Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Формирование требований к информационной системе «Трекер ухода за питомцами PawesomeCare»

Студент: Козека Е. М.

ФИТ 4 курс 4 группа

Преподаватель: Парамонов А. И.

Минск 2025

# **Введение**

В современном мире все больше людей стремятся окружить себя заботой и теплом, даже в условиях городской среды. Одним из способов обрести верного друга и снизить уровень стресса является содержание домашних питомцев. Многие люди сталкиваются с трудностями в уходе за животными из-за нехватки знаний о правильном кормлении, поддержании здоровья, воспитании и особенностях каждой породы. В связи с этим растет потребность в создании удобных и доступных инструментов для систематизации и упрощения процесса заботы о наших маленьких друзьях.

Актуальность данной темы обусловлена тем, что технологии, направленные на улучшение и облегчение повседневных задач, играют важную роль в жизни современного человека. Автоматизированные системы и web-приложения, предоставляющие своевременные уведомления и полезные рекомендации, могут значительно упростить уход за домашними любимцами, особенно для людей с насыщенным графиком.

Существующие приложения и сервисы для ухода за домашними животными предлагают разнообразные функции, такие как напоминания о кормлении, выгуле, вакцинациях или советы по выбору питомцев. Однако, большинство из них либо узко специализированы, либо не обладают функцией взаимодействия между пользователями, что ограничивает их применение. Таким образом, возникает необходимость в разработке web-приложения, который не только предоставлял бы информацию о животных, но и создавал пространство для общения и обмена опытом среди пользователей.

Основные задачи приложения:

* разработать удобный интерфейс, который позволит пользователям получать полную информацию о животных;
* предоставить функционал для управления своим списком питомцев и получения пользователем напоминаний об уходе за ними;
* создать систему взаимодействия пользователей, где они смогут делиться советами по уходу за различными животными.

**1 Назначение разработки и целевая аудитория**

Разработка приложения, посвященного уходу за домашними животными, имеет следующие **ключевые цели**:

* функциональность: страницы должны загружаться быстро, навигация должна быть удобной, а интерфейс - простым и понятным;
* информативность: на сайте должна быть полная и понятная информация об услугах. Клиент должен найти все ответы на свои вопросы на сайте;
* ориентированность на клиента: сайт должен содержать актуальную информацию о различных животных, поиск информации должен быть быстрым;
* визуальная привлекательность: внешний вид сайта играет важную роль, так как пользователь сначала оценивает его визуально.

**Целью разработки** трекера ухода за питомцами является предоставление удобного и полезного инструмента для всех пользователей, включая как опытных заводчиков, так и тех, кто только что принес в дом своего первого питомца. Приложение будет стремиться удовлетворить потребности каждой группы пользователей, предлагая простой доступ к базе знаний по ветеринарии и содержанию животных, систему напоминаний для своевременного кормления, ветеринарных осмотров и гигиенических процедур, а также персонализированные рекомендации и советы.

**Система прав доступа** построена по принципу наращивания привилегий: каждая следующая роль включает в себя все права предыдущей с дополнительными уникальными функциями.

**Гость**. Это пользователь, который не авторизован в приложении. Его возможности строго ограничены для мотивации пройти регистрацию. Он может либо войти в уже существующий аккаунт, либо создать новый.

**Обычный пользователь**. Это основная и самая массовая роль. Это авторизованный пользователь, который активно использует приложение в качестве помощника по уходу за питомцами. Пользователь имеет свой персональный профиль, который является обязательным условием для работы с приложением. Он может просматривать общий каталог животных и информацию о них, имеет доступ к поиску и фильтрации животных в каталоге, может читать публикации с советами по уходу от других пользователей. Также он может добавлять животных из общего каталога в свой профиль и создавать для них персональные карточки, получать напоминания о кормлении, выгуле, прививках и приеме витаминов и предлагать новые виды питомцев для добавления в общий каталог.

**Администратор**. Это пользователь с неограниченными правами, который отвечает за работоспособность приложения, его контентную и пользовательскую базу. Он осуществляет непосредственное добавление, редактирование и удаление списков видов животных, пород, типов напоминаний и категорий статей без необходимости заявок, просмотр всех статей в системе и их удаление, если они нарушают правила сообщества. Также он может принимать или отклонять заявки пользователей на добавление новой породы животного.

