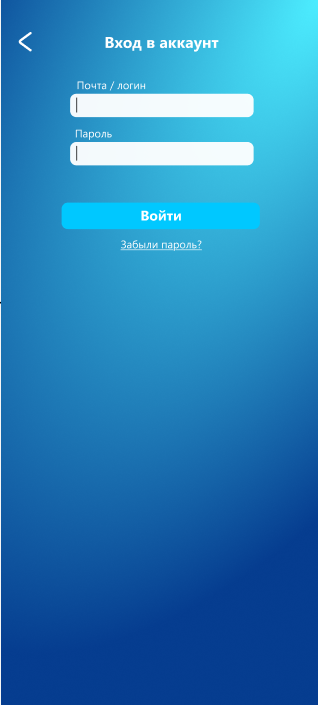


Рисунок 3 - экран входа

На экране входа есть функциональная кнопка «Восстановить пароль» при нажатии на неё открывается экран восстановления пароля рисунок 4.

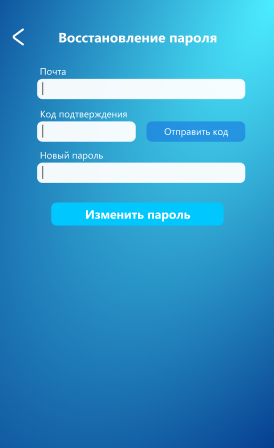


Рисунок 4 - экран восстановления пароля

После регистрации, пользователь может выбрать демо режим, либо сразу купить подписку рисунок 5.

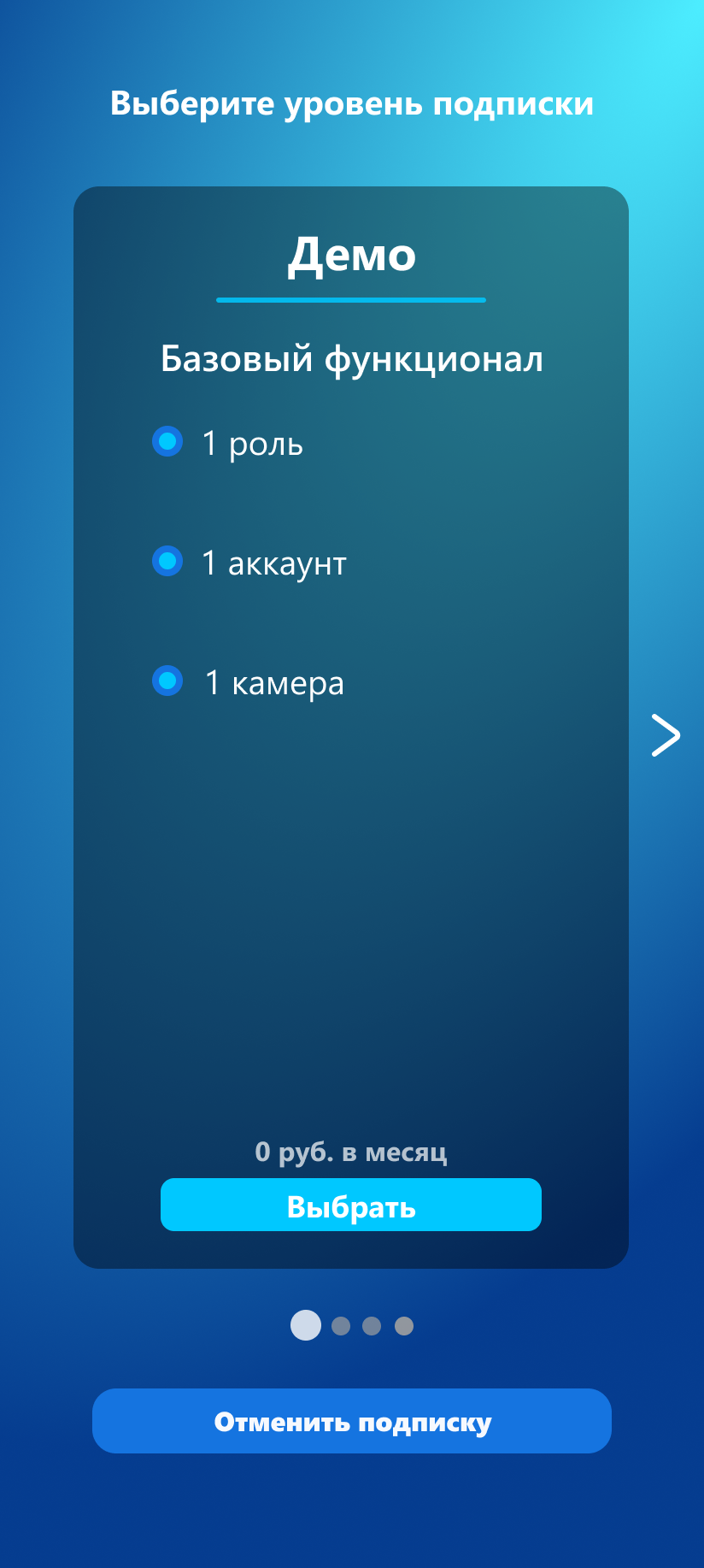


Рисунок 5 - экран выбора демо режима или уровня подписки

После заполнения реквизитов и оплаты подписки или выбора демо режима, создается аккаунт, которым можно управлять на экране профиля рисунок 6.

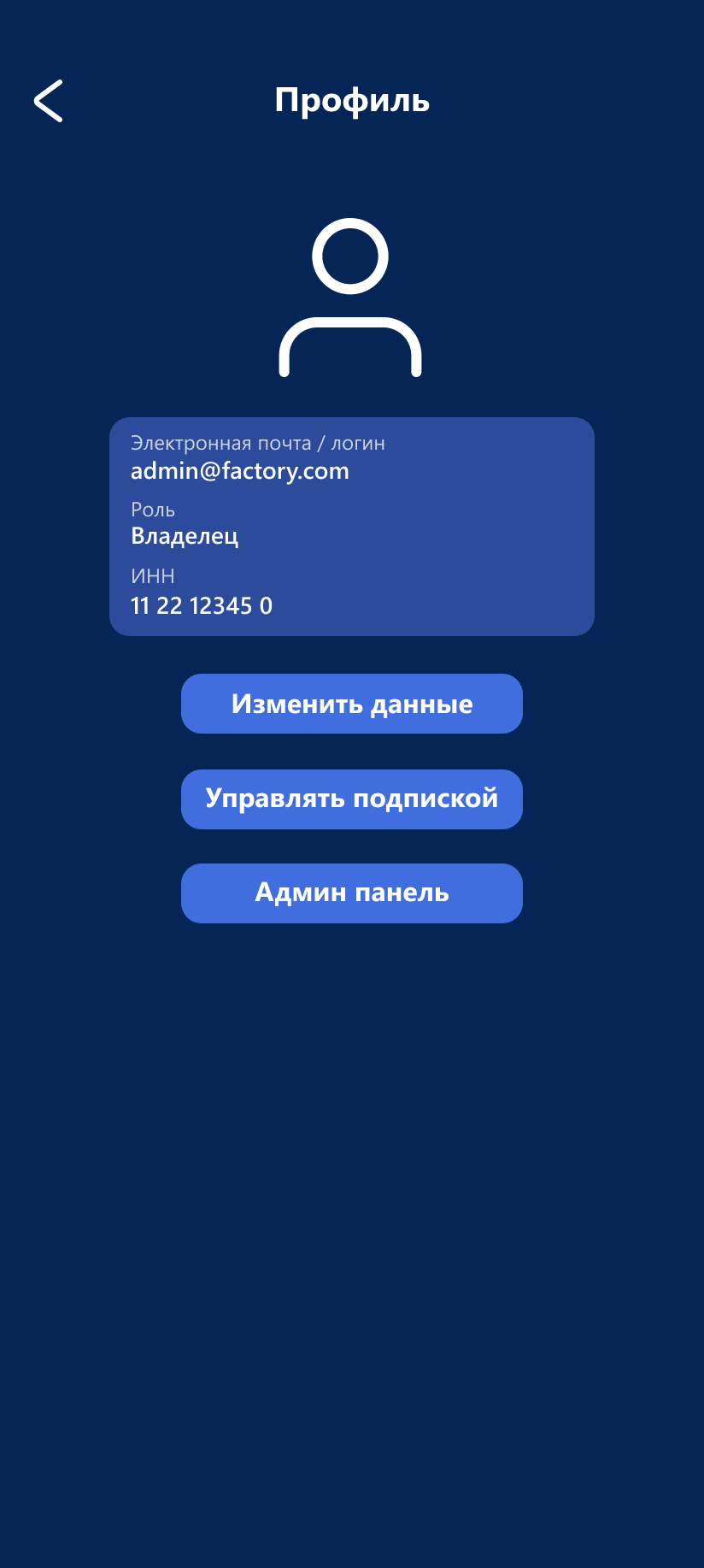


Рисунок 6 - экран профиля

На экране профиля есть 2 функциональные кнопки «Изменить данные» и «Админ панель» при нажатии на кнопку изменить данные осуществляется переход на экран редактирования данных пользователя рисунок 7.

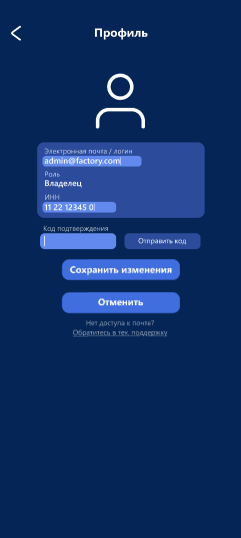


Рисунок 7 - экран редактирования данных пользователя

При нажатии на кнопку «Админ панель» осуществляется переход на экран админ панели рисунок 8.



Рисунок 8 - экран админ панели

На админ панели есть функциональная кнопка «Управлять аккаунтами», при нажатии на неё осуществляется переход на экран управления аккаунтами рисунок 9.

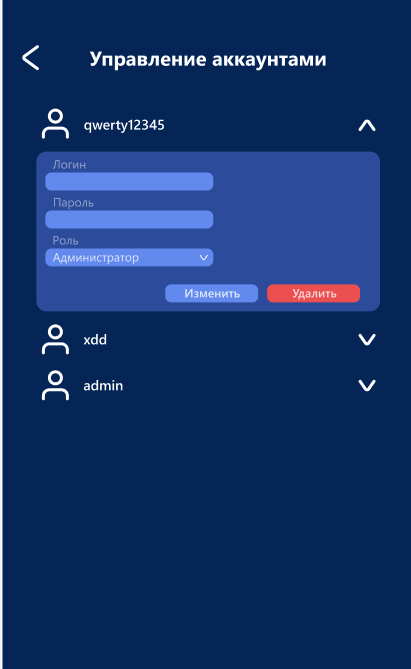


Рисунок 9 - экран управления аккаунтами

Из профиля пользователь может перейти на главный экран рисунок 10.

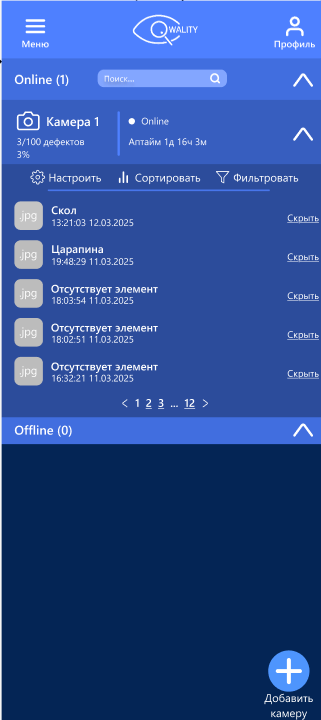


Рисунок 10 - главный экран

С главного экрана пользователь может перейти на экран добавления камеры рисунок 11.

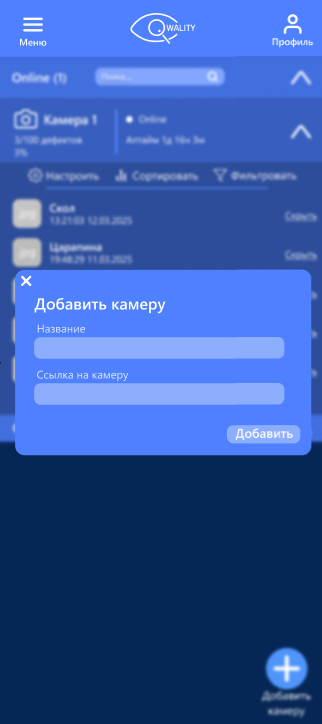


Рисунок 11 - экран добавления камеры

Также с главного экрана пользователь может через меню перейти на экран корзины рисунок 12.



Рисунок 12 - экран корзины

С экрана корзины пользователь может перейти на экран очистки рисунок 13.

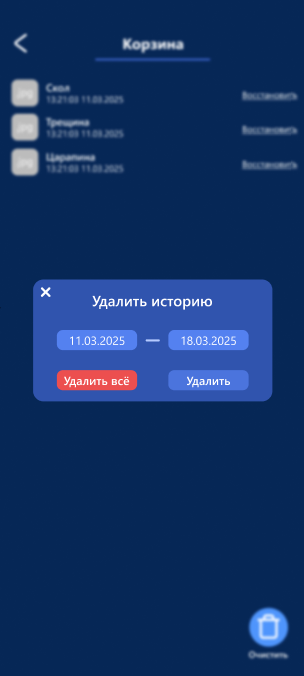


Рисунок 13 - экран очистки

Пользователь с главного экрана может перейти на экран настроек рисунок 14.

