

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра технологий обработки и защиты информации

Курсовая работа
Разработка мобильного приложения
«Система для учета и анализа питания домашних животных, с рекомендациями
по кормлению и уходу»

Зав. кафедрой _____ Сирота А.А., д.т.н., профессор
Обучающийся _____ Блинова О.А., 3 курс, д/о
Обучающийся _____ Семенов А.А., 3 курс, д/о
Обучающийся _____ Вдовикова Е.Ю., 3 курс, д/о
Руководитель _____ Тарасов В.С., ст. преподаватель

Воронеж 2024

Содержание

Введение	4
1 Постановка задачи	5
1.1 Цели создания системы	5
1.2 Задачи проекта	5
1.3 Функциональные и нефункциональные требования	5
1.3.1 Функциональные требования	5
1.3.2 Нефункциональные требования	5
2 Анализ предметной области	7
2.1 Терминология	7
2.2 Обзор аналогов	8
2.2.1 11pets	9
2.2.2 Animal ID – Защита и Уход	10
2.2.3 Petapet	11
2.3 Моделирование системы	12
2.3.1 Диаграмма в стиле методологии IDEF0	12
2.3.2 Диаграмма прецедентов	13
2.3.3 Диаграмма последовательности	14
2.3.4 Диаграмма развертывания	16
2.3.5 Диаграмма состояния	16
2.3.6 Диаграмма объектов	18
2.3.7 Диаграмма активности	19
3 Реализация	23
3.1 Средства реализации	23
3.2 Реализация базы данных	24
3.2.1 ER-диаграмма базы данных	24
3.3 Реализация клиентской части	25
3.3.1 Форма экрана «Вход»	26
3.3.2 Форма экрана «Регистрация»	27
3.3.3 Форма экрана «Профиль»	28
3.3.4 Форма экрана «Настройки»	28

3.3.5	Форма экрана «Советы»	29
3.3.6	Форма экрана «Еда»	30
3.4	Серверная часть.....	31
3.4.1	Общее описание проекта.....	32
3.4.2	Папка Pets	33
3.4.3	Папка Users	34
3.4.4	Папка Articles	34
3.4.5	Папка Config.....	35
3.4.6	Папка Static	35
4	Аналитика	36
	Заключение	39
	Список использованных источников	40

Введение

В современном мире забота о домашних животных является важным аспектом для многих людей. Для облегчения этого процесса и обеспечения наших питомцев правильным уходом и кормлением, разрабатывается специализированная система, предоставляющая удобный доступ к необходимой информации и рекомендациям.

Система для учета и анализа питания домашних животных представляет собой программное приложение для мобильных устройств, которое помогает владельцам животных отслеживать рацион своих питомцев, анализировать их потребности в питании и получать рекомендации по кормлению и уходу.

В данной курсовой работе рассматривается процесс разработки собственного мобильного приложения.

В рамках работы будут рассмотрены различные аспекты разработки мобильного приложения, начиная с анализа предметной области, определения его концепции и основных требований. Затем будет изучено проектирование пользовательского интерфейса и пользовательского опыта, с учетом современных тенденций и личных практик в этой области. Важное внимание будет уделено выбору и интеграции соответствующих технологий и API для обеспечения необходимых функций.

Эта система поможет владельцам домашних животных эффективно управлять питанием и уходом за своими питомцами, обеспечивая им здоровье и комфорт.

1 Постановка задачи

1.1 Цели создания системы

Целью данной работы является создание мобильного приложения, которое поможет владельцам животных следить за рационом кормления, анализировать состав кормов, получать рекомендации и улучшать заботу о питомце.

1.2 Задачи проекта

1. учет питания;
2. анализ корма;
3. мониторинг здоровья;
4. получение полезной информации по теме домашних животных.

1.3 Функциональные и нефункциональные требования

1.3.1 Функциональные требования

1. Система должна предоставлять возможность ввода информации о питании питомца, включая тип корма, количество.
2. Система должна предоставлять возможность ввода информации о домашнем питомце, включая его вид, породу, возраст, вес и другие параметры
3. Система должна предоставлять возможность анализа питания.
4. Система должна предоставлять возможность ведения учета изменений веса питомца и активности для премиум пользователей.
5. Система должна предоставлять возможность просмотра рекомендаций по темам: «Еда», «Здоровье», «Дом и игры» и «Помощь».

1.3.2 Нефункциональные требования

1. Система должна быть легкодоступной для пользователей и иметь интуитивно понятный интерфейс.
2. Система должна быть легкодоступной для пользователей и иметь интуитивно понятный интерфейс.

3. Система должна обеспечивать безопасность и конфиденциальность данных о питомце и его владельце.
4. Система должна быть надежной и обеспечивать непрерывную работу без сбоев и ошибок.
5. Система должна быть масштабируемой и обеспечивать возможность расширения функционала в будущем.
6. Система должна обеспечивать быструю обработку и анализ данных.

2 Анализ предметной области

2.1 Терминология

Таблица 1 - Глоссарий

Мобильное приложение	Программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы.
Клиент	Это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.
Сервер	Выделенный или специализированный компьютер для выполнения сервисного программного обеспечения.
База данных	Это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных.
HTTPS	Это протокол безопасной передачи гипертекста, который используется для обеспечения безопасной связи между клиентом и сервером
Android	Это операционная система с

	открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux.
Фреймворк	Программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.
Аккаунт	Это персональная страница пользователя или личный кабинет, который создается после регистрации на сайте.
Django REST framework	Свободный фреймворк для разработки на языке Python.
PostgreSQL	Объективно-реляционная система управления базами данных.
Dart	Язык программирования, разработанный Google, который используется для создания веб-приложений, мобильных приложений и серверных приложений.
Flutter	Это открытая среда разработки от Google для создания кроссплатформенных мобильных, веб- и настольных приложений с использованием одного кода.

2.2 Обзор аналогов

Разрабатывая приложение, основной задачей которого является учет и анализ питания, необходимо рассматривать разработку с точки зрения актуальности и уникальности проекта. Для оценки этих качеств необходимо прибегнуть к рассмотрению аналогов разрабатываемого приложения,

адекватно оценивая все положительные и негативные черты того или иного продукта.

2.2.1 11pets

11pets — это приложение для владельцев питомцев, предоставляющее простой и удобный доступ, широкий спектр функций, облегчающих уход за питомцами и помогающих отслеживать их здоровье и благополучие. Главная цель приложения — сделать жизнь с питомцем более организованной, продуктивной и приятной для владельцев.

11pets обладает широким спектром предоставляемых услуг и с точки зрения авторского контента. На рисунке 1 представлен главный экран данного приложения.

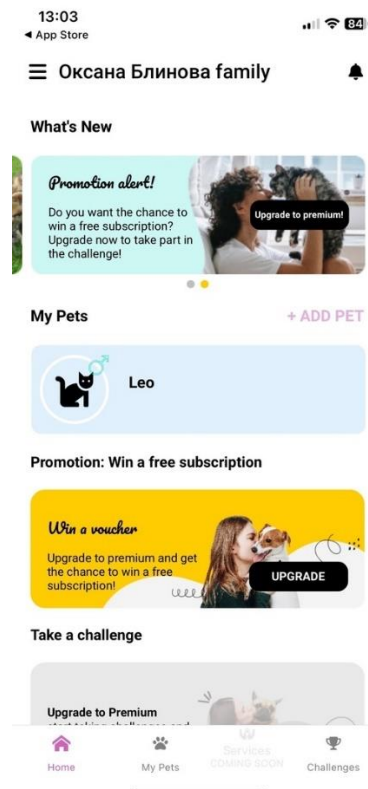


Рисунок 1 - Интерфейс страницы «11pets»

Недостатки:

- невозможно выполнить процесс авторизации или регистрации в приложении, так как операция перенаправляет пользователя на веб-сайт;
- в приложении отсутствует поддержка русского языка;
- возможность добавлять только одного питомца;

- могут возникать технические неполадки и сбои.

2.2.2 Animal ID – Защита и Уход

Animal ID — это международная онлайн-платформа для ответственных владельцев домашних питомцев, которые заботятся о безопасности своего любимца.

В данном приложении вы можете сохранять ветеринарные документы, родословную и фотографии животного, а при наличии идентификаторов – QR паспорта и микрочипа – легко найти своего питомца, если тот убежал и потерялся. Также, в приложении действует кэшбек-система на добрые дела в виде помощи бездомным животным из приютов. Программа имеет простой, приятный интерфейс, полезные функции для вас и вашего питомца, и платный контент.

На рисунке 2 представлен главный экран данного приложения.