**2 Функциональные требования**

Разрабатываемое программное средство «PawesomeCare» должно представлять собой web-приложение, разделенное на серверную и клиентскую составляющие. Взаимодействие с клиентской частью приложения осуществляется путем использования браузера, установленного на персональный компьютер. В системе должен быть обеспечен надлежащий уровень защиты информации. Разрабатываемое приложение должно иметь удобный и интуитивно понятный интерфейс. Надежное и устойчивое функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечисленных ниже:

* защита персональных данных пользователей паролем, которые хранятся в захешированном виде;
* выполнением требований «ГОСТ 31078-2002. Защита информации. Испытания программных средств на наличие компьютерных вирусов»;
* высокая квалификация персонала, работающего с ИС.

Необходимо обеспечить максимально безотказную работу программы при любых входных данных, а также при любых действиях пользователя.

В «PawesomeCare» предусмотрены **3 основные роли пользователей**: администратор, обычный пользователь и гость.

Для **гостя** следует реализовать безопасную регистрацию и авторизацию в приложении, а также возможность восстанавливать пароль.

Для **зарегистрированного пользователя** веб-приложения для ухода за питомцами сформулированы следующие функциональные требования:

* создать возможности управления своим профилем;
* предоставить пользователям возможность управления карточками своих питомцев;
* реализовать поиск и фильтрацию животных по различным критериям;
* обеспечить возможность отправки заявок на добавление новых животных в общий каталог;
* дать возможность пользователям читать и публиковать советы по уходу за различными видами животных;
* разработать систему напоминаний о кормлении и необходимых процедурах.

Функциональные требования для **администратора** включают в себя все функциональные требования для обычного пользователя, а также:

* разработать интерфейс для управления каталогом животных, пород животных, категорий статей и напоминаний (добавление, редактирование, удаление);
* разработать интерфейс для управления статьями пользователей (проверка и удаление);
* реализовать просмотр и обработку заявок пользователей на добавление животных в общий каталог.

**3 Основные аппаратные требования**

Ниже представлены минимальные и рекомендуемые аппаратные требования, которые обеспечат корректную работу всех функций приложения, включая получение персональных рекомендаций по уходу, синхронизацию данных между устройствами и работу с уведомлениями.

Минимальные аппаратные требования к клиентской машине:

* операционная система – iOS 14 и выше / Android 9.0 (Pie) и выше;
* процессор – 64-битный, архитектура ARM, не менее 2 ядер;
* оперативная память – 2 ГБ;
* свободное место для хранения – 200 МБ;
* Интернет – стабильное подключение 3G/LTE/Wi-Fi для синхронизации и загрузки данных.

Рекомендуемые аппаратные требования к клиентской машине:

* операционная система – iOS 16 / Android 12 и выше;
* процессор – 64-битный, 4+ ядра;
* оперативная память – 4 ГБ и более;
* свободное место для хранения – 1 ГБ и более (для кэширования фото, видео и офлайн-данных);
* Интернет – высокоскоростное подключение LTE/5G/Wi-Fi.

Минимальная конфигурация серверной машины:

* процессоры – 4 ядра;
* оперативная память – 8 ГБ;
* дисковое пространство – 100 ГБ;
* сеть – пропускная способность не менее 1 Гбит/с.

Рекомендуемая конфигурация серверной машины:

* веб-сервер / API: CPU – 8 ядер, RAM – 16 ГБ, Storage – 200 ГБ;
* сервер баз данных: CPU – 8 ядер, RAM – 32 ГБ, Storage – 500 ГБ;

Таким образом, сочетание удобного интерфейса, мощного функционала, устойчивой технической базы и грамотного позиционирования позволит создать востребованный продукт, который станет незаменимым помощником для хозяев различных животных.

**4 Основные системные требования**

Проект реализован по клиент-серверной архитектуре с разделением на фронтенд и бэкенд компоненты. Фронтенд представляет собой одностраничное приложение на React, обеспечивающее динамический пользовательский интерфейс. Бэкенд построен на платформе Node.js с использованием фреймворка Express.js, который обрабатывает бизнес-логику и взаимодействие с базой данных.

В качестве основной системы управления базами данных используется PostgreSQL – реляционная СУБД с открытым исходным кодом. Выбор обусловлен надежностью, поддержкой транзакций ACID и богатым функционалом для работы со структурированными данными. База данных спроектирована с нормализованной схемой для минимизации избыточности данных и обеспечения целостности. Также была использована ORM Prisma, которая упрощает взаимодействие с БД, позволяя эффективно управлять моделями данных.

Серверная часть предоставляет RESTful API с JSON-форматом данных. API реализует стандартные HTTP-методы (GET, POST, PUT, DELETE) для операций CRUD. Аутентификация и авторизация реализованы через JWT-токены. Все эндпоинты защищены HTTPS и следуют соглашениям REST. Связь между сервером и СУБД – локальная.