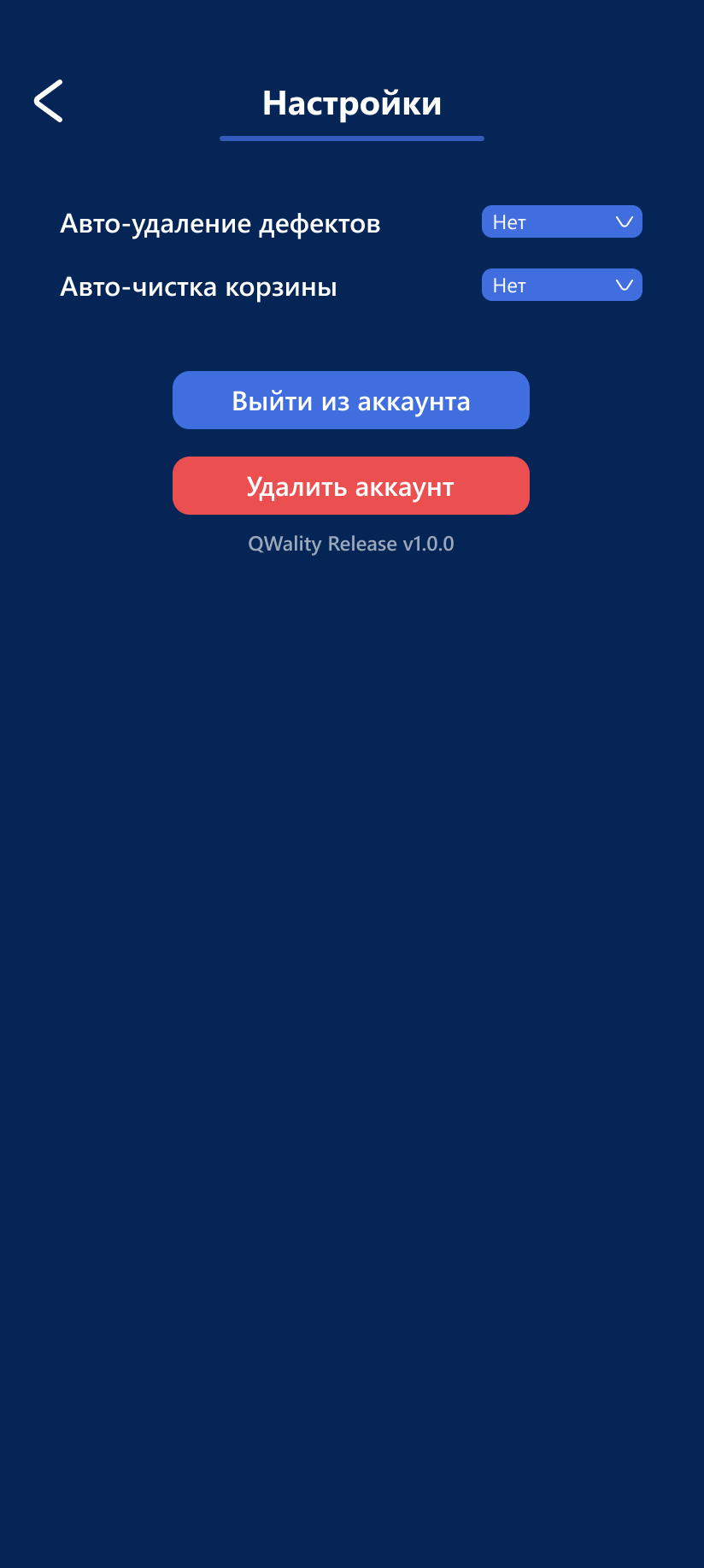


Рисунок 14 - экран настроек

На экране настроек есть одна функциональная кнопка для выхода из аккаунта, при нажатии на которую осуществляется переход на экран выхода из аккаунта рисунок 15.

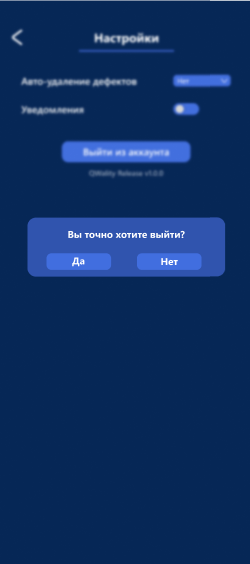


Рисунок 15 - экран выхода из аккаунта

1. User stories

* Как новый пользователь, я хочу увидеть экран с информацией о приложении и компании при первом запуске, чтобы ознакомиться с его возможностями перед регистрацией.
* Как новый пользователь, я хочу зарегистрироваться в системе, чтобы получить доступ к возможностям приложения.
* Как зарегистрированный пользователь, я хочу иметь возможность войти в систему, чтобы использовать свой аккаунт.
* Как пользователь, забывший пароль, я хочу иметь возможность восстановить доступ к своему аккаунту, чтобы продолжить использовать приложение.
* Как новый пользователь, я хочу выбрать между демо-режимом или подпиской, чтобы решить, хочу ли я протестировать приложение перед покупкой.
* Как пользователь, выбравший подписку, я хочу перейти к оплате и заполнению реквизитов, чтобы активировать подписку и получить полный доступ.
* Как новый пользователь, я хочу попасть в свой профиль после регистрации и выбора подписки или демо-режима, чтобы начать использовать приложение.
* Как пользователь, я хочу иметь возможность редактировать свои личные данные, чтобы обновлять информацию в своем профиле.
* Как администратор, я хочу перейти в админ-панель, чтобы управлять пользователями и камерами.
* Как администратор, я хочу иметь возможность управлять учетными записями пользователей, чтобы назначать роли и удалять аккаунты.
* Как пользователь, я хочу иметь возможность вернуться на главный экран из профиля, чтобы переключиться на другие функции приложения.
* Как пользователь, я хочу иметь возможность добавить новую камеру, чтобы использовать её для мониторинга и инспекций.
* Как пользователь, я хочу иметь доступ к корзине удаленных камер, чтобы просмотреть и восстановить их в течение 30 дней.
* Как пользователь, я хочу очистить корзину вручную, чтобы освободить место и удалить камеры окончательно.
* Как пользователь, я хочу иметь доступ к разделу FAQ, чтобы быстро найти ответы на свои вопросы о функционале приложения.
* Как пользователь, я хочу изменять настройки приложения, чтобы настроить язык, тему и размер шрифта под себя.
* Как пользователь, я хочу выйти из аккаунта через настройки, чтобы завершить работу в системе.

1. Use Case
   * 1. Сценарий: Администратор изменяет параметры ИИ

Дано: администратор вошел в систему

И: у него есть права редактирования

Когда: администратор изменяет чувствительность ИИ к определенным дефектам

Тогда: система применяет новые параметры

И: изменения фиксируются в журнале логов

* + 1. Сценарий: Попытка изменения параметров ИИ пользователем без прав администратора

Дано: пользователь вошел в систему

И: у него нет прав редактирования

Когда: пользователь пытается изменить чувствительность ИИ к определенным дефектам

Тогда: система отклоняет попытку изменения

И: отображается сообщение об ошибке «Недостаточно прав для выполнения операции»

И: попытка изменения фиксируется в журнале логов как неудачная попытка доступа

* + 1. Сценарий: Проверка качества товара

Дано: изделие передано на проверку

И: изображение загружено в систему

* + - 1. Успешная проверка:

Когда: система анализирует изображение

Тогда: дефекты не обнаружены

И: запись сохраняется в базе данных

* + - 1. Неуспешная проверка:

Когда: система анализирует изображение

Тогда: обнаружены дефекты

И: система классифицирует дефекты

И: дефект фиксируется в базе данных с изображением и описанием

* + 1. Сценарий: Просмотр статистики

Дано: инженер по качеству вошел в систему

И: у него есть доступ к разделу аналитики

Когда: инженер открывает раздел статистики

Тогда: система загружает сводные данные о проверках

И: отображаются графики и таблицы с динамикой дефектов

И: доступны фильтры для отбора данных (по дате, смене, типу дефекта, партии продукции)

И: инженер может выбрать нужный фильтр и обновить отображаемую статистику

Когда: инженер хочет экспортировать данные

Тогда: он выбирает формат отчета (PDF, Excel)

И: система генерирует файл и предлагает его скачать

* + 1. Сценарий: Регистрация нового пользователя в системе

Дано: новый пользователь хочет получить доступ к системе

Когда: пользователь открывает страницу регистрации

Тогда: система запрашивает ввод данных (ИНН, email, пароль).

И: пользователь заполняет все обязательные поля

Когда: пользователь нажимает кнопку «Зарегистрироваться».

Тогда: система проверяет корректность введенных данных

И: если данные корректны, система отправляет запрос на подтверждение регистрации администратору

И: пользователь видит сообщение «Ваша заявка на регистрацию отправлена»

Когда: администратор одобряет регистрацию

Тогда: система отправляет пользователю письмо с подтверждением

И: пользователь может войти в систему с указанными данными

Когда: данные некорректны

Тогда: система выводит сообщение об ошибке и предлагает исправить данные

* + 1. Сценарий: Пользователь переходит с экрана регистрации на экран входа

Дано: пользователь находится на экране регистрации

И: на экране присутствует кнопка "Уже есть аккаунт?"

Когда: пользователь нажимает на кнопку "Уже есть аккаунт?"

Тогда: система перенаправляет пользователя на экран входа

И: на экране входа отображаются поля для ввода логина и пароля.

* + 1. Сценарий: Пользователь восстанавливает пароль

Дано: пользователь находится на экране входа

И: на экране есть кнопка "Забыли пароль?"

Когда: пользователь нажимает на кнопку "Забыли пароль?"

Тогда: система предлагает пользователю восстановить пароль

И: после завершения восстановления, система перенаправляет пользователя обратно на экран входа.

* + 1. Сценарий: Пользователь выбирает уровень подписки

Дано: пользователь завершил регистрацию и вошел в систему

И: система отображает экран выбора уровня подписки

Когда: пользователь видит доступные уровни подписки и их цены

И: рядом указана информация о наличии демо-версии или бесплатного периода

Тогда: пользователь может выбрать подходящий уровень подписки и продолжить процесс оформления.

* + 1. Сценарий: Пользователь настраивает параметры приложения

Дано: пользователь находится на экране настроек

И: на экране доступны параметры для изменения языка, темы, размера шрифта и выхода из аккаунта

Когда: пользователь изменяет язык интерфейса

Тогда: система применяет выбранный язык и обновляет интерфейс

Когда: пользователь выбирает темную или светлую тему

Тогда: система переключает тему и сохраняет выбор

Когда: пользователь изменяет размер шрифта

Тогда: система применяет новый размер шрифта к интерфейсу

Когда: пользователь нажимает кнопку "Выйти из аккаунта"

Тогда: система выполняет выход, очищает пользовательскую сессию

И: перенаправляет пользователя на экран входа.

* + 1. Сценарий: Пользователь использует экран FAQ

Дано: пользователь находится на экране FAQ

И: на экране отображается список часто задаваемых вопросов и ответов

Когда: пользователь вводит запрос в поле поиска

Тогда: система показывает соответствующие вопросы и ответы

Когда: пользователь выбирает конкретный вопрос из списка

Тогда: система раскрывает ответ на этот вопрос

Когда: пользователь не находит нужного ответа

Тогда: система предлагает контакты службы поддержки

И: пользователь может связаться с поддержкой для получения дополнительной информации.

* + 1. Сценарий: Пользователь управляет профилем

Дано: пользователь находится на экране профиля

И: на экране отображается общая статистика профиля

Когда: пользователь просматривает статистику

Тогда: система показывает ключевые данные (например, активность, количество использованных функций, дата регистрации)

Когда: пользователь выбирает изменение личных данных

Тогда: система предоставляет форму редактирования (например, имя, email, аватар)

И: после сохранения изменений обновляет данные в профиле

Когда: пользователь изменяет уровень подписки

Тогда: система отображает доступные тарифы

И: позволяет пользователю выбрать новый тариф и оформить изменение

Когда: пользователь хочет удалить аккаунт

Тогда: система запрашивает подтверждение удаления

И: предупреждает о необратимости действия

* + 1. Сценарий: Пользователь добавляет метод оплаты

Дано: пользователь находится на экране оплаты

И: у пользователя нет активного метода оплаты

Когда: пользователь выбирает способ оплаты (банковская карта, электронный кошелек и т. д.)

Тогда: система отображает форму ввода платежных данных

Когда: пользователь вводит платежные данные и подтверждает оплату

Тогда: система проверяет корректность данных и привязывает метод оплаты

Когда: оплата успешно обработана

Тогда: система уведомляет пользователя о привязке метода оплаты

И: пользователь может использовать его для оформления подписки

Когда: происходит ошибка оплаты

Тогда: система сообщает пользователю об ошибке

И: предлагает повторить попытку или выбрать другой способ оплаты.

* + 1. Сценарий: Роли и права пользователей с системой уведомлений и логированием

Дано: пользователь имеет роль "Рядовой пользователь"

И: он вошел в систему

Когда: пользователь открывает приложение

Тогда: система предоставляет доступ только к просмотру контента (например, просмотр камер, данных профиля и FAQ)

И: не предоставляет доступ к админ-панели и к управлению камерами

И: фиксирует событие входа в систему в журнале логов.

* + 1. Сценарий: Управление камерами модератором

Дано: пользователь имеет роль "Модератор"

И: он вошел в систему

Когда: пользователь открывает список камер

Тогда: система отображает доступные камеры

Когда: пользователь пытается добавить новую камеру

Тогда: система предоставляет интерфейс для добавления камеры

И: после успешного добавления камера отображается в общем списке

И: создается запись в журнале логов о добавлении камеры модератором

Когда: пользователь пытается редактировать параметры существующей камеры

Тогда: система предоставляет доступ к настройкам камеры

И: сохраняет внесенные изменения

И: записывает событие изменения параметров в журнал логов

* + 1. Сценарий: Управление админ панелью администратором

Дано: пользователь имеет роль "Администратор"

И: он вошел в систему

Когда: пользователь открывает админ-панель

Тогда: система предоставляет доступ к управлению камерами и настройками системы

И: не предоставляет доступ к платежной системе и управлению ролями администраторов

И: фиксирует в журнале логов событие входа администратора в админ-панель.

* + 1. Сценарий: Управление админ панелью владельцем

Дано: пользователь имеет роль " Владелец "

И: он вошел в систему

Когда: пользователь открывает админ-панель

Тогда: система предоставляет полный доступ к управлению системой

И: позволяет изменять подписку, управлять платежами

И: позволяет снимать или назначать администраторов

И: записывает в журнал логов изменения подписки и изменения прав пользователей

* + 1. Сценарий: Пользователь добавляет камеру

Дано: пользователь находится на экране управления устройствами

И: на экране есть кнопка "Добавить камеру"

Когда: пользователь нажимает на кнопку "Добавить камеру"

Тогда: система предоставляет форму для ввода данных камеры (например, название, тип, IP-адрес и т.д.).

И: после ввода данных, пользователь подтверждает добавление камеры

И: система добавляет камеру и отображает ее в списке доступных устройств.

* + 1. Сценарий: Изменение статуса камеры

Дано: в системе есть камеры

Когда: камера становится неактивной

Тогда: она перемещается в список неактивных камер

И: отображается соответствующий статус в интерфейсе.

Когда: камера снова становится активной

Тогда: она возвращается в список активных камер

* + 1. Сценарий: Удаление камеры

Дано: пользователь имеет права на удаление камеры

Когда: пользователь нажимает кнопку "Удалить камеру"

Тогда: камера удаляется из списка камер

И: во всех отчетностях вместо данных о камере указывается "Удалено"

И: система записывает событие удаления в логи.