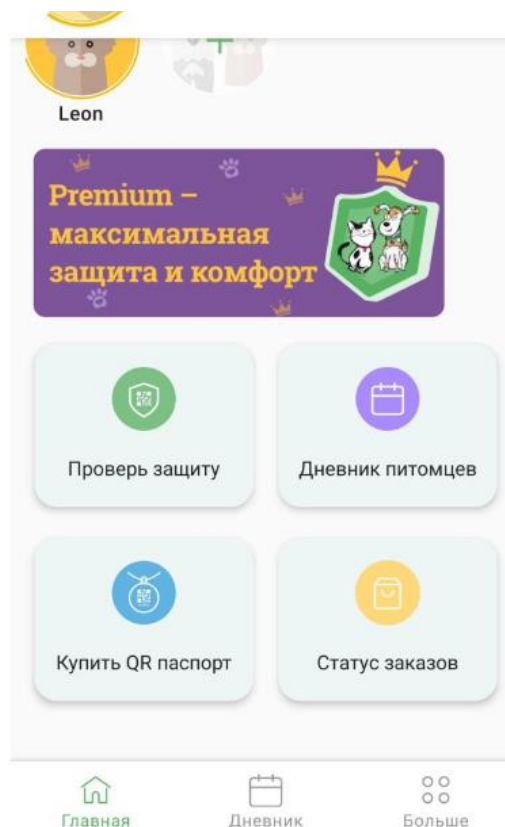


Рисунок 2 - Интерфейс страницы «Animal ID»

Недостатки:

- платные услуги;

- доступно только на Android;
- ограниченная поддержка видов питомцев;
- ограниченная географическая доступность.

2.2.3 Petapet

Petapet — это уникальное приложение-дневник для домашних животных, созданное с целью облегчить и сделать более приятной жизнь владельцев питомцев. С помощью Petapet вы можете эффективно управлять всеми аспектами жизни вашего любимца, отслеживать его здоровье, рост и развитие.

В приложении доступны следующие функции:

1. отслеживание событий: Petapet позволяет вам отмечать и планировать различные события в жизни вашего питомца, такие как визиты к ветеринару, вакцинацию и другое;
2. отслеживание веса: Petapet предоставляет удобный инструмент для отслеживания веса вашего питомца. Вы можете регулярно вводить данные о весе и наблюдать за динамикой изменений на графике;
3. написание заметок: в приложение есть возможность создавать заметки о питомце, фиксируя интересные моменты, достижения и другое;
4. полезные советы и статьи: в Petapet вы найдете множество полезной информации о здоровье, уходе, тренировках и питании домашних животных.

На рисунке 3 представлен главный экран данного приложения.

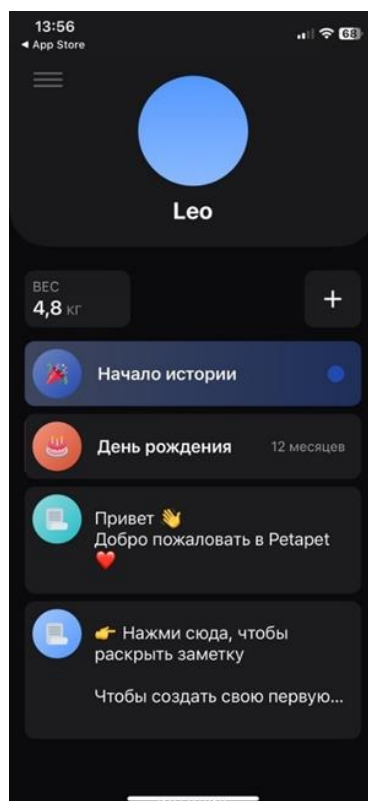


Рисунок 3 - Интерфейс страницы «Petapet»

Недостатки:

- частая ошибка, при которой приложение прерывает свою работу;
- ограниченная поддержка видов питомцев;
- отсутствие синхронизации с внешними устройствами;
- недостаточная персонализация;
- отсутствие напоминаний о кормлении.

2.3 Моделирование системы

2.3.1 Диаграмма в стиле методологии IDEF0

IDF0 диаграмма представляет собой графическое представление бизнес-процесса в виде иерархической структуры функций. Основная цель IDF0 диаграммы состоит в том, чтобы показать, как различные функциональности взаимодействуют друг с другом и как они влияют на достижение целей организации. Она помогает улучшить понимание процессов и оптимизировать их для повышения эффективности организации. Данная диаграмма представлена на рисунке 4.

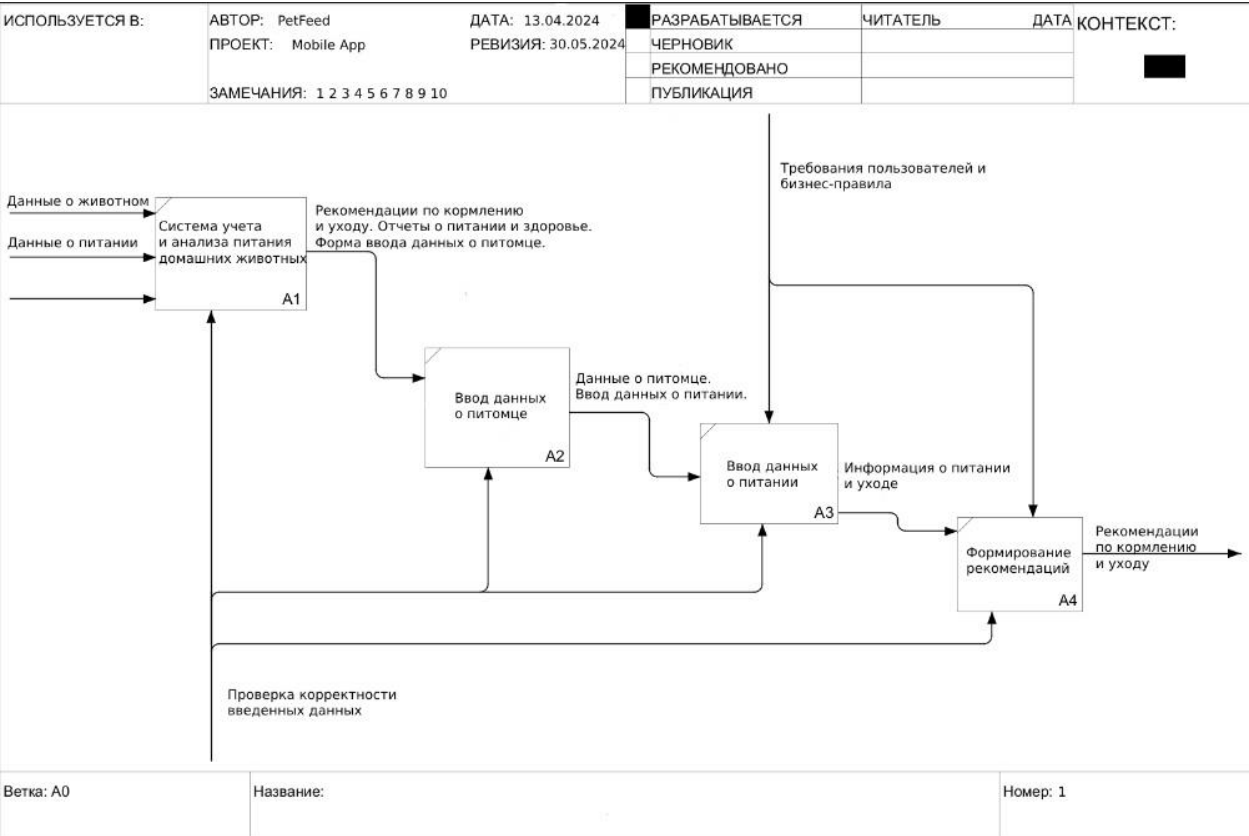


Рисунок 4 - Диаграмма в стиле методологии IDEF0

2.3.2 Диаграмма прецедентов

Рассмотрим полную диаграмму для использования приложения разными типами пользователей. В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Они помогают в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определенных целей. Они также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы. Данная диаграмма представлена на рисунке 5.