Объекты информационной вербальной модели для приложения могут быть представлены следующим образом:

1) Объект «Пользователь»:

* Описание: владелец домашнего животного, который использует приложение для управления уходом за питомцами, отслеживания их здоровья и обмена опытом с другими владельцами;
* Атрибуты: ID пользователя (никнейм), email, пароль, имя, аватар, описание профиля, роль;
* Связи: связан с Питомцем, Напоминанием, Статьей.

2) Объект «Вид животного»:

* Описание: биологический вид домашнего животного, существующий в справочнике приложения;
* Атрибуты: ID вида, название вида;
* Связи: cвязан с Породой животного, Питомцем.

3) Объект «Порода животного»:

* Описание: порода внутри вида животного с характерными особенностями;
* Атрибуты: ID породы, ID вида животного, название породы, средний размер, средний вес, средняя продолжительность жизни, описание породы, фото, гипоаллергенность, дрессируемость;
* Связи: связан с Животным, Питомцем.

4) Объект «Питомец»:

* Описание: питомец пользователя с индивидуальными характеристиками;
* Атрибуты: ID питомца, ID пользователя, ID вида животного, ID породы, кличка, дата рождения, вес, фото, особенности здоровья;
* Связи: связан с Пользователем, Видом животного, Породой животного.

5) Объект «Тип напоминания»:

* Описание: категория события, о котором нужно напомнить;
* Атрибуты: ID типа напоминания, название;
* Связи: связан с Напоминанием.

6) Объект «Напоминание»:

* Описание: уведомление о предстоящем событии;
* Атрибуты: ID напоминания, ID типа напоминания, ID питомца, дата напоминания, тип события, статус;
* Связи: связан с Типом напоминания, Питомцем.

7) Объект «Статья»:

* Описание: публикация с полезной информацией на платформе;
* Атрибуты: ID статьи, ID автора, ID категории, заголовок, содержание, дата публикации, дата редактирования;
* Связи: Связана с Пользователем, Категорией статьи.

8) Объект «Категория статьи»:

* Описание: категории публикации в сообществе;
* Атрибуты: ID категории, название категории;
* Связи: связан со Статьей.

9) Объект «Запрос на добавление породы животного»:

* Описание: запрос пользователя на добавление новой породы животного в общий каталог;
* Атрибуты: ID запроса, ID пользователя, ID вида животного, название породы, средний размер, средний вес, средняя продолжительность жизни, описание породы, фото, гипоаллергенность, дрессируемость;

Связи: связан с Пользователем, Видом животного.

Приложение полностью контейнеризировано с использованием Docker. Каждый компонент системы (фронтенд, бэкенд, база данных) развертывается в изолированных контейнерах. Это обеспечивает идентичность окружений на всех стадиях разработки и упрощает развертывание.

**Заключение**

Приложение «PawesomeCare» – полностью функциональное приложение для предоставления пользователям возможности поиска информации о домашних животных и об уходе за ними, а также получения напоминаний о предстоящих событиях и процедурах.

За серверную часть отвечает платформа NodeJS, а именно фреймворк ExpressJS, который позволил реализовать быструю асинхронную серверную часть.

Клиентская часть была написана при помощи библиотеки ReactJS для языка JavaScript. Она позволяет строить отзывчивые и удобные интерфейсы с минимальной задержкой.

Для работы с данными используется реляционная база данных PostgreSQL, а ORM Prisma упрощает взаимодействие с ней, позволяя эффективно управлять моделями данных. База данных содержит функционал для работы с ней, также в ней описаны роли, которые ограничивают доступ к данным.

В итоге было разработано веб-приложение, которое не только предоставляет пользователям возможность осуществлять поиск по каталогу животных и получать ключевую информацию о том, как за ними ухаживать, но и предлагает ряд дополнительных функций. Пользователи могут легко ознакомиться с подробной информацией о каждом виде животных и формировать карточки своих питомцев. Они также имеют возможность получать напоминания о приеме лекарств и витаминов, кормлении, вакцинации и выгуле своих маленьких друзей и устанавливать статусы для них. Важным аспектом является возможность пользователями публиковать полезные статьи, редактировать их, а также просматривать статьи, опубликованные другими пользователями.

Тестирование проекта показало, что приложение функционирует правильно и все поставленные тесты прошло. В ходе тестирования ошибок не было найдено.

На этапе развертывания в сети Интернет были выполнены все необходимые настройки серверов и базы данных для обеспечения устойчивости и доступности приложения.

Веб-приложение является законченным программным продуктом, который реализует все перечисленные возможности в полной форме. Веб-приложение поддерживает архитектуру, которая позволяет в дальнейшем быстро и безопасно расширить функционал.