* + 1. Сценарий: Автоматическая чистка корзины

Дано: включен переключатель “автоматическая чистка корзины”, в корзине есть записи о дефектах, хранящиеся более 30 дней.

Когда: запускается автоматическая очистка корзины.

Тогда: все такие записи удаляются из базы данных

И: создается запись в логах об автоматической очистке корзины.

* + 1. Сценарий: Ручная чистка корзины

Дано: пользователь с ролью модератор или выше находится в корзине.

Когда: такой пользователь нажимает кнопку “очистить” и выбирает интервал для очисти истории.

Тогда: все такие записи удаляются из базы данных

И: создается запись в логах об ручной очистке корзины.

1. Перечень основных функциональных блоков системы

* Аутентификация и управление пользователями
* Загрузка и обработка изображений продукции
* Модуль анализа дефектов на основе ИИ
* Статистика и аналитика
* Настройки системы

1. Ограничения проекта

Проект предназначен для автоматического выявления дефектов солнечных панелей на этапе их производства или обслуживания. Он работает в условиях конвейерной линии, где панели движутся с постоянной скоростью под фиксированным освещением.

* + 1. Поддерживаемые типы солнечных панелей

Система поддерживает работу с кристаллическими кремниевыми типами панелей:

* **Кристаллические кремниевые панели** – монокристаллические и поликристаллические. Они наиболее распространены в производстве, поэтому их поддержка реализована в первую очередь.
  + 1. Качество и формат данных

Чтобы система корректно выявляла дефекты, изображения должны соответствовать определенным требованиям:

* **Разрешение** – не ниже 200х200, не выше 1920x1080 пикселей (Full HD). Большая разница между оригинальным изображением и изображением после обработки для корректной работы нейросети не должны быть слишком большими, так как это приведет к значительному снижению точности.
* **Форматы файлов** – поддерживаются JPEG, PNG.
* **Качество изображений** – снимки должны быть четкими, без размытия и шумов. Важно исключить засветку и резкие перепады освещения. Камера должна быть закреплена неподвижно, чтобы каждый кадр был снят под одинаковым углом.
* **Тип снимка** – снимки должны быть предоставлены в электролюминесцентном спектре.
  + 1. Ограничения модели машинного обучения

Модель обучена на выявлении производственных дефектов, таких как:

* Микротрещины в структуре панели
* Нарушения в структуре отдельных элементов
* Загрязнения и инородные включения

Также важно учитывать, что система адаптирована под конкретные производственные линии. Если в процессе работы изменится оборудование, угол съемки или освещение, потребуется перенастройка модели.

Еще один важный фактор – положение панели на конвейере. Если она будет смещена, повернута или расположена нестандартно, точность детекции может снизиться.

* + 1. Требования к оборудованию

Для стабильной работы системы необходимо следующее оборудование:

* **Камеры** – индустриального класса, способные предоставлять снимки в электролюминесцентном спектре.
* **Сервер для обработки изображений** – минимальные требования включают 8 ГБ оперативной памяти (RAM) и видеокарту с поддержкой CUDA и наличием не менее 6 ГБ видеопамяти (VRAM), так как обработка данных будет выполняться с использованием GPU.
  + 1. Скорость обработки

Чтобы соответствовать требованиям конвейерного производства, система должна анализировать одну панель за 20-30 секунд. Если обработка займет больше времени, это может вызвать задержки в производственном процессе.

Если на конвейере идет поток панелей, система должна поддерживать параллельную обработку нескольких изображений одновременно. Это необходимо для обеспечения бесперебойной работы линии.