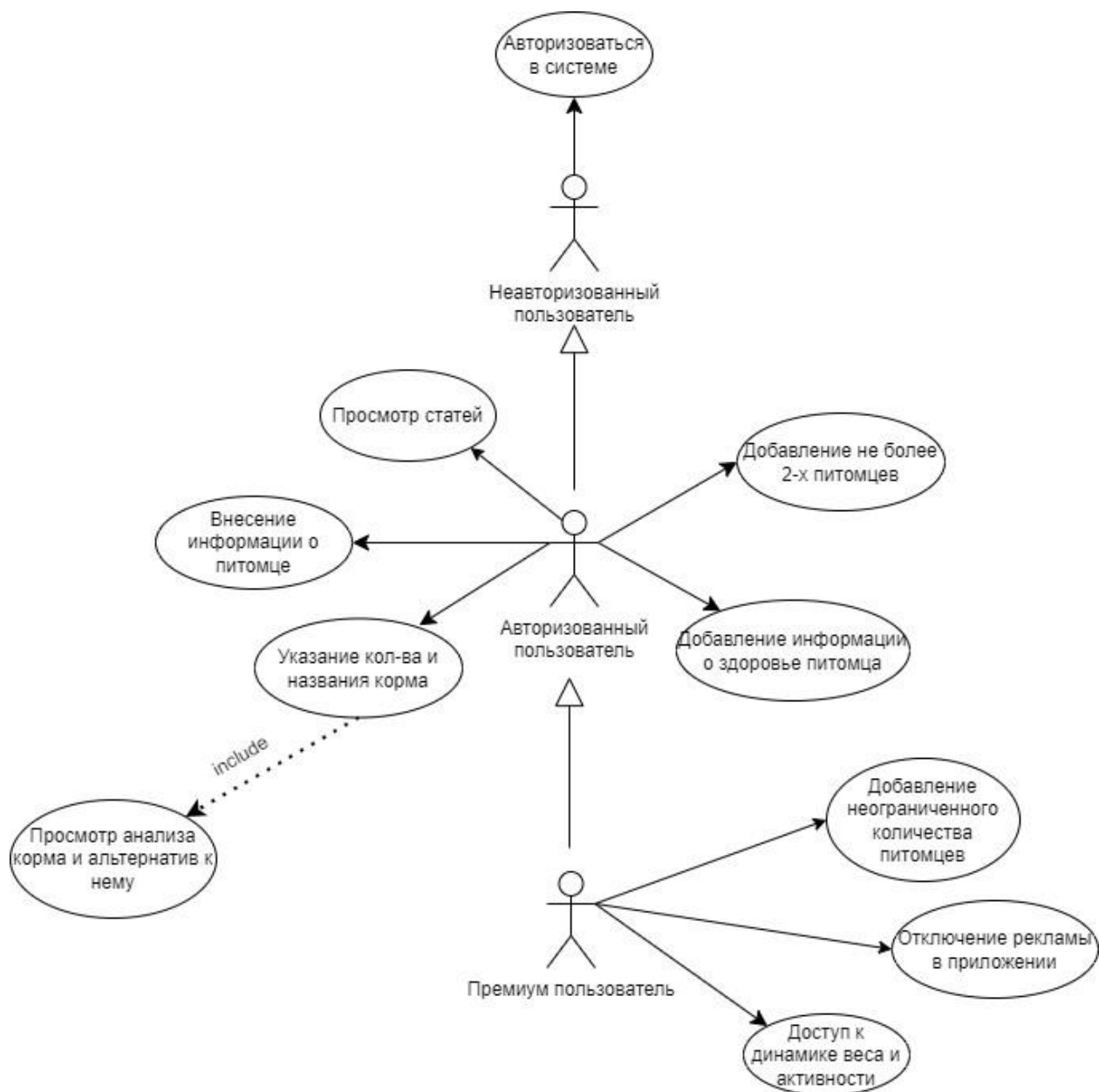


Рисунок 5 - Use-Case диаграмма пользования приложением

2.3.3 Диаграмма последовательности

Диаграмма последовательности является важным инструментом для проекта, который помогает более глубоко понимать процесс, улучшать его эффективность и упрощать взаимодействие.

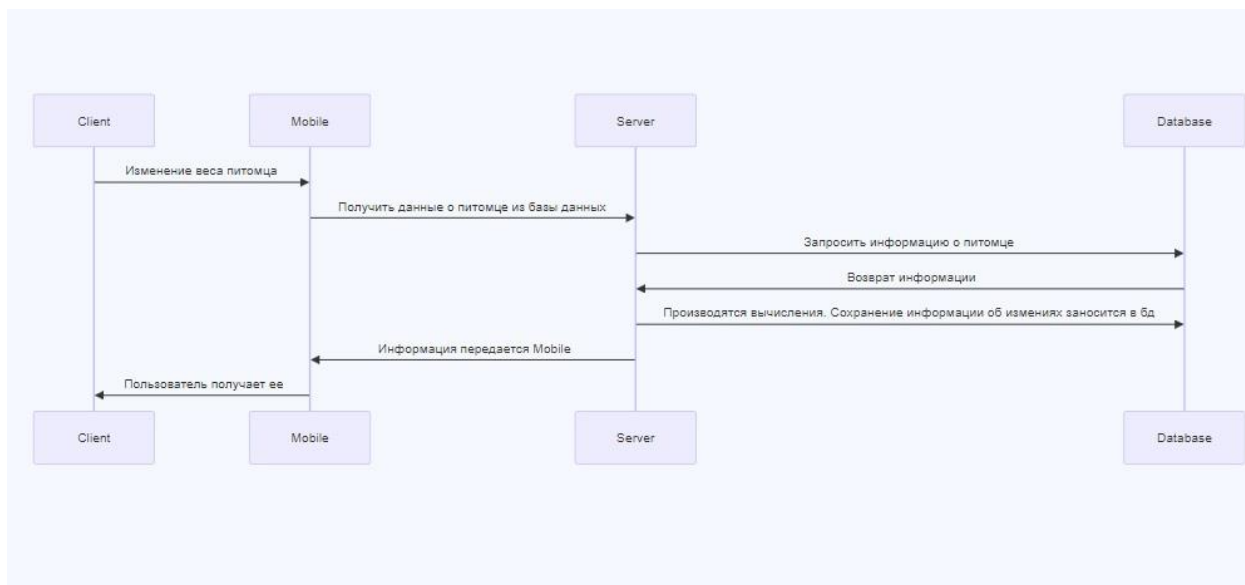


Рисунок 6 - Диаграмма последовательности

2.3.4 Диаграмма развертывания

Диаграмма развертывания позволяет определить требования к аппаратному обеспечению, планировать установку и настройку компонентов системы, а также оценивать ее производительность и масштабируемость. Данная диаграмма представлена на рисунке 7.

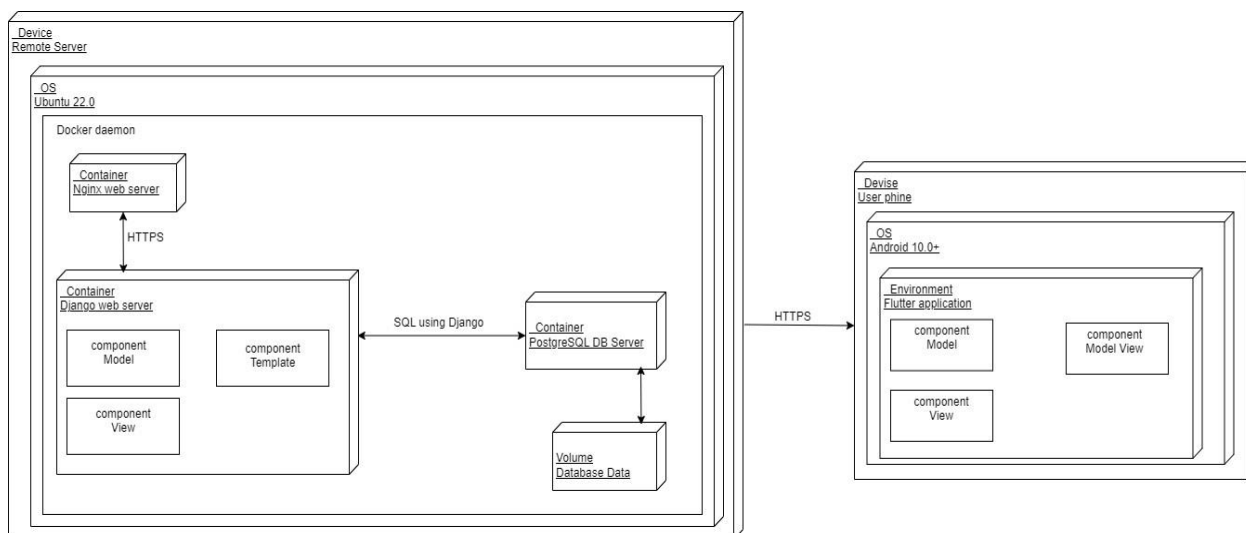


Рисунок 7 - Диаграмма развертывания приложения

2.3.5 Диаграмма состояния

Диаграмма состояния позволяет определить возможные сценарии поведения системы, выделить ключевые состояния и переходы между ними, а также оценить ее надежность и устойчивость к ошибкам. Для нашего проекта были спроектированы 3 диаграммы для состояний премиум пользователь, авторизованного пользователя и неавторизованного пользователя. Данные диаграммы представлены на рисунках 8-10.