ПРИЛОЖЕНИЕ

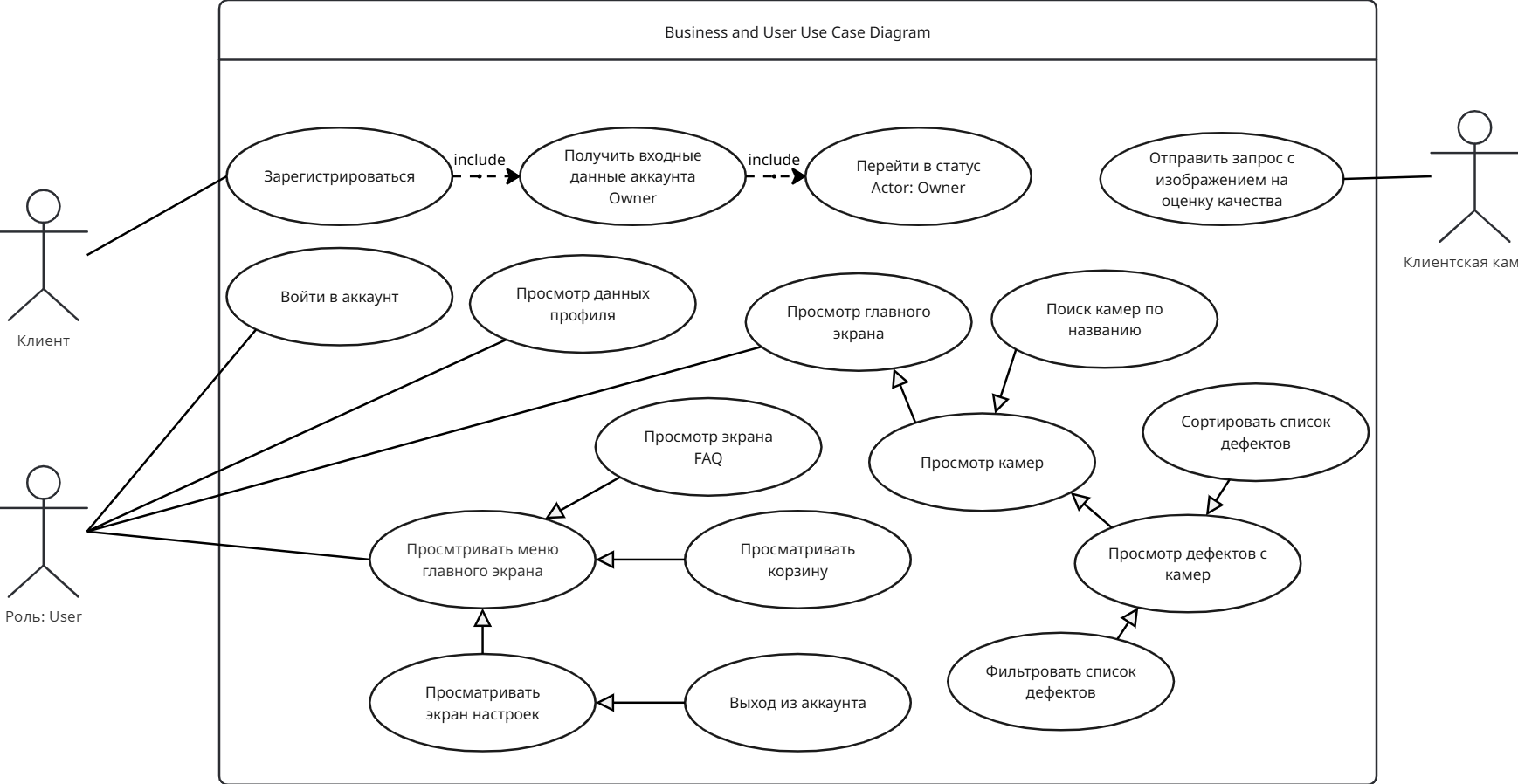


Рисунок 16 – Use case диаграмма

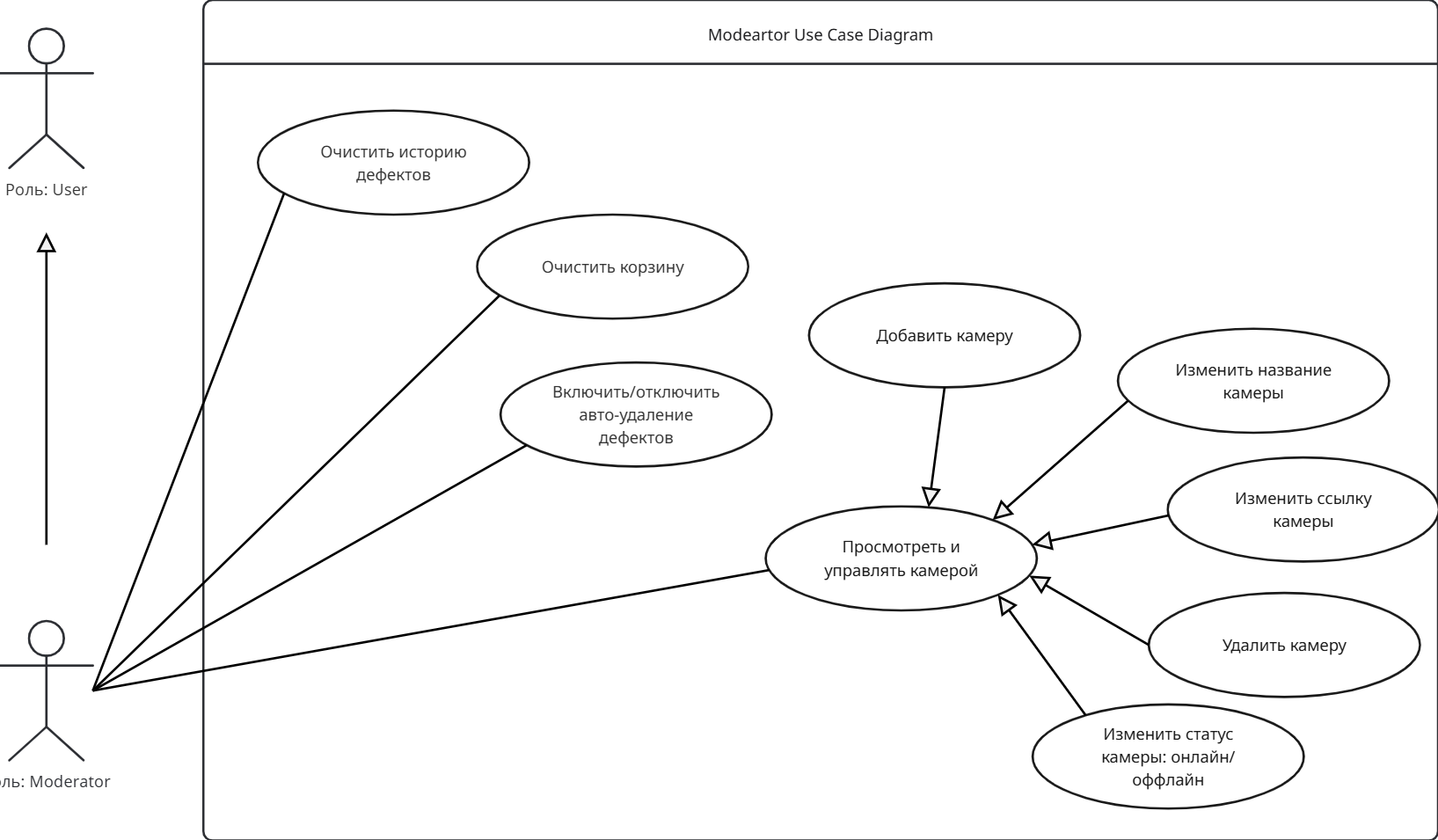


Рисунок 17 – Use case диаграмма модератор

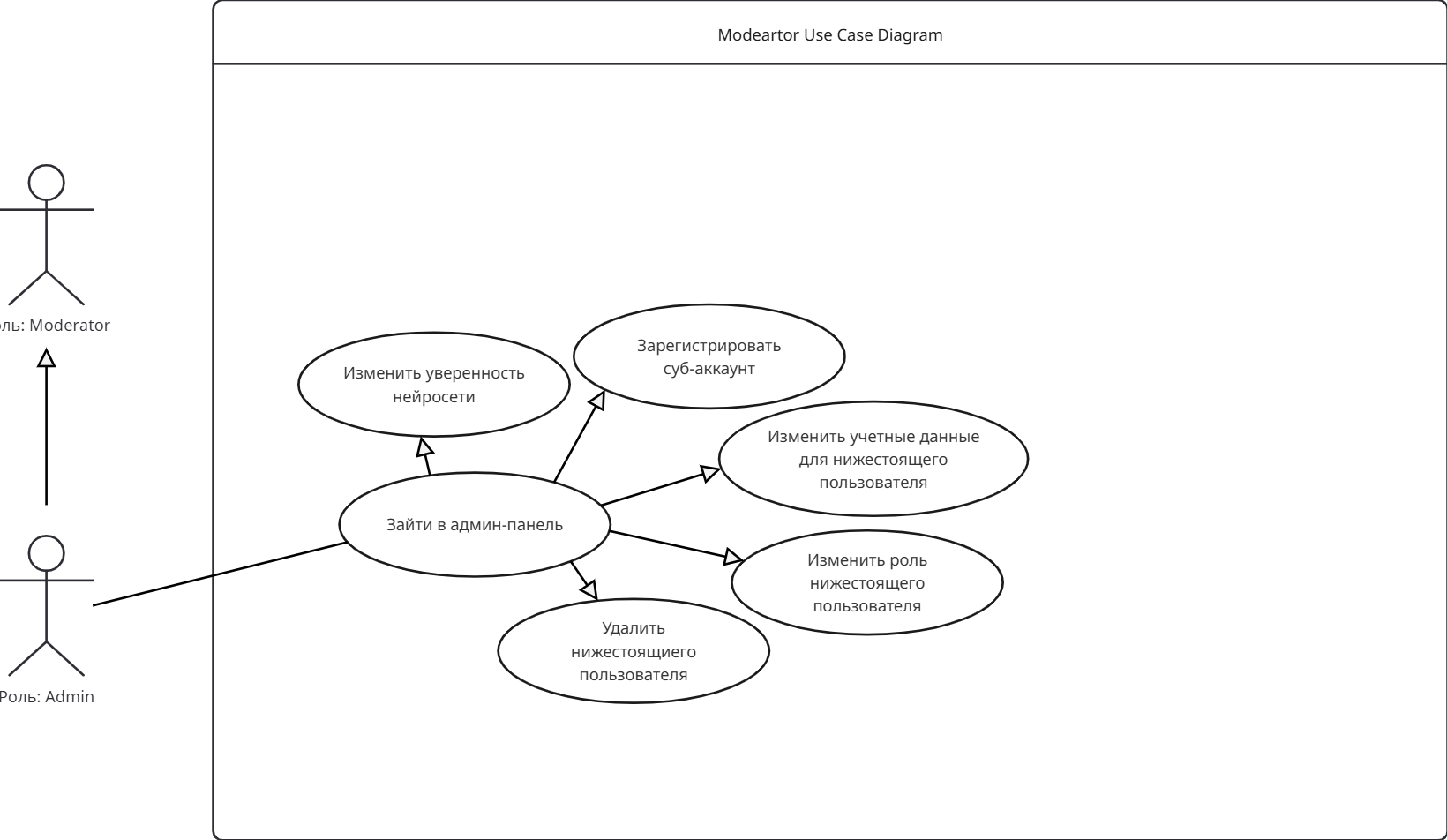


Рисунок 18 - Use case диаграмма модератор

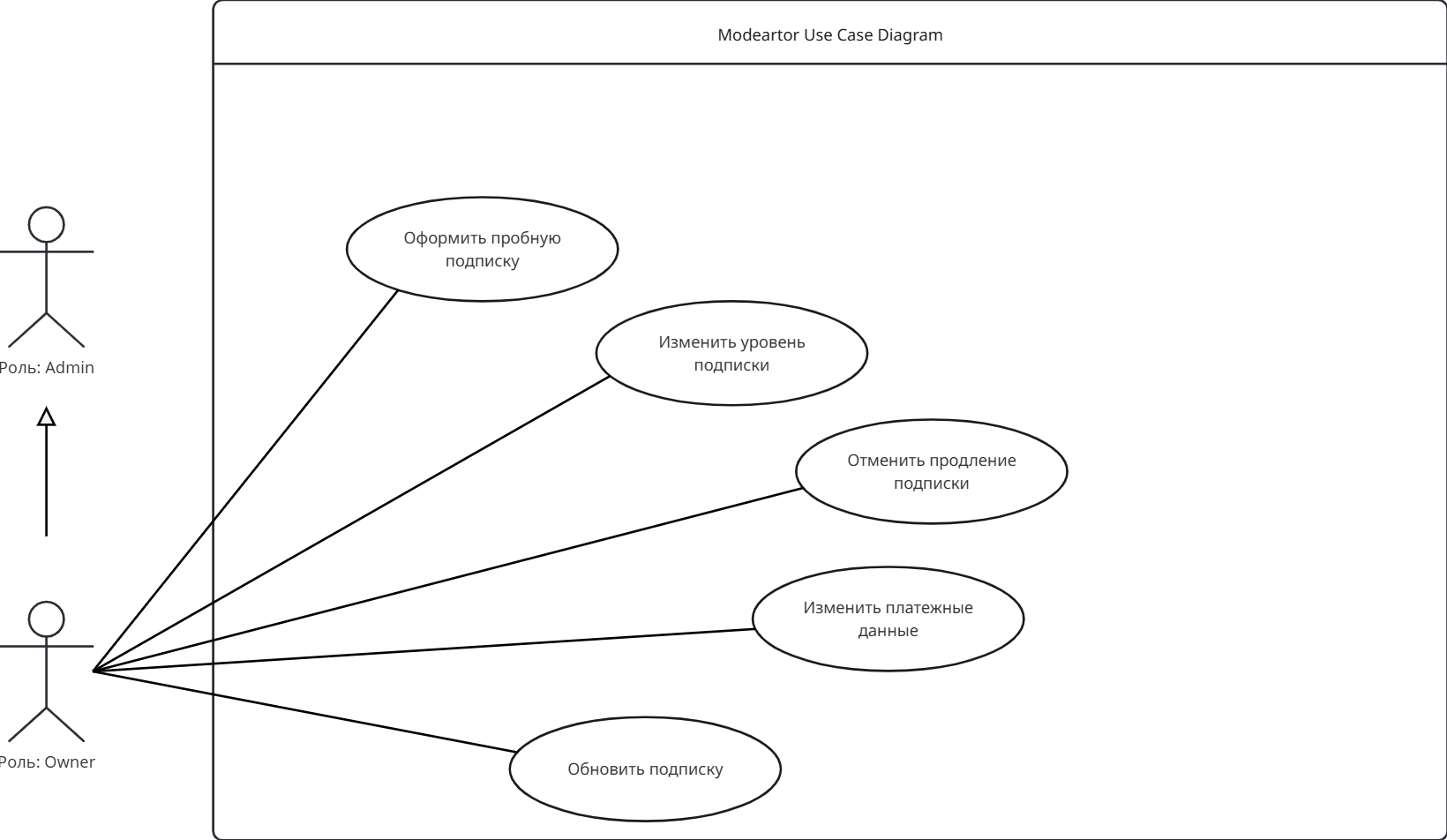


Рисунок 19 - Use case диаграмма модератор

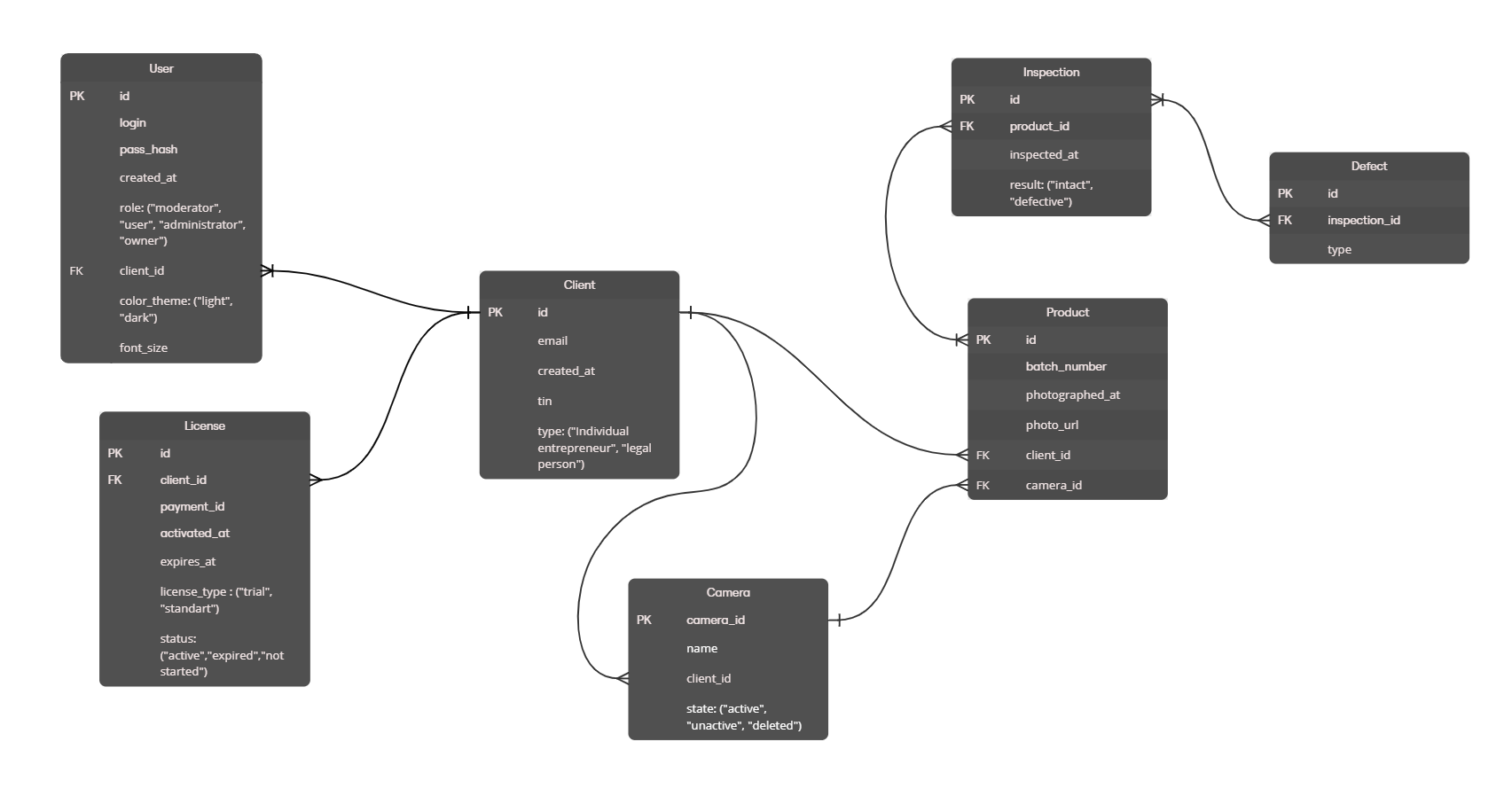


Рисунок 20 - ER-Диаграмма

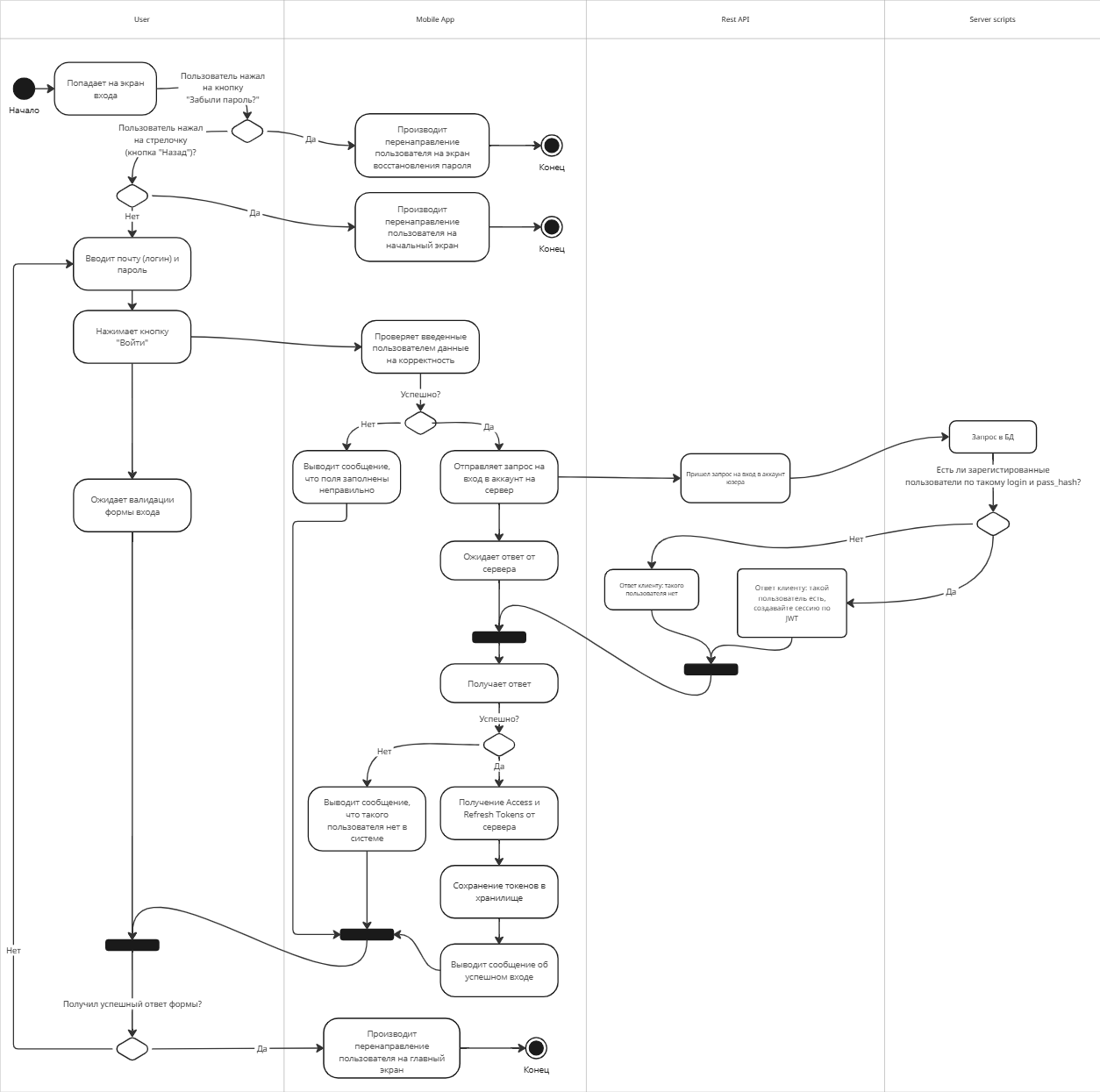


Рисунок 21 - Login диаграмма

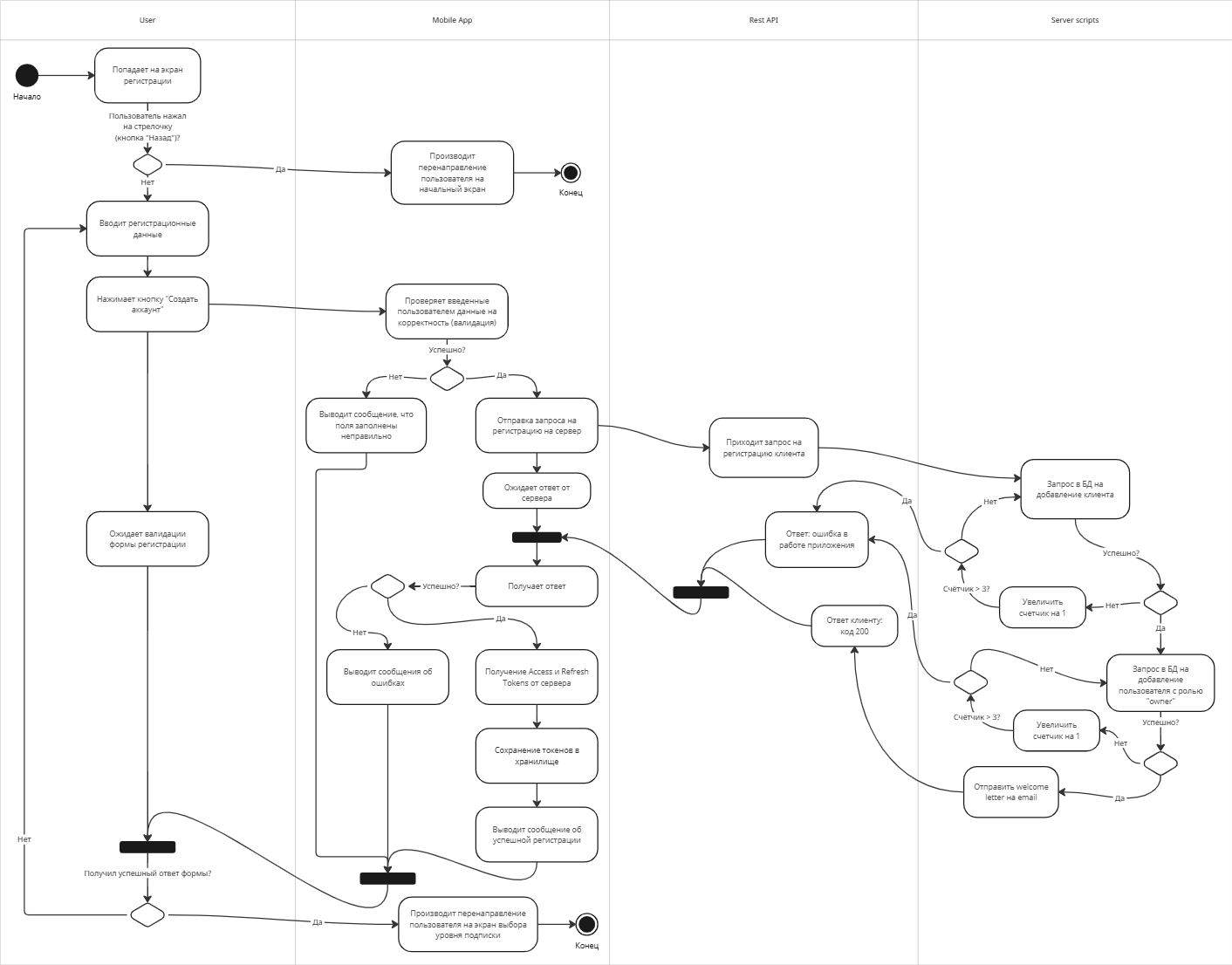


Рисунок 22 - Client registration диаграмма

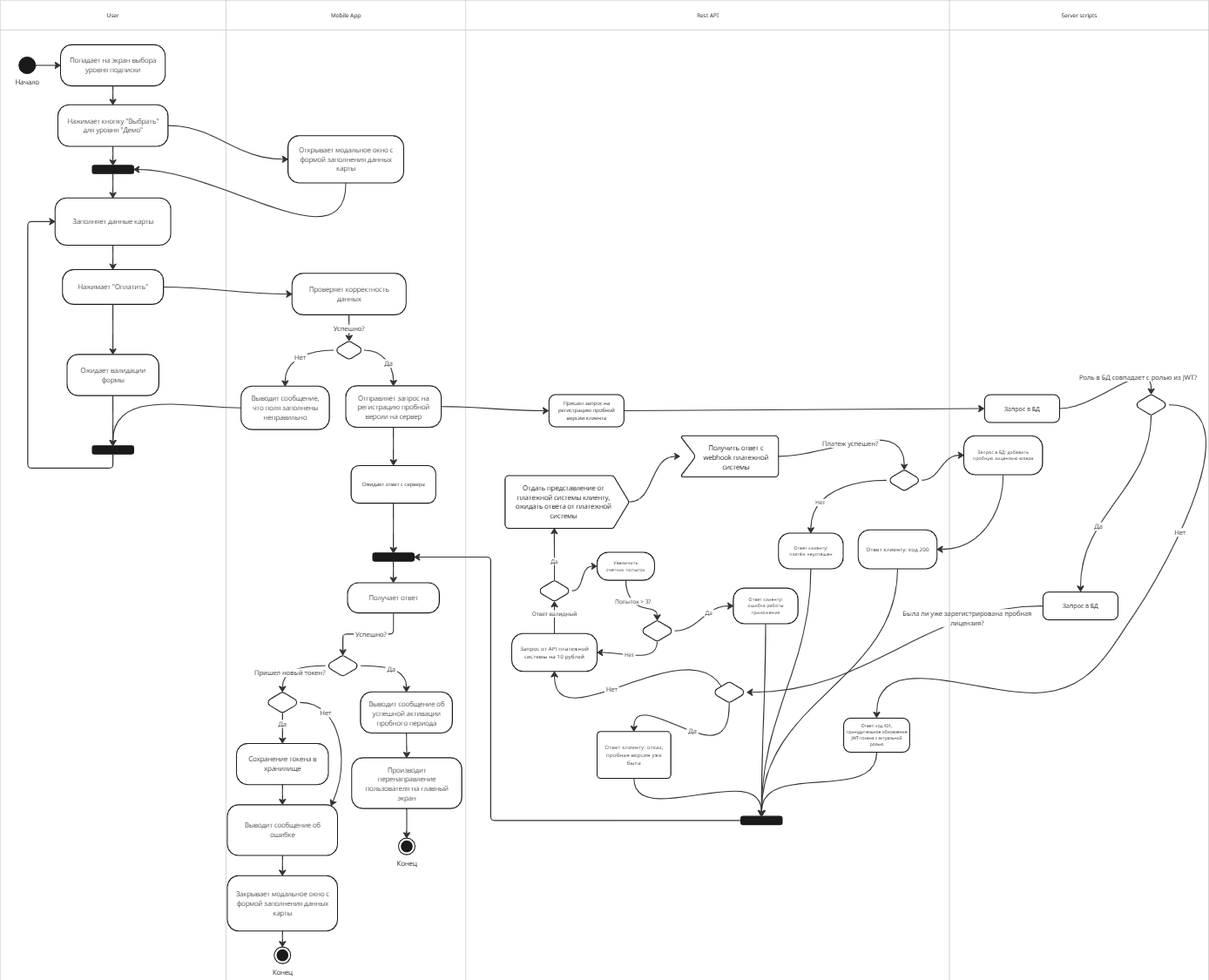


Рисунок 23 - Trial version registration диаграмма