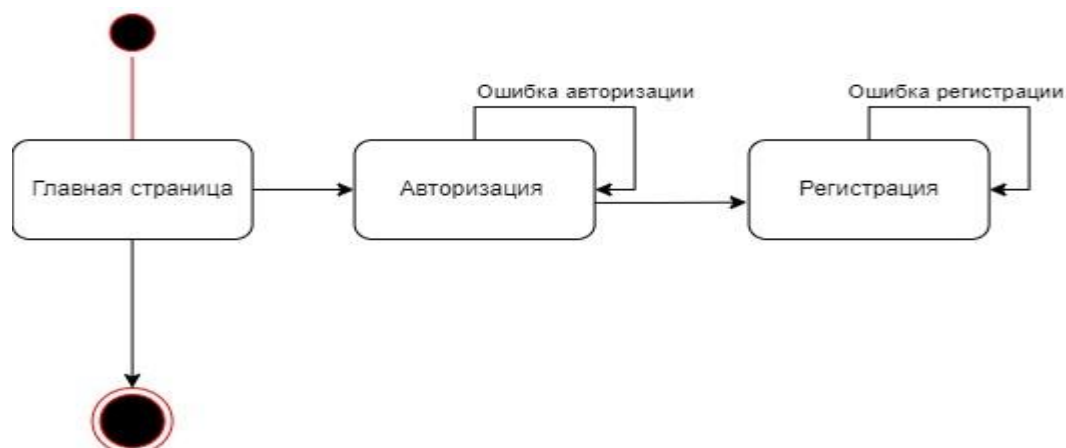


Рисунок 8 - Диаграмма состояния неавторизованного пользователя

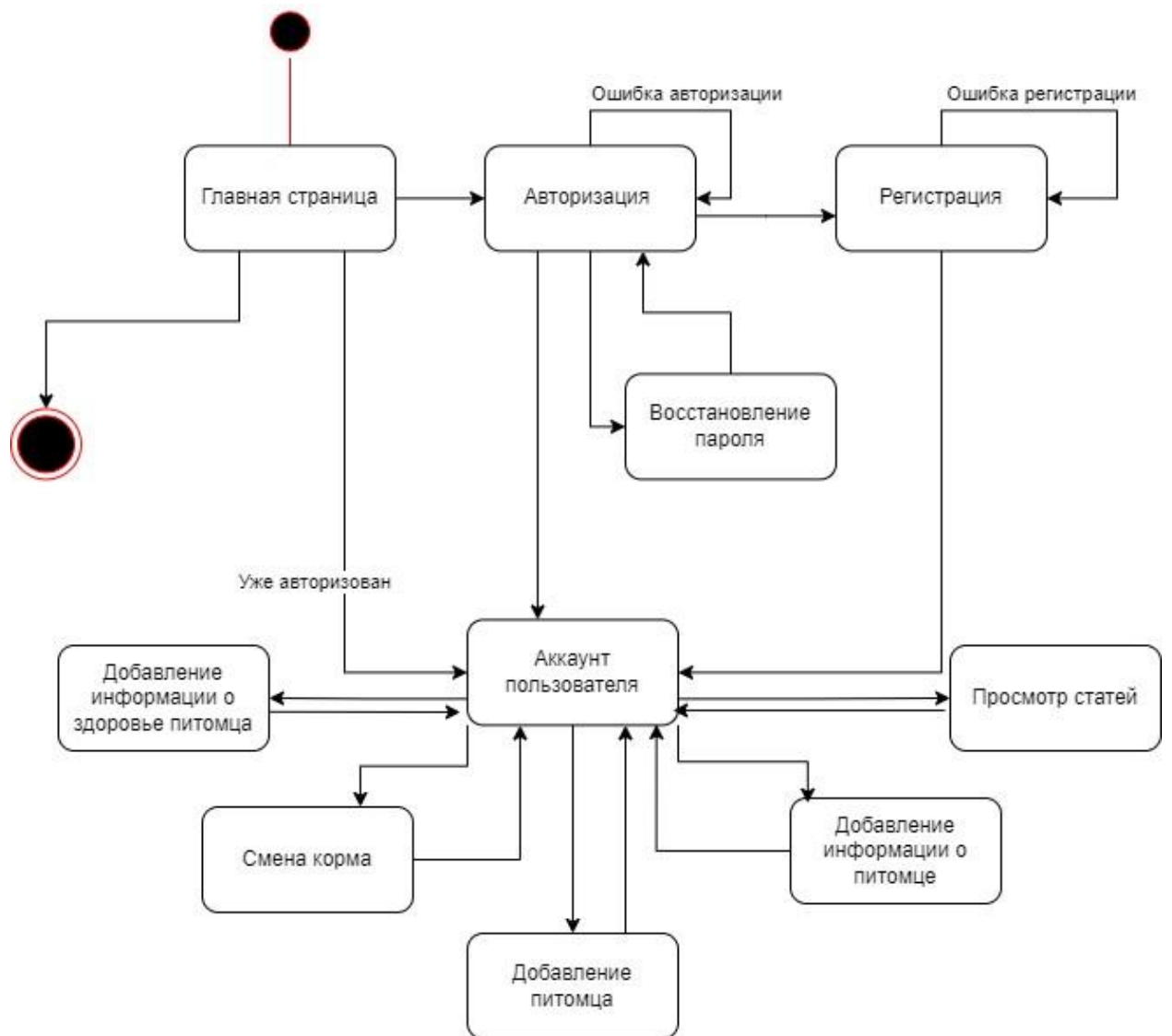


Рисунок 9 - Диаграмма состояния авторизованного пользователя

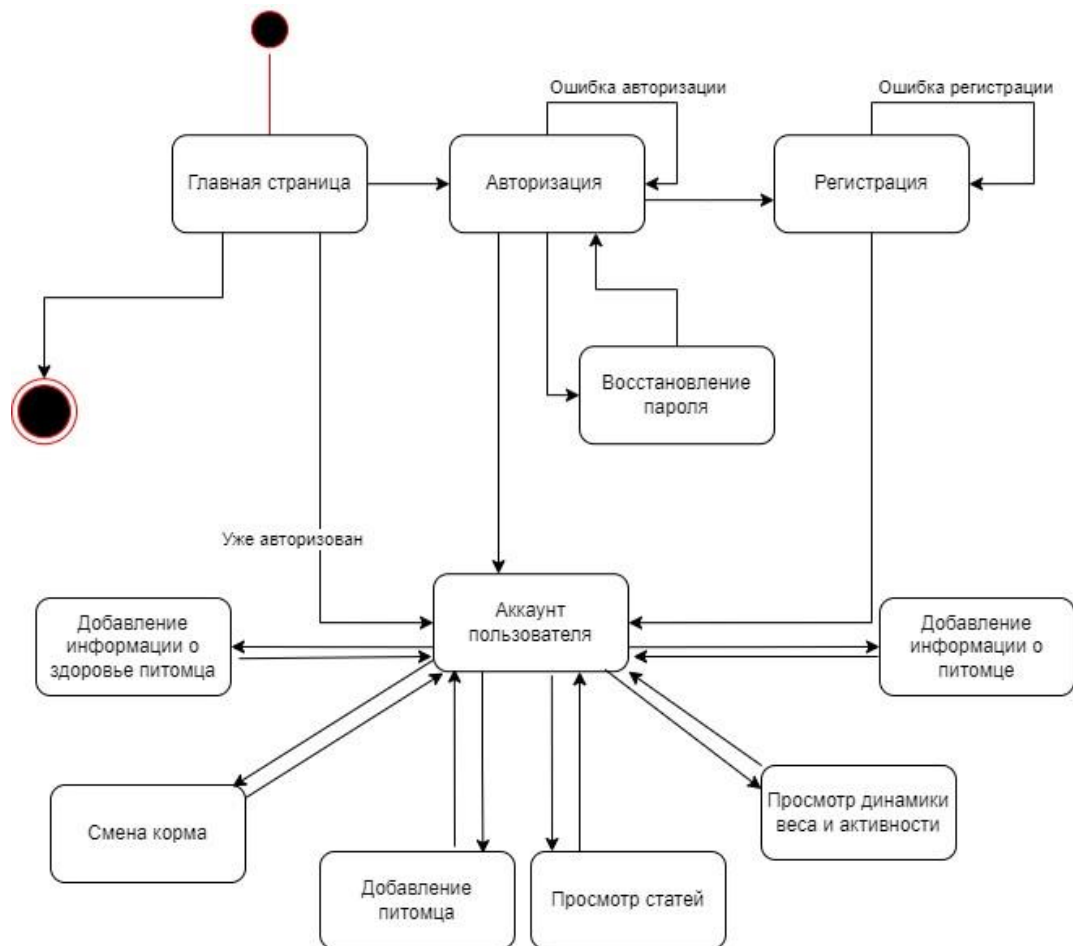


Рисунок 10 - Диаграмма состояния премиум пользователя

2.3.6 Диаграмма объектов

Диаграмма объектов позволяет определить классы объектов, их атрибуты и методы, а также взаимодействие между ними. Она помогает разработчикам лучше понимать структуру системы и проектировать ее более эффективно. Данная диаграмма представлена на рисунке 11.