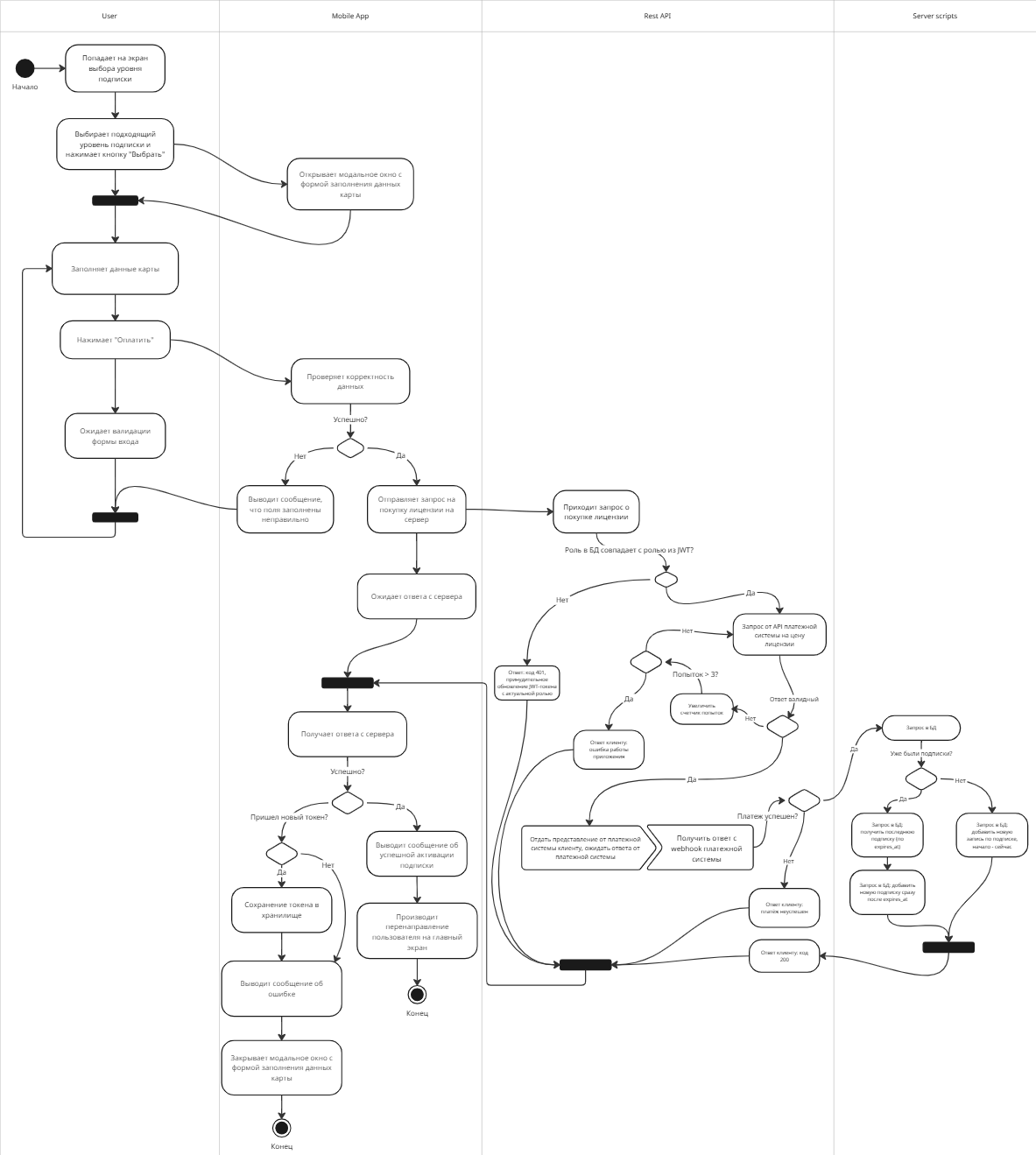


Рисунок 24 - License purchase диаграмма

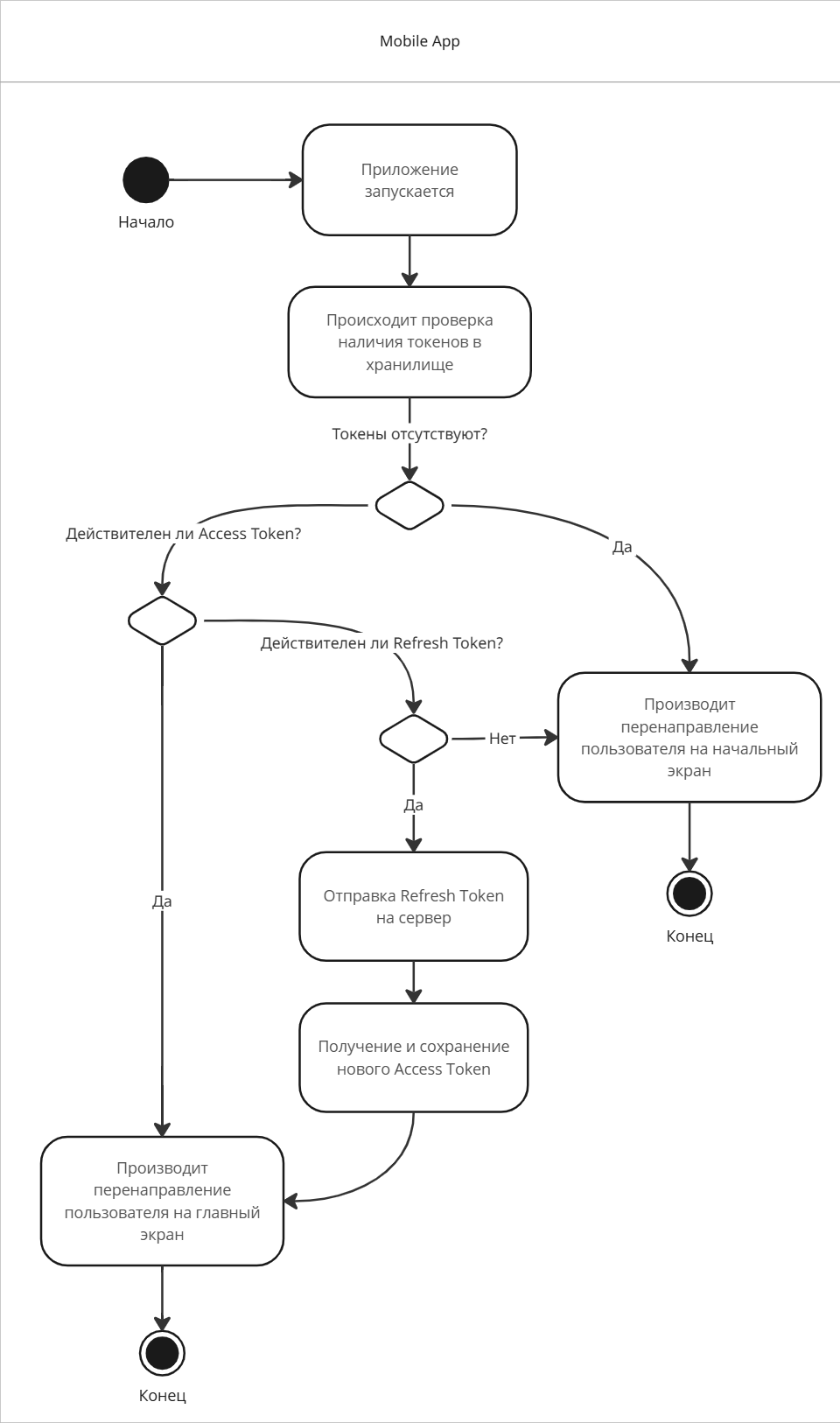


Рисунок 25 - Launch the app диаграмма

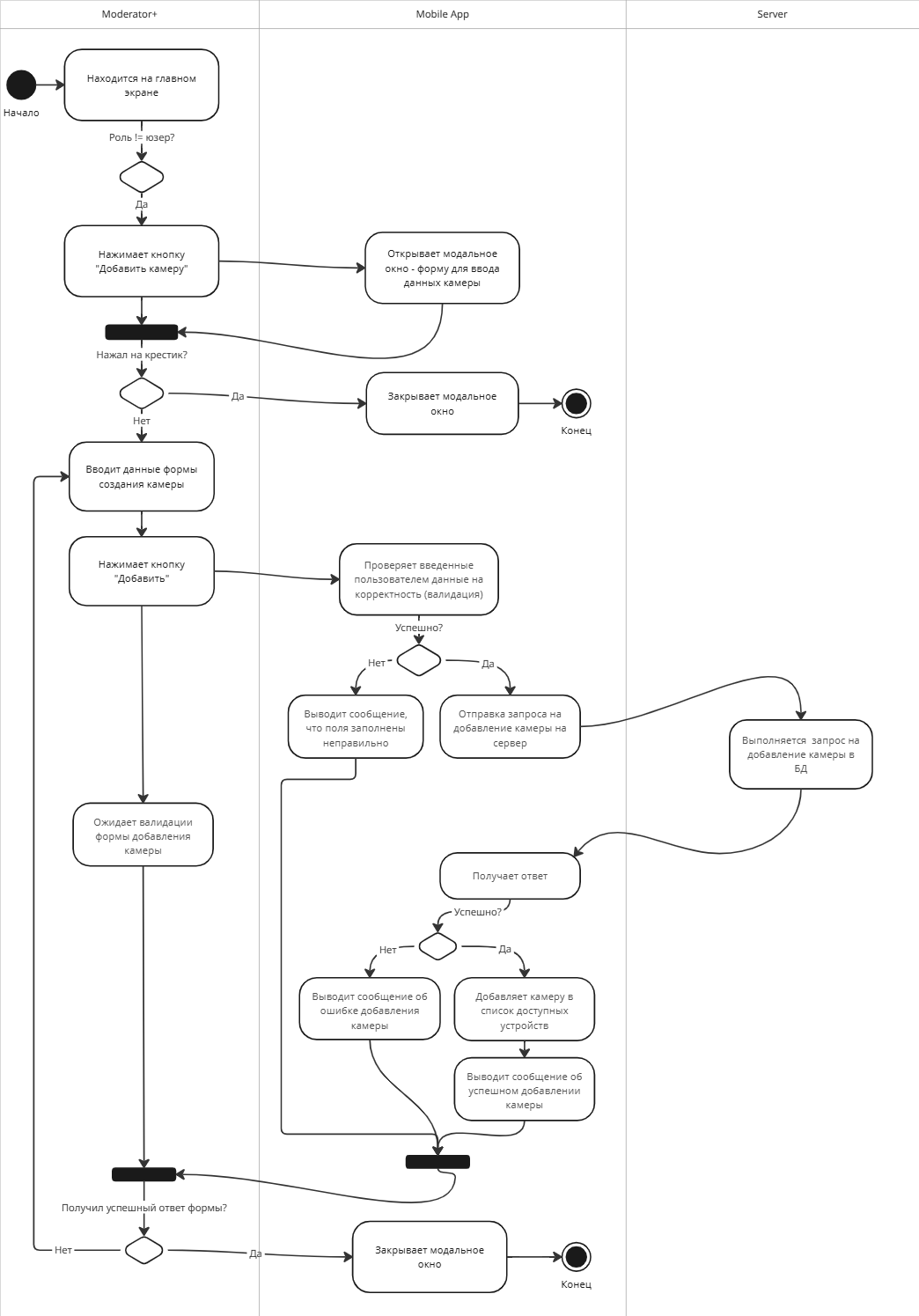


Рисунок 26 - Adding camera by moderator диаграмма

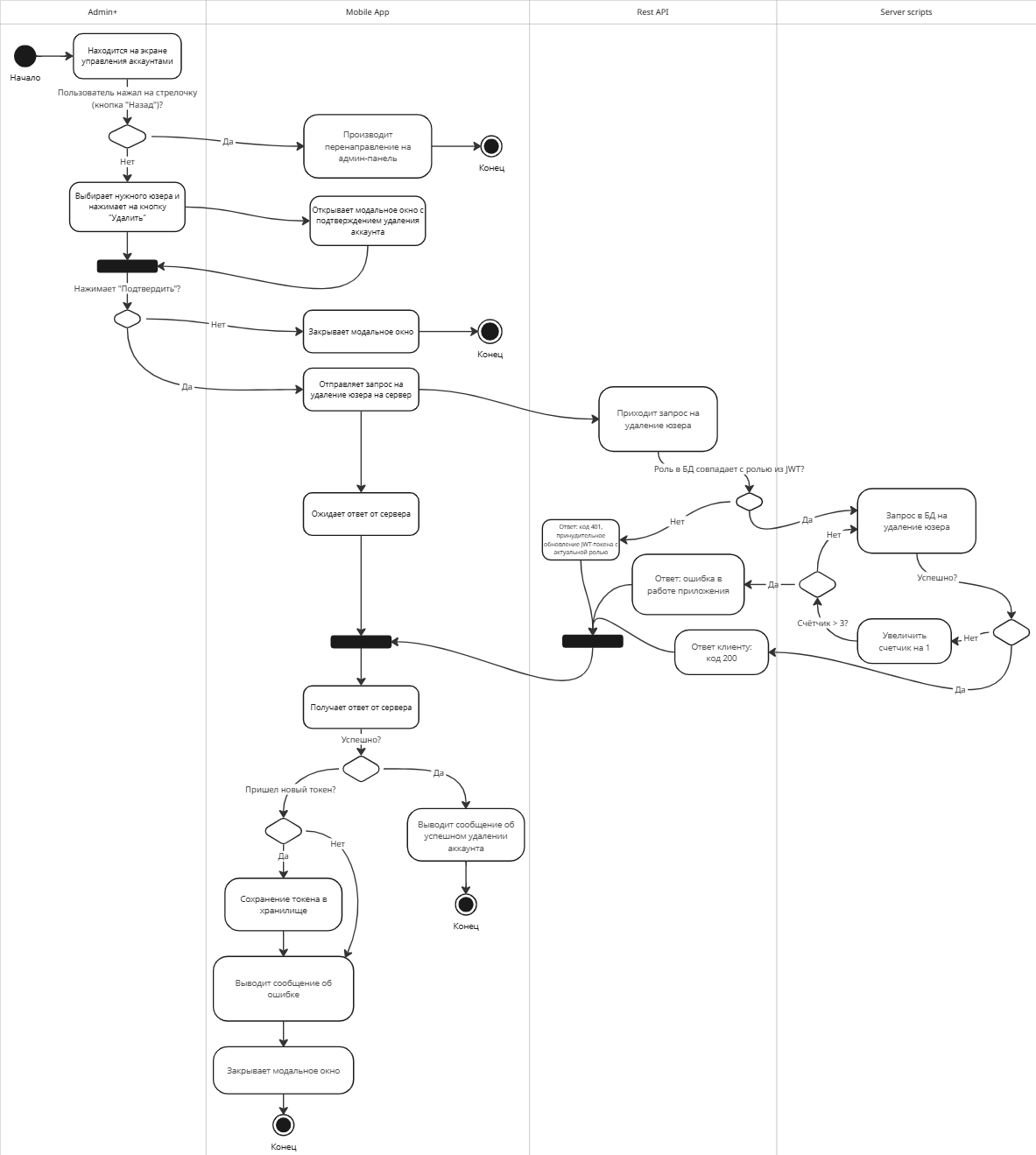


Рисунок 27 - User deleting by admin диаграмма

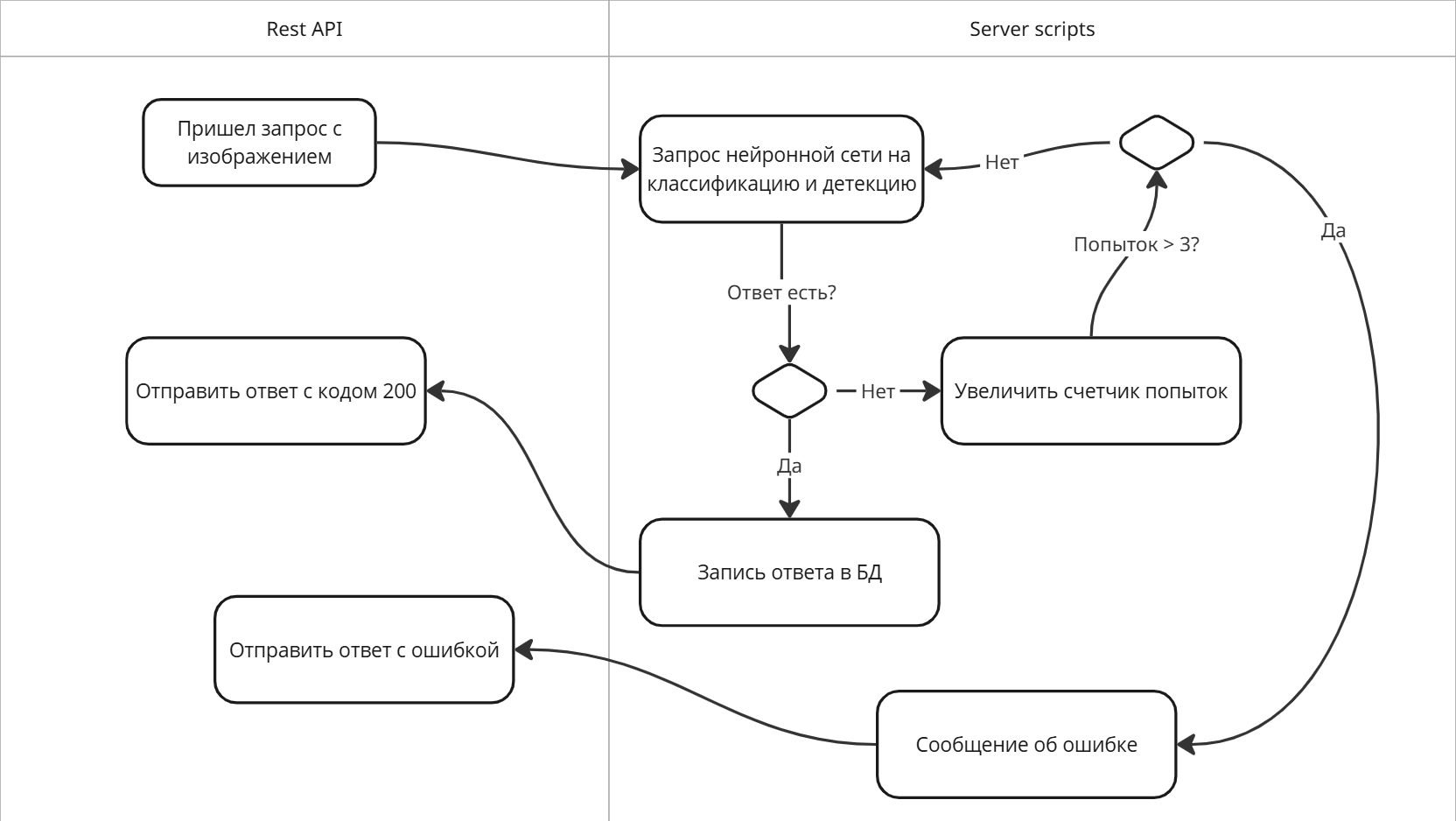


Рисунок 28 - Quality assesment диаграмма

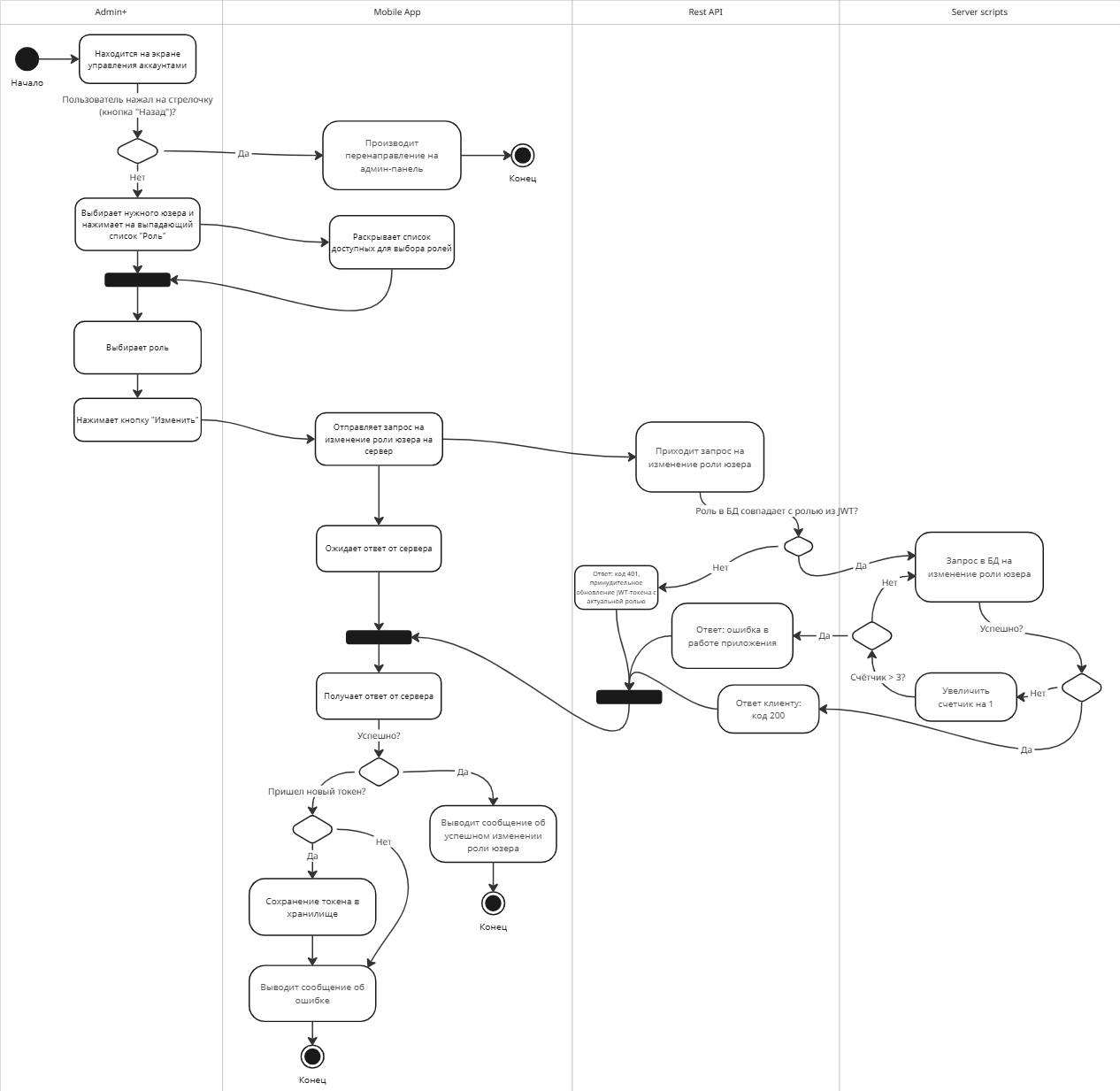


Рисунок 29 - Role changing by admin диаграмма

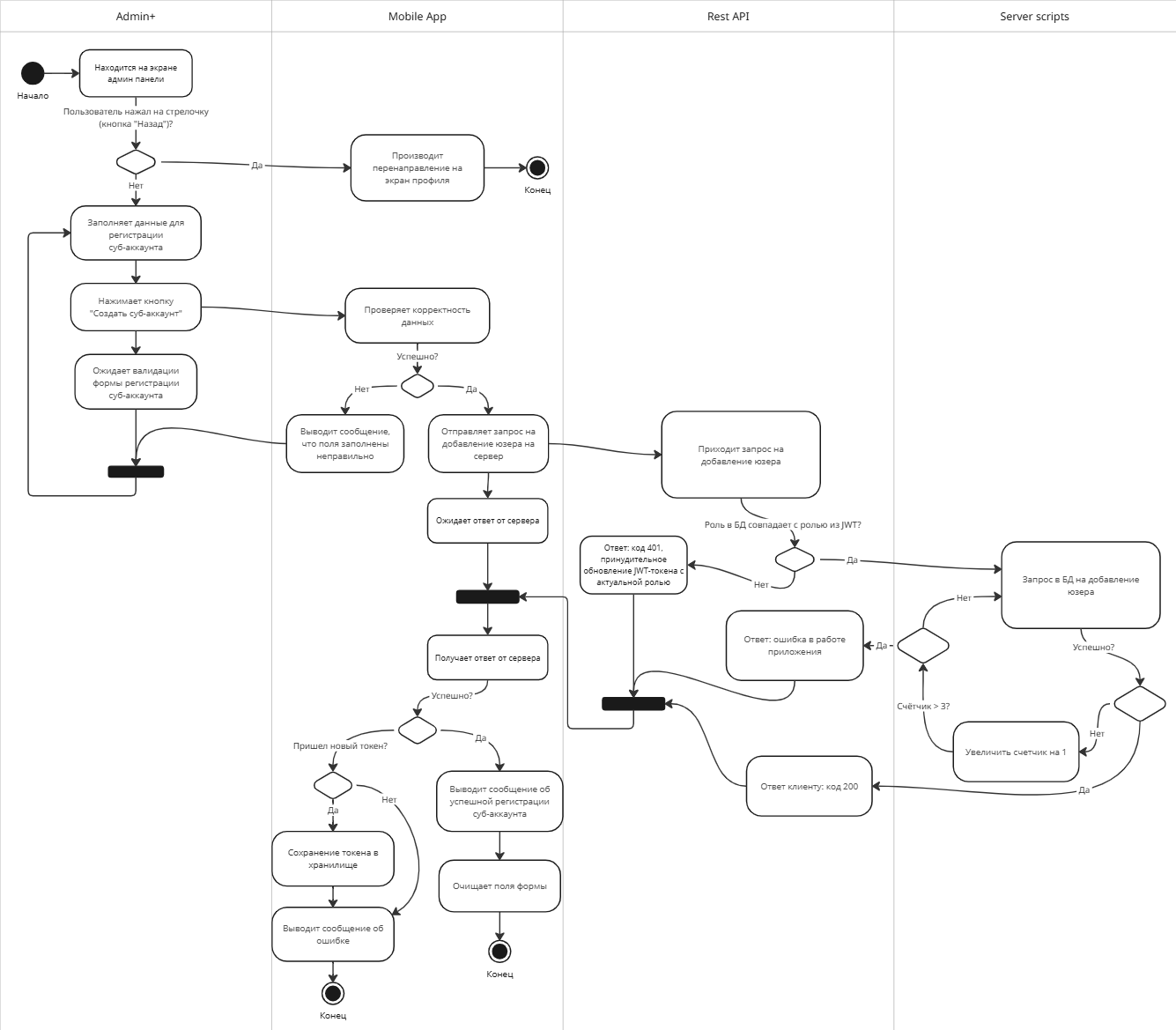


Рисунок 30 - User adding by admin диаграмма

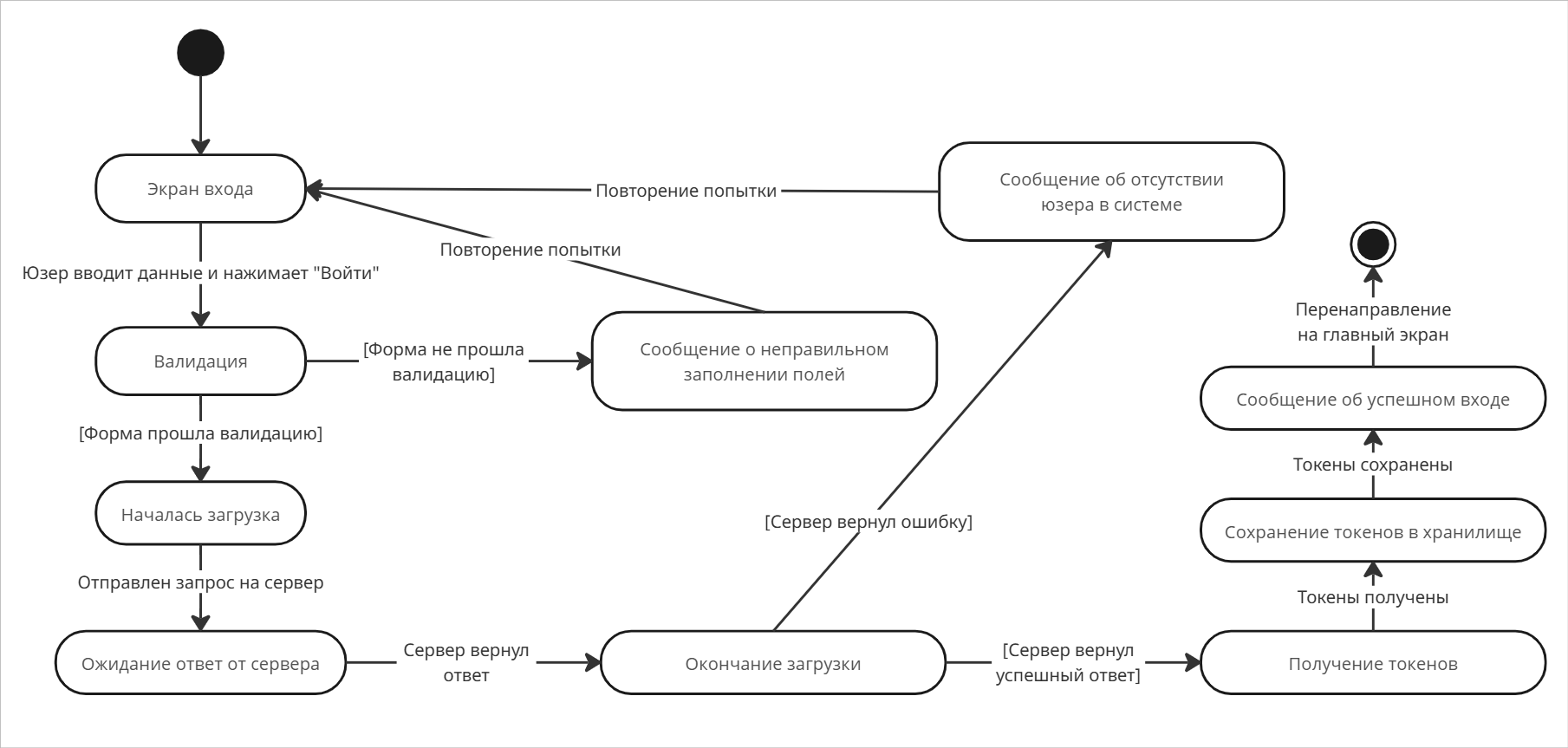


Рисунок 31 - Login screen state chart диаграмма