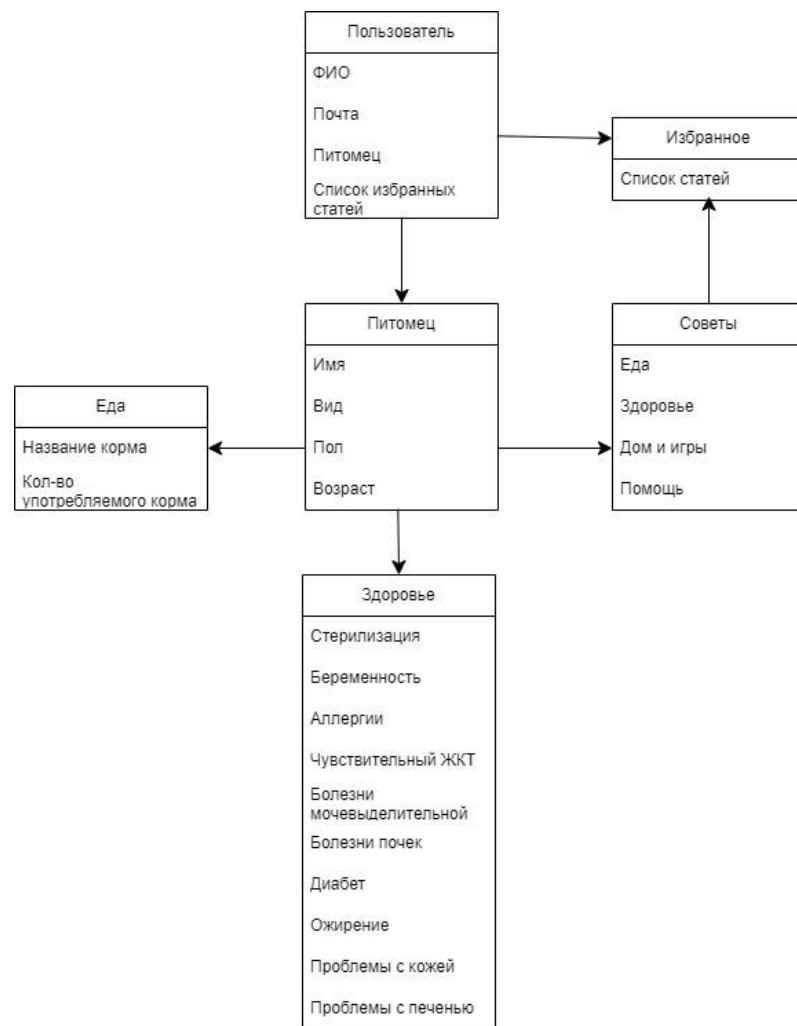


Рисунок 11 - Диаграмма объектов

2.3.7 Диаграмма активности

Диаграмма активности помогает разработчикам лучше понимать процессы в системе, выявлять узкие места и оптимизировать их. Она также может использоваться для описания бизнес-процессов и управления проектами. Для данного проекта были спроектированы 3 диаграммы активности для премиум пользователя, авторизованного пользователя и неавторизованного пользователя. Данные диаграммы представлены на рисунках 12-14.



Рисунок 12 - Диаграмма активности неавторизованного пользователя

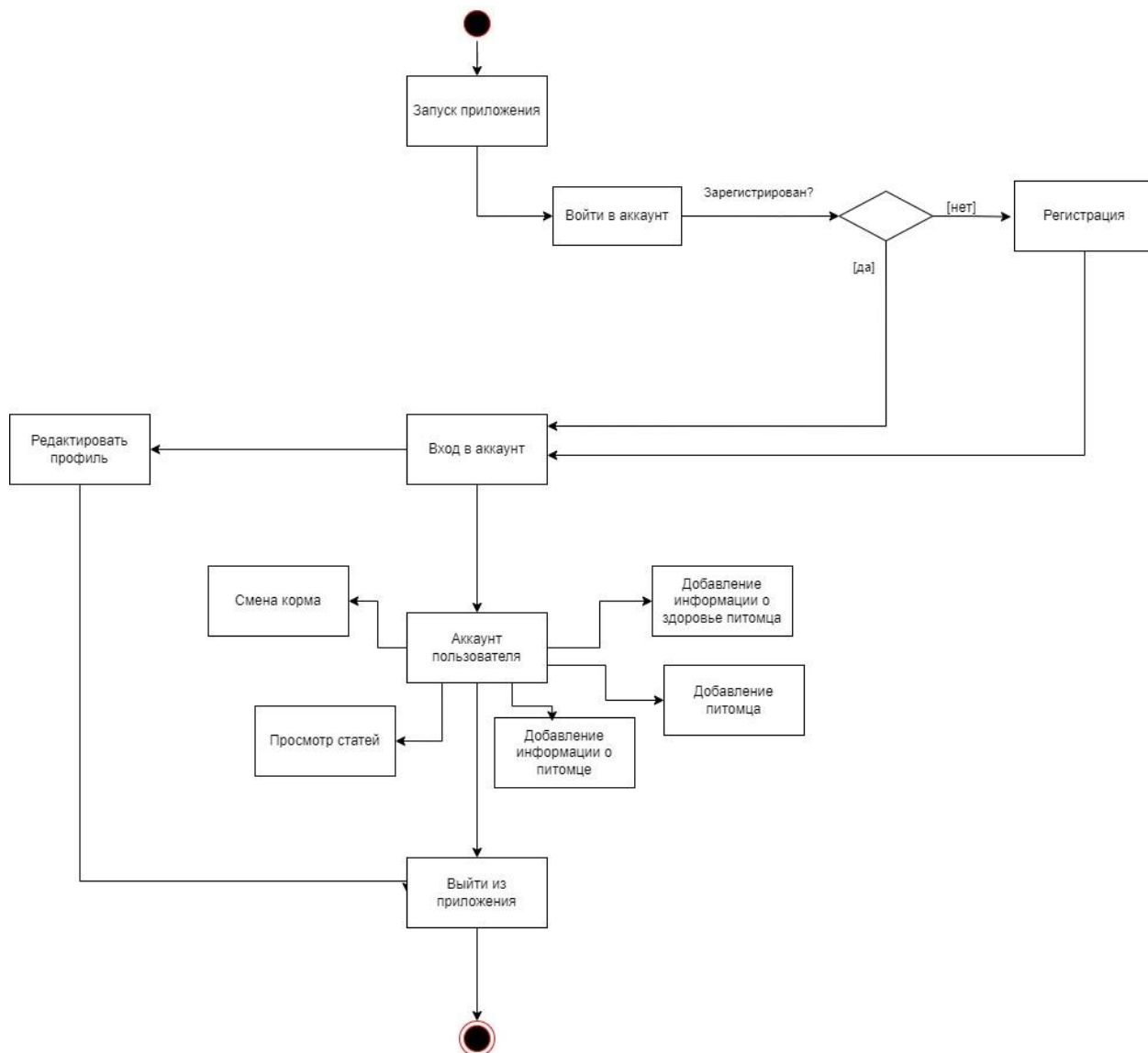


Рисунок 13 - Диаграмма активности авторизованного пользователя

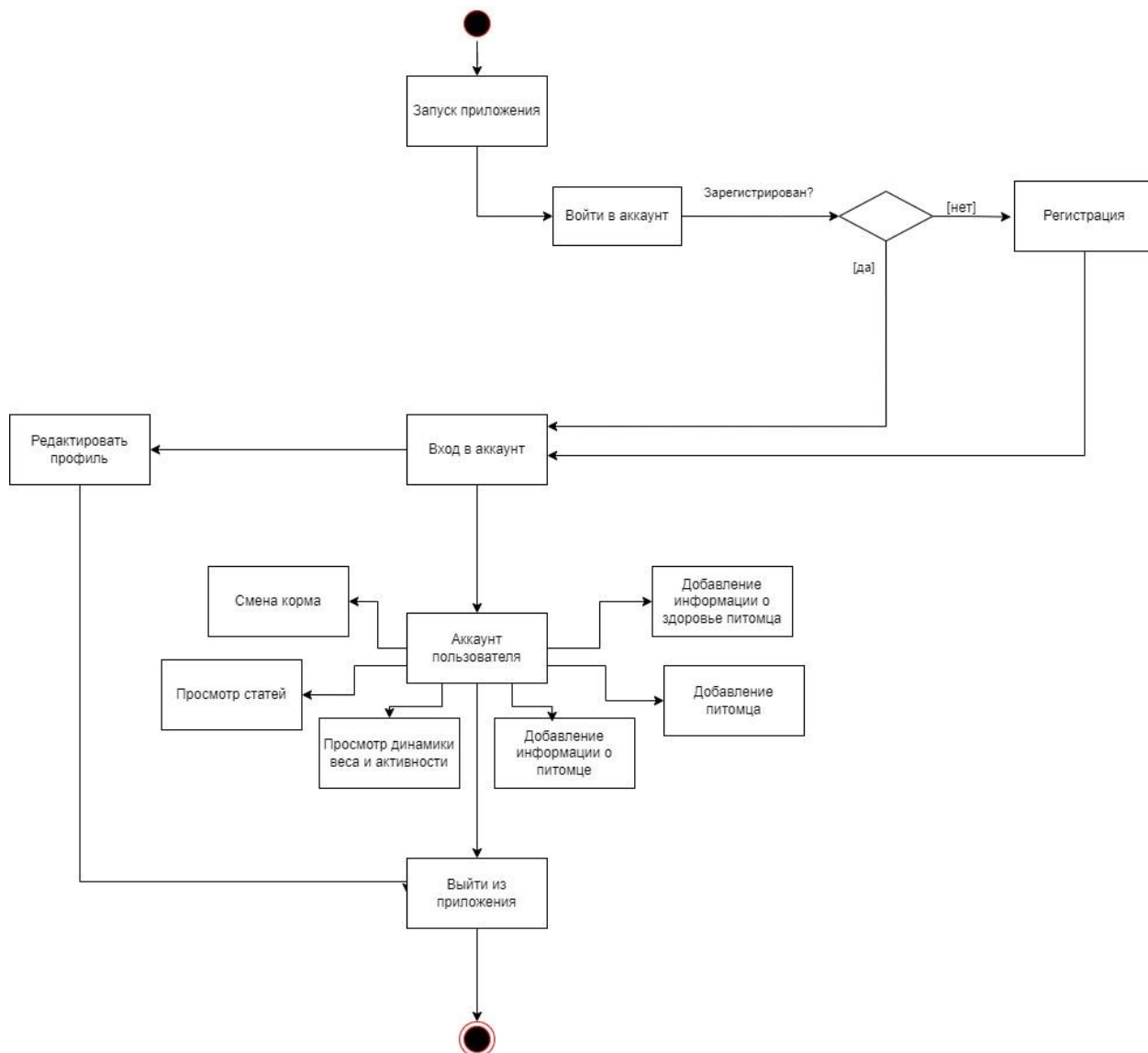


Рисунок 14 - Диаграмма активности премиум пользователя

3 Реализация

3.1 Средства реализации

Ниже приведен перечень используемых технологий.

Backend

- Python 12-ой версии;
- Django Rest framework;
- PostgreSQL;
- API Swagger;
- uWSGI.

Frontend:

- Dart;
- Flutter.

Для развертывания приложения:

- клиент Certbot для создания и получения SSL сертификата;
- Docker;
- Nginx – прокси-сервер с поддержкой SSL.

Инструменты для ведения документации:

- Miro;
- YouTrack;
- Figma.

Дополнительный инструментарий:

– GitHub.

3.2 Реализация базы данных

Для хранения данных была выбрана база данных PostgreSQL. Она является продуктом с открытым исходным кодом, который поддерживается многими серверами. PostgreSQL поддерживает множественные типы данных, такие как числа разной точности, тексты с различными кодировками, изображения, звуки, видео, XML-документы, JSON-объекты и многие другие.

3.2.1 ER-диаграмма базы данных

ER-диаграмма используется для описания структуры базы данных, проектирования новых баз данных и анализа существующих баз данных. Она позволяет разработчикам и аналитикам лучше понять логические связи и зависимости между различными сущностями и их атрибутами. Данная диаграмма представлена на рисунке 15.

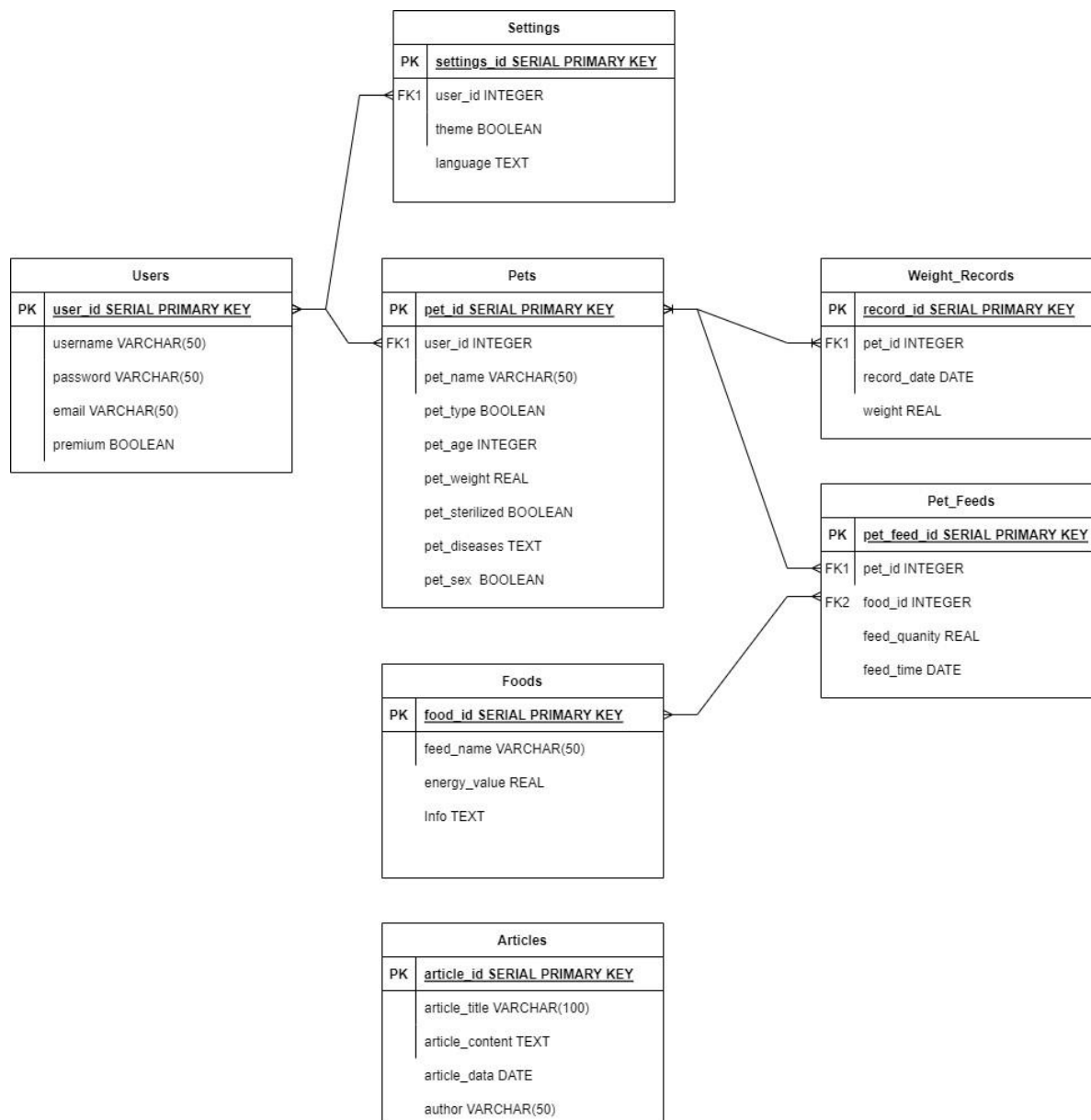


Рисунок 15 - ER-диаграмма базы данных

3.3 Реализация клиентской части

Для реализации клиентской части приложения было выбрано сочетание Flutter и языка Dart. Этот набор обеспечивает разработчикам множество возможностей для создания мобильных приложений под Android. Flutter предоставляет набор готовых виджетов для разработки пользовательского интерфейса, а также инструменты для доступа к аппаратным возможностям устройства и взаимодействия с различными сервисами и API. Dart, в свою очередь, предоставляет надежный и мощный язык программирования,

который позволяет создавать сложные и высокопроизводительные приложения.

Клиентская часть имеет архитектуру, основанную на модели MVVM. Контроллеры обрабатывают входящие запросы и отправляют ответы клиенту. Они являются интерфейсом между клиентом и бизнес-логикой приложения. Сервисы предоставляют бизнес-логику приложения. Они представлены в виде классов активностей и фрагментов приложения. Модели представлены отдельными классами для пользователей.