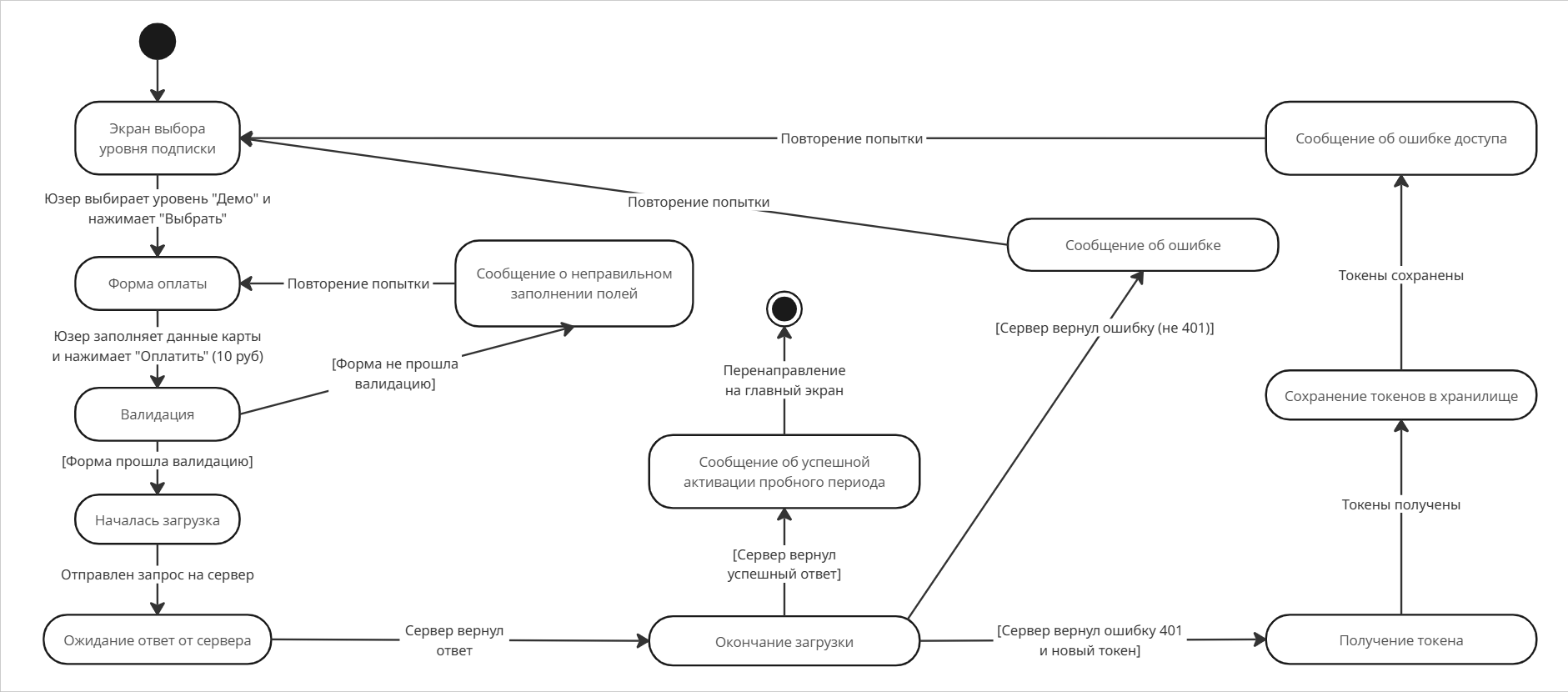


Рисунок 32 - Trial version registration state-chart диаграмма



Рисунок 33 - Client registration state-chart диаграмма

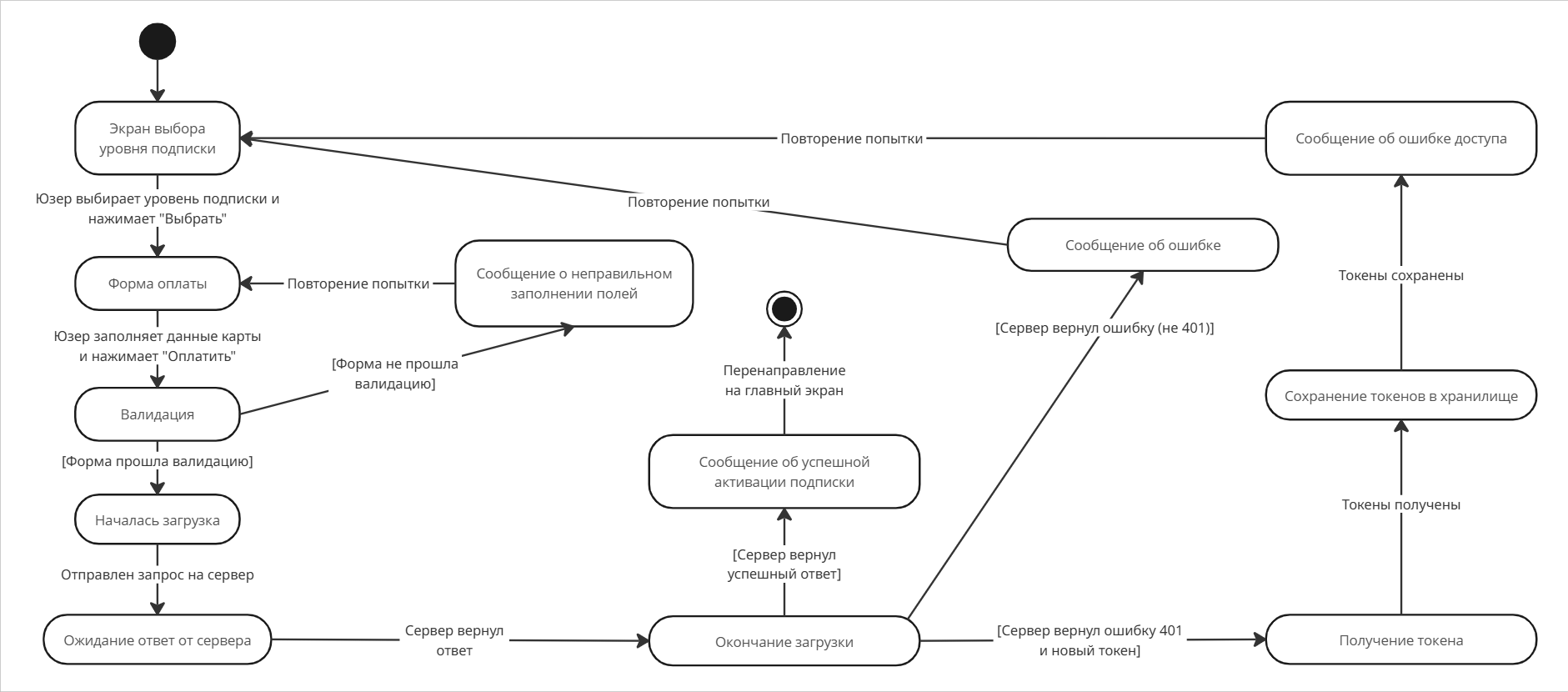


Рисунок 34 - License purchase state-chart диаграмма



Рисунок 35 - Adding camera by moderator state-chart диаграмма

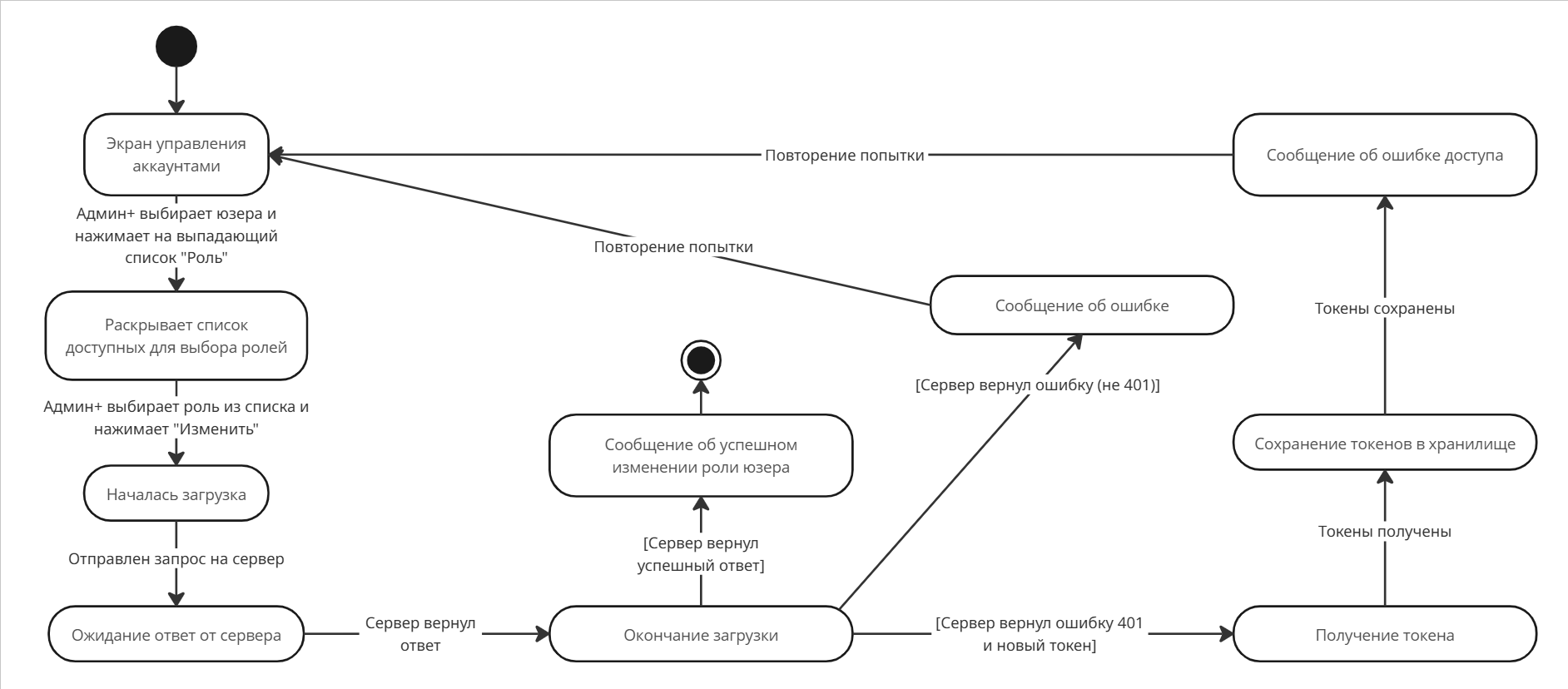


Рисунок 36 - Role changing by admin state-chart диаграмма

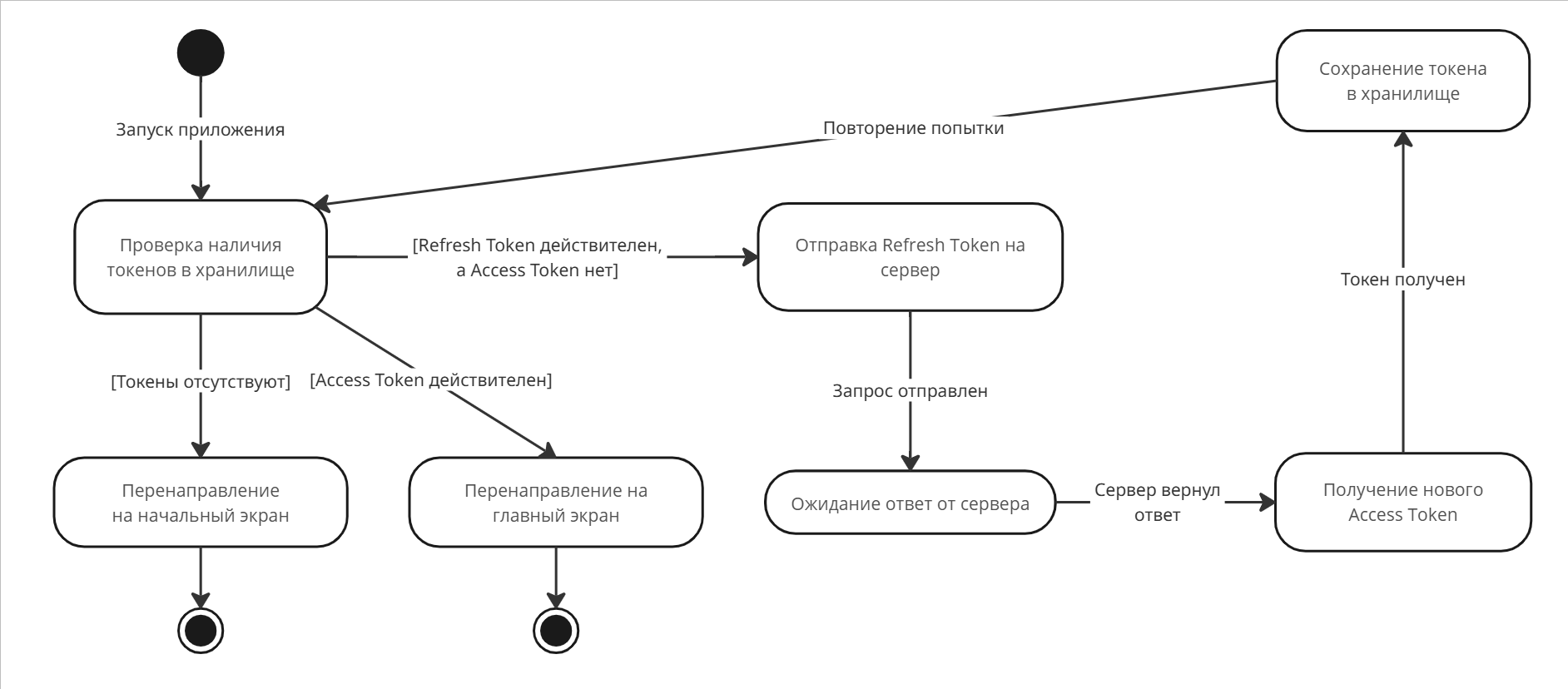


Рисунок 37 - App launching state-chart диаграмма

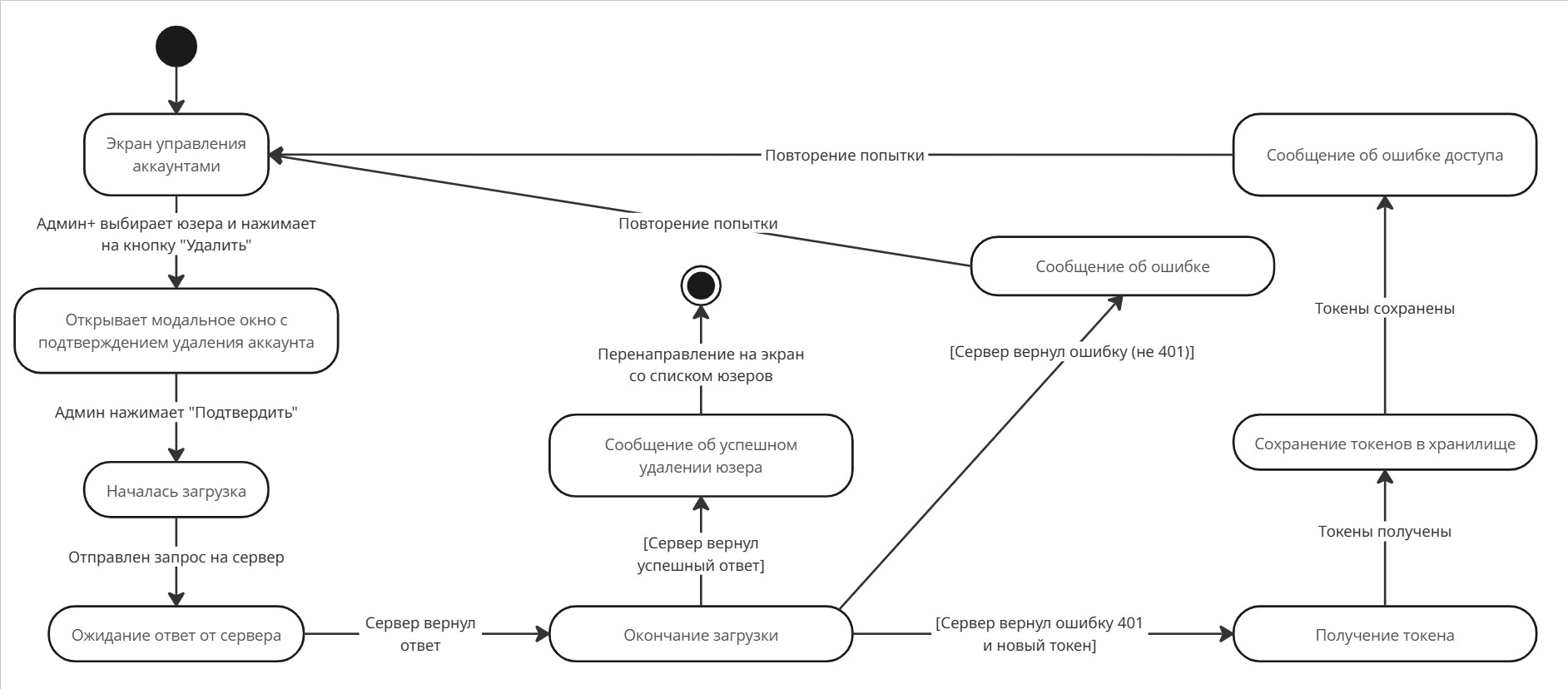


Рисунок 38 - User deleting by admin state-chart диаграмма

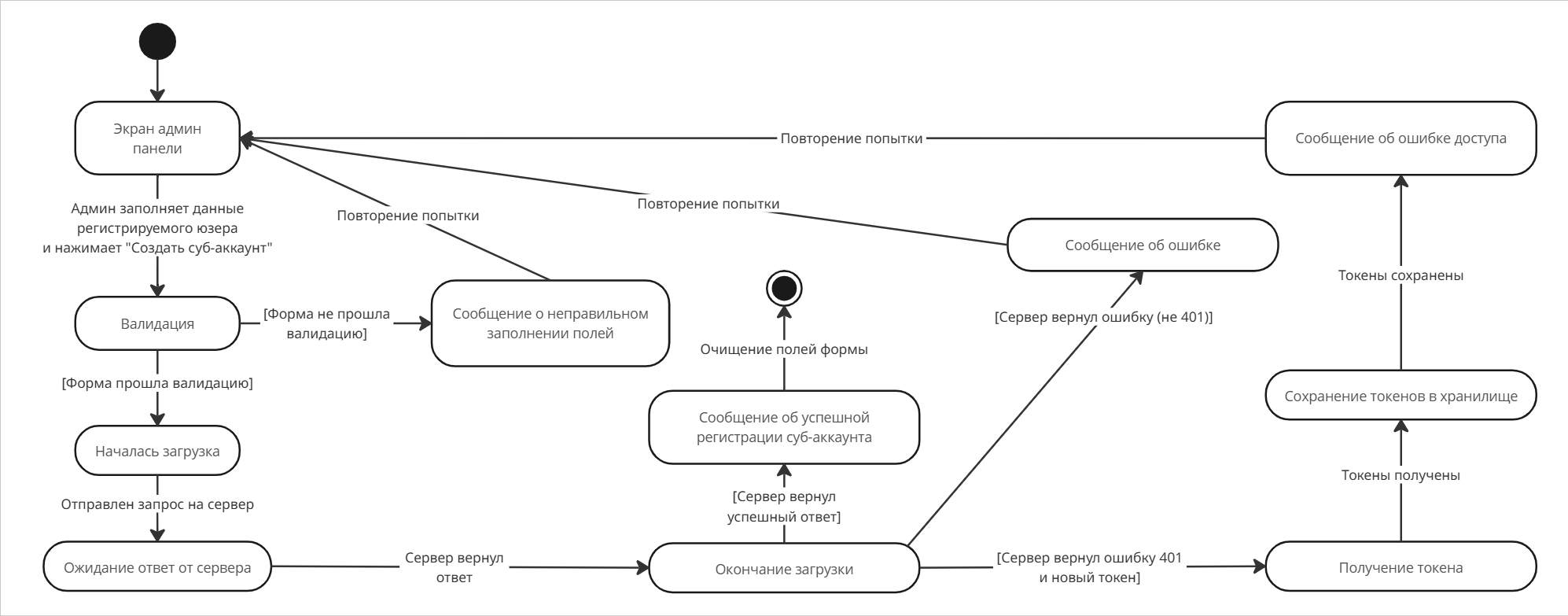


Рисунок 39 - User adding by admin state-chart диаграмма