Все эти компоненты взаимодействуют друг с другом, чтобы обеспечить функциональность приложения. Контроллеры получают запросы от клиента, направляют их в сервисы, которые используют репозитории для доступа к данным в базе данных, а затем возвращают ответы в контроллеры, которые отправляют их клиенту.

3.3.1 Форма экрана «Вход»

Этот экран предназначен для ввода электронной почты и пароля пользователя для процедуры авторизации. После успешной авторизации происходит переход на экран «Профиль» для авторизованных пользователей. Также на экране присутствует кнопка «Забыли пароль» для возможности восстановления пароля. Пользователь также может выбрать кнопку «Регистрация» для перехода на экран регистрации.

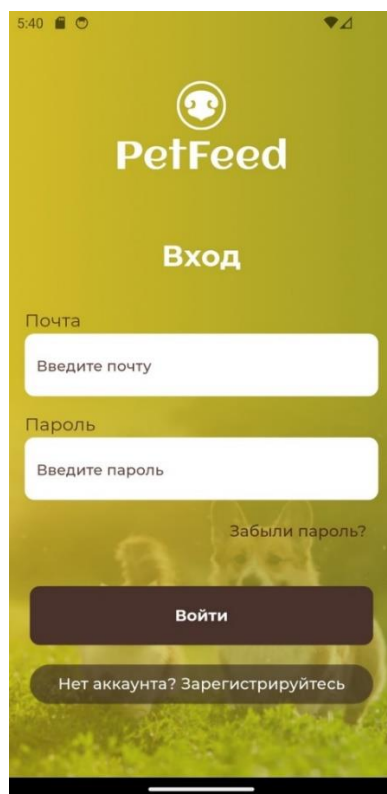


Рисунок 16 - Экран «Вход»

3.3.2 Форма экрана «Регистрация»

На данном экране предлагается ввести адрес электронной почты, придумать пароль и нажать кнопку «Регистрация». Затем необходимо последовательно указать следующие данные:

- ваше имя;
- вид питомца;
- кличку питомца;
- пол питомца;
- дату рождения;
- уровень активности;
- вес;
- особенности здоровья;

– основной корм.

В процессе регистрации можно воспользоваться кнопками «<» и «Далее>» для навигации между разделами регистрации. После заполнения всех форм регистрации необходимо нажать кнопку «Завершение».

3.3.3 Форма экрана «Профиль»

Данный экран появляется при входе в приложение авторизованного пользователя или после успешной регистрации. Здесь содержится информация о питомце, его весе, активности и наличии заболеваний. В верхней части экрана находятся кнопки «Еда» и «Советы», а в нижней части расположены кнопки «Питомцы», «Главное» и «Настройки».



Рисунок 17 - Экран «Профиль»

3.3.4 Форма экрана «Настройки»

На данном экране представлены кнопки для изменения имени, адреса электронной почты и пароля. Пользователь может выбрать светлую или темную тему оформления, а также изменить язык интерфейса. Предусмотрены опции выхода из аккаунта и удаление аккаунта.

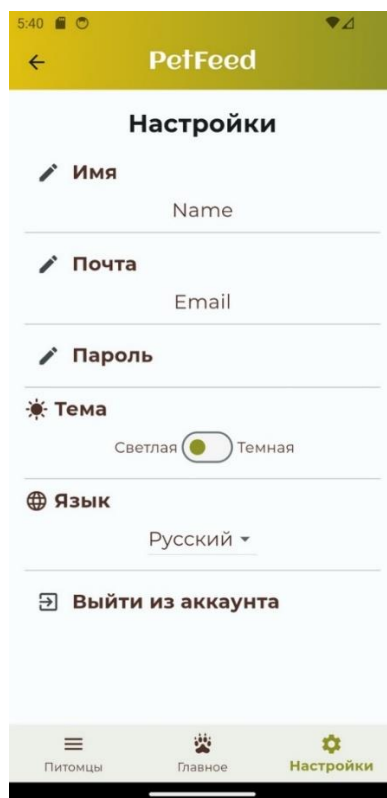


Рисунок 18 - Экран «Настройки»

3.3.5 Форма экрана «Советы»

На данном экране расположены кнопки для разделов «Еда», «Здоровье», «Дом и игры» и «Помощь». В каждом разделе пользователь может найти статьи, предоставленные разработчиками из научных журналов и посвященные соответствующей тематике. Кроме того, статьи также могут быть получены из академических исследований, университетских изданий и других официальных источников. Пользователи могут добавлять статьи в избранное. Также имеется кнопка «Все статьи», позволяющая просматривать содержимое всех разделов одновременно.

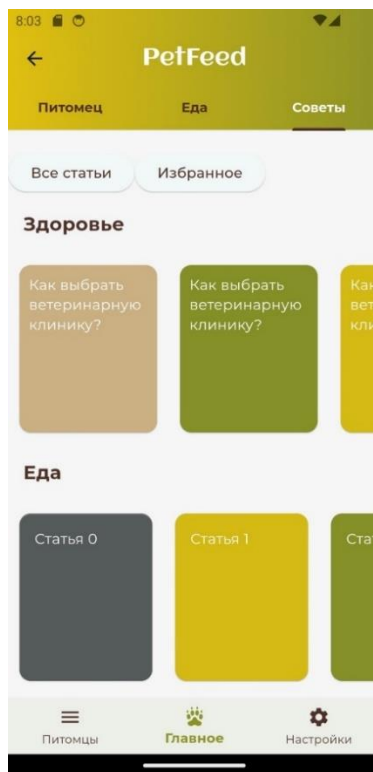


Рисунок 19 - Экран «Советы»

3.3.6 Форма экрана «Еда»

На данном экране предлагается ввести следующие данные о кормлении питомца:

- количество употребляемого корма;
- название корма.

Также на странице представлены:

- Подходящие альтернативы, которые могут быть предложены в случае, если текущий корм не подходит питомцу;
- Анализ состава корма, включающий количество белков, жиров, витаминов и других питательных элементов, а также минимальные рекомендуемые значения для каждого из них.



Рисунок 20 - Экран «Еда»

The screenshot shows the 'Food' screen of the PetFeed app with a detailed 'Анализ корма' (Feed Analysis) table. The table has three columns: 'Элемент' (Element), 'Есть' (Is there), and 'Минимум' (Minimum). The data is as follows:

Элемент	Есть	Минимум
Белки	0.0%	26%
Жиры	0.0%	9%
Кальций	0.0%	0.6%
Фосфор	0.0%	0.5%
Витамин А	0 МЕ	3332 МЕ
Витамин D	0 МЕ	280 МЕ
Витамин E	0 МЕ	40 МЕ
Витамин B1	0.0 мг	5.6 мг
Витамин B2	0.0 мг	4 мг
Витамин B3	0.0 мг	60 мг
Витамин B4	0.0 мг	2400 мг
Витамин B5	0.0 мг	5.75 мг
Витамин B6	0.0 мг	4 мг
Витамин B9	0.0 мг	0.8 мг
Витамин B12	0.0 мг	0.02 мг
Витамин H	0.0 мг	0.07 мг

The bottom navigation bar is the same as in the previous screenshot, with 'Питомцы', 'Главное' (selected), and 'Настройки'.

Рисунок 21 - Экран «Еда»

3.4 Серверная часть

В качестве языка был выбран строго типизированный объектно-ориентированный язык программирования Python 12-ой версии. Этот язык отличается строгой типизацией и объектно-ориентированным подходом, что

делает его идеальным выбором для создания мобильных приложений. Python имеет широкий спектр инструментов для разработки и большое сообщество разработчиков, которые создали множество фреймворков и библиотек, которые можно легко интегрировать в проект. Кроме того, Python известен своей простотой и читаемостью кода, что облегчает разработку и сопровождение приложений.

Также применяется программная платформа Docker для тестирования и развертывания приложений.

3.4.1 Общее описание проекта

Структура проекта включает в себя конфигурационные файлы и директории, содержащие модели, представления, сериализаторы, URL-адреса и другие компоненты, относящиеся к различным функциональным областям, таким как аутентификация, авторизация, обработка запросов, база данных, исключения и т.д.

Основные конфигурационные файлы включают в себя:

- «settings.py» - содержит настройки проекта, такие как база данных, средства аутентификации, middleware и так далее;
- «urls.py» - содержит маршрутизацию проекта, которая определяет какое представление будет обрабатывать запрос для каждого URL-адреса;
- «wsgi.py» - точка входа для WSGI-совместимого веб-сервера;
- migrations - отвечают за управление изменениями в базе данных, автоматическое создание и обновление таблиц;
- Настройки (settings.py, .env) - отвечают за конфигурацию приложения, такие как подключение к базе данных, настройки безопасности, параметры развертывания и так далее;

Сервер был развернут на хостинге Яндекс.Клауда. База данных находится на сервере. После создания базы данных были предоставлены данные для подключения, такие как хост, порт, имя базы данных, имя пользователя и пароль. Эта информация используется для настройки

подключения к базе данных из приложения.

```
petfeed/  
manage.py  
petfeed_app/  
__init__.py  
settings.py  
urls.py  
asgi.py  
wsgi.py  
pets/  
__init__.py  
models.py  
views.py  
serializers.py  
urls.py  
permissions.py  
users/  
__init__.py  
models.py  
views.py  
serializers.py  
urls.py  
permissions.py  
articles/  
__init__.py  
models.py  
views.py  
serializers.py  
urls.py  
permissions.py  
config/  
__init__.py  
settings.py  
urls.py  
static/  
css/  
img/  
js/  
templates/  
.env  
.gitignore  
.dockerignore  
Dockerfile  
nginx.conf
```

Рисунок 22 - Общая структура серверной части приложения

3.4.2 Папка Pets

Django, Views отвечают за обработку входящих HTTP-запросов и управление потоком выполнения приложения. Данные классы образуют так называемый первый микросервисный слой. В папке Pets, Views определяются различные методы для обработки запросов, связанных с питомцами. Эти методы включают в себя создание, чтение, обновление и удаление записей о питомцах в базе данных, а также выполнение других

действий, связанных с питомцами.

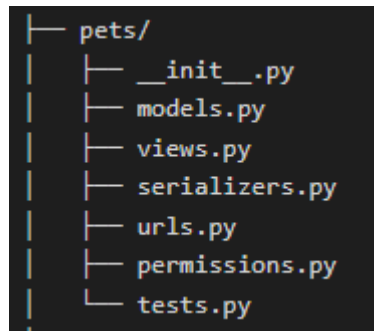


Рисунок 23 - Папка Pets

3.4.3 Папка Users

В папке Users, Views отвечают за обработку запросов, связанных с пользователями приложения. Эти запросы включают в себя регистрацию новых пользователей, аутентификацию существующих пользователей, обновление информации о пользователях и выполнение других действий, связанных с пользователями. Кроме того, в папке Users находятся классы, отвечающие за аутентификацию и авторизацию пользователей, а также за управление их доступом к различным частям приложения.

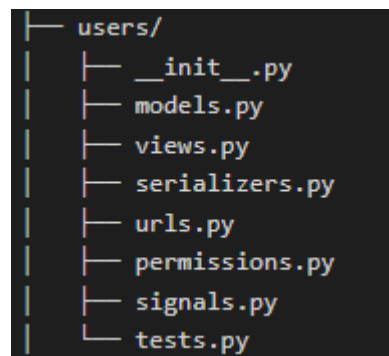


Рисунок 24 - Папка Users

3.4.4 Папка Articles

В папке Articles, Views отвечают за обработку запросов, связанных со статьями, которые могут быть опубликованы в приложении. Эти запросы включают в себя создание новых статей, чтение существующих статей, обновление информации о статьях и выполнение других действий, связанных со статьями. Кроме того, в папке Articles находятся классы, отвечающие за хранение и извлечение статей из базы данных, а также за их форматирование и отображение в приложении.

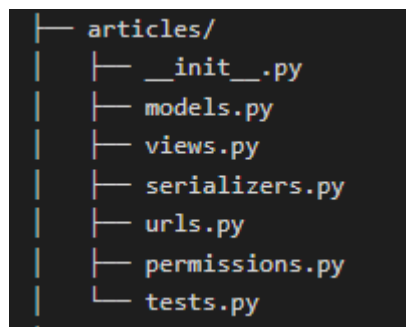


Рисунок 25 - Папка Articles

3.4.5 Папка Config

Папка Config содержит общую конфигурацию для всего приложения. В ней находятся классы, отвечающие за настройку базы данных, маршрутизацию URL-адресов, управление средствами массовой информации и выполнение других общих задач, связанных с конфигурацией приложения.

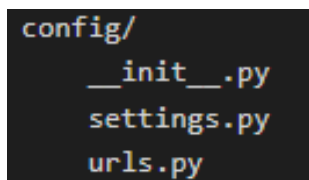


Рисунок 26 - Папка Config

3.4.6 Папка Static

Папка Static содержит статические файлы, которые используются в приложении.

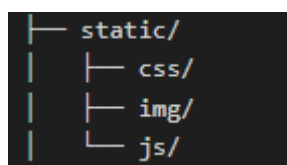


Рисунок 27 - Папка Static

4 Аналитика

Для сбора метрик для приложения было выбрано AppMetrica от Яндекс. Это система для сбора данных об использовании приложения пользователями. Она отличается высокой скоростью и удобством настройки метрик для мобильных приложений. Кроме того, она предлагает интуитивный интерфейс и понятное руководство по использованию.

Были составлены четыре воронки конверсии:

- общая статистика;
- аккаунт;
- еда;
- советы.

Воронка «Общая статистика» нужна для просмотра информации о количестве скачиваний приложения и количества запуска приложения. Изображение данной воронки на рисунке 28.

Конверсия шагов

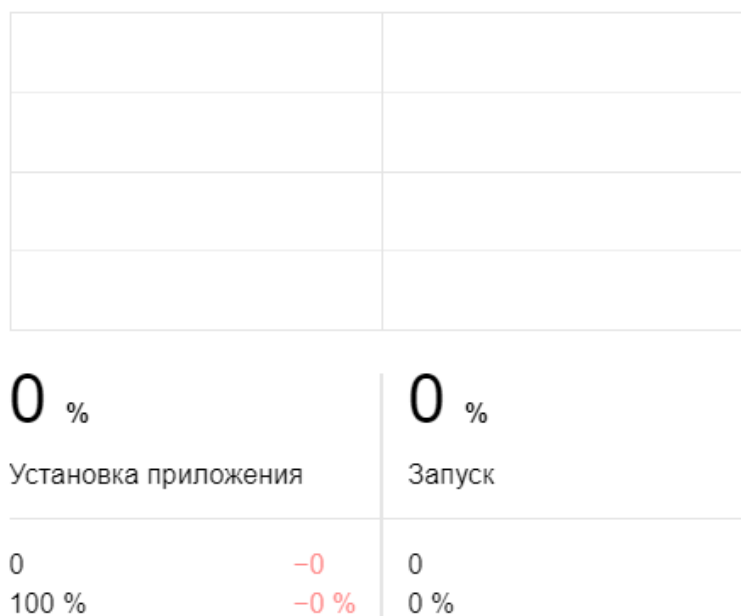


Рисунок 28 - Воронка «Общая статистика»

Воронка «Аккаунт» нужна для просмотра статистики по количеству входа в аккаунт и выхода из него, а также регистрации и изменении аккаунта. Изображение данной воронки на рисунке 29.

Конверсия шагов

0 %	0 %	0 %	0 %
Вход в аккаунт	Выход из аккаунта	Регистрация	Изменение

0	-0	0	-0	0	-0	0
100 %	-0 %	0 %	-0 %	0 %	-0 %	0 %

Рисунок 29 - Воронка «Аккаунт»

Воронка «Еда» нужна для отслеживания и анализа статистики, связанной с процессом добавления корма. Изображение данной воронки на рисунке 30.

Конверсия шагов

0 %	0 %
Ввод корма	Просмотр статистики

0	-0	0
100 %	-0 %	0 %

Рисунок 30 - Воронка «Еда»

Воронка «Советы» нужна для просмотра статистики по количеству просмотренных статей, а также для отслеживания процесса добавления их в избранное. Изображение данной воронки на рисунке 31.

Конверсия шагов

0 %

Просмотр статей

0

100 %

-0

-0 %

0 %

Добавление в избранное

0

0 %

Рисунок 31 - Воронка «Советы»

Заключение

В ходе выполнения данного курсового проекта был выполнен анализ предметной области и аналогов разрабатываемого приложения.

Для разработки приложения были разработаны макеты интерфейса, выбрана платформа приложения, построены UML диаграммы.

Для контроля версий был создан репозиторий GitHub.

При разработке приложения были реализованы следующие задачи:

- учет питания;
- анализ корма;
- мониторинг здоровья;
- получение полезной информации по теме домашних животных.

Список использованных источников

1. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)/ K. Marsicano, B. Gardner, B. Phillips, C Stewart. – New York: Big Nerd Ranch Guides, 2019. – 1036 с.
2. Android Application Development Cookbook: 93 Recipes for Building Winning Apps / L. Wei-Meng. – New York: Wrox, 2013. – 408 с.
3. Официальная документация Yandex Metrica [Электронный ресурс]. – Режим доступа: <https://metrica.yandex.com/about?> – Заглавие с экрана. – (Дата обращения: 21.04.2024)