|  |  |  |
| --- | --- | --- |
| **Период** | **Разработа основного приложения** | **Создание нейросетевого алгоритма** |
| **31.03-05.04** | Проектирование архитектуры системы. Формирование структуры базы данных. Определение моделей данных. | Утверждение архитектуры нейросети. |
| **07.04-12.04** | Создание функций главного экрана мобильного приложения. | Подготовка исходных данных. |
| **14.04-19.04** | Написание алгоритма детектирования дефектов на солнечных панелях. | Создание прототипа нейросети. |
| **21.04-26.04** | Создание интерфейса администратора. Реализация возможностей модератора, юзера, администратора и овнера. | Интеграция нейросети с сервером приложения. |
| **26.04** | **Представление MVP проекта** | |
| **28.04-03.05** | Разработка системы регистрации и входа пользователей. Разработка функционала для профиля, корзины и экрана настроек. | Исследование статистики работы приложения. |
| **05.05-10.05** | Подключение системы оплаты подписки. Разработка функционала экрана FAQ. |  |
| **12.05-17.05** | Разработка основного функционала камер, создания отчёта. |  |
| **19.05-24.05** | Разработка системы мульти-уровневых подписок, мульти-ролевого доступа, админ панели. |  |
| **26.05-31.05** | Подгтовка проекта к деплою. | Подгтовка проекта к деплою. |
| **31.05** | **Итоговая презентация проекта** | |

Таблица 1 – Дорожная карта