**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc60796559)

[1 ИССЛЕДОВАНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ 8](#_Toc60796560)

[1.1 Анализ области применения мобильного приложения 8](#_Toc60796561)

[1.2 Описание предметной области 8](#_Toc60796562)

[1.3 Обзор существующих мобильных платформ 11](#_Toc60796563)

[1.4 Анализ существующих программных решений 13](#_Toc60796564)

[1.5 Постановка цели и задач дипломного проекта 18](#_Toc60796565)

[2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ 19](#_Toc60796566)

[2.1 Описание функциональности программного продукта 19](#_Toc60796567)

[2.2 Спецификация функциональных требований 20](#_Toc60796568)

[2.3 Разработка функциональной модели 20](#_Toc60796569)

[2.4 Разработка информационной модели 24](#_Toc60796570)

[3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 26](#_Toc60796571)

[3.1 Разработка модели взаимодействия пользователя с интерфейсом 26](#_Toc60796572)

[3.2 Разработка алгоритма поиска ветеринарных клиник 27](#_Toc60796573)

[3.3 Разработка алгоритма работы мобильного приложения 29](#_Toc60796574)

[3.4 Выбор и обоснование языка и среды разработки программного продукта 31](#_Toc60796575)

[3.5 Безопасность операционной системы iOS 34](#_Toc60796576)

[3.6 Проектирование и разработка программного интерфейса 36](#_Toc60796577)

[4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 38](#_Toc60796578)

[4.1 Методы и типы тестирования 38](#_Toc60796579)

[4.2 Тест-кейсы 39](#_Toc60796580)

[4.3 Результаты тестирования 47](#_Toc60796581)

[5 РУКОВОДСТВО ПО УТСАНОВКЕ И ИСПОЛЬЗОВАНИЮ 48](#_Toc60796582)

[5.1 Требования к программным и аппаратным средствам 48](#_Toc60796583)

[5.2 Работа с приложением «MyPet» 48](#_Toc60796584)

[6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «MYPET» 58](#_Toc60796585)

[6.1 Описание функций, назначения и потенциальных пользователей МП 58](#_Toc60796586)

[6.2 Расчет затрат на разработку ПС 58](#_Toc60796587)

[6.3 Оценка эффекта от продажи МП 60](#_Toc60796588)

[ЗАКЛЮЧЕНИЕ 62](#_Toc60796589)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 63](#_Toc60796590)

[ПРИЛОЖЕНИЕ А 64](#_Toc60796591)

**ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ**

В настоящей пояснительной записке применяются следующие определения и сокращения.

*Геолокация* — это географические данные (широта и долгота) о местонахождении устройства.

*IDE – integrated development environment*, интегрированная среда разработки, комплекс программных средств для разработки ПО.

*iOS* – мобильная операционная система для смартфонов, электронных планшетов, носимых проигрывателей и некоторых других устройств, разрабатываемая и выпускаемая американской компанией Apple.

*UI – user interface*, интерфейс пользователя.

*UX – user experience*, опыт пользователя.

*Xcode* – интегрированная среда разработки (IDE) программного обеспечения для платформ macOS, iOS, watchOS и tvOS, разработанная корпорацией Apple.

БД – база данных

МП – мобильное приложение

ПО – программное обеспечение

# ****ВВЕДЕНИЕ****

Сегодня мобильный телефон в жизни человека является незаменимым помощником, он несет пользу тем, кто ценит свое время и не готов тратить его попусту. Представьте, сколько бы мы тратили времени на поиски вдруг понадобившейся нам вещи, если бы в кармане не лежала эта замечательная технология. Взять хотя бы ремонт автомобиля и поиск какой-нибудь редкой запчасти.

Человеку всегда характерно желание достичь наибольшего комфортного существования, а в наше время его атрибутом считается выход в Интернет, причем оставаясь постоянно в онлайн. В связи с этим, стала наиболее актуальной разработка различных приложений для мобильных устройств под IOS с расширением возможностей мобильного Интернета.

Таким образом за последние годы сотовые телефоны стали неотъемлемой частью нашей жизни. И это неудивительно, ведь высокий ритм жизни, реализация повседневных целей и задач диктует нам необходимость быть всегда на связи и иметь под рукой верного помощника.

Сейчас наиболее актуальными становятся приложения с определением геолокации пользователя. Геолокация — это географические данные (широта и долгота) о местонахождении устройства. Приложения с геолокацией — это приложения, функции которых завязаны на данных о местоположении пользователя. Данные о местоположении получаются с помощью GPS либо данных сотовой и Wi-Fi сети.

Целью дипломного проекта является создание мобильного приложения «MyPet» под операционную систему iOS. Для разработки программного продукта используются такие средства, как IDE Xcode 11, язык программирования Swift 5.

Задачи дипломного проекта:

* изучить предметную область;
* проанализировать схожие программные продукты, а также используемые ими технологии и их альтернативы;
* выбрать и описать необходимые технологии для разработки мобильного приложения «MyPet» для операционной системы iOS;
* по разработанному проекту создать мобильное приложение «MyPet» для операционной системы iOS;
* провести тестирования программного средства;
* рассчитать экономический эффект от разработки программного средства

Пояснительная записка имеет четко выраженную структуру, которая поделена на разделы и подразделы, в каждом из которых осуществляется пояснение определенного аспекта поставленной задачи и определенного этапа разработки программного средства. В пояснительной записке дипломного проекта было выделено шесть разделов.

Первый раздел «Исследование поставленной задачи» включает в себя подробное описание анализа предметной области и обзор аналогов программного средства.

Второй раздел «Моделирование предметной области и разработка функциональных требований» содержит в себе описание требований, которые поставлены перед разработчиком для реализации поставленных целей, описание функциональности программного продукта, разработку функциональной модели, а также разработку информационной модели программного продукта.

Третий раздел «Проектирование программного средства» содержит в себе модель взаимодействия пользователя с интерфейсом мобильного приложения, порядок создания алгоритма решения задачи, также раскрываются причины выбора программных средств разработки для проектирования программного продукта.

Четвертый раздел «Тестирование программного средства», в котором описывается назначение и сущность тестирования. Так же описываются основные типы ошибок, степень надежности алгоритма и результаты тестирования проектируемого программного средства.

В пятом разделе пояснительной записки «Руководство по разработке и использованию программного средства» отражены сведения об области применения разработанной программы и руководство по использованию.

Данный дипломный проект так же содержит разделы «Технико-экономическое обоснование разработки мобильного приложения «MyPet» под операционную систему iOS».

Дипломный проект выполнен самостоятельно, проверен в системе «Антиплагиат». Процент оригинальности соответствует норме, установленной кафедрой, и составляет 92%.[1]

# 1 ****ИССЛЕДОВАНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ****

## ****1.1 Анализ области применения мобильного приложения****

В настоящее время стремительно возрастает количество домашних питомцев на душу населения. В жизни каждого человека рано или поздно наступает момент, когда приходит мысль о покупке себе домашнего питомца. Ведь, зачастую именно питомец помогает человеку справляться со сложными жизненными ситуациями, даря свое внимание и ласку. Помимо этого, домашние животные способствуют развитию ответственности и дисциплинированности у своих хозяев. Большим плюсом является заведение домашнего питомца в семье с маленьким ребенком, поскольку животное способствует формированию у ребенка положительных качеств, таких как преданность, забота, сострадание и любовь.

Однако, как любое другое живое существо, домашние животные, независимо от их вида, требуют постоянного внимания и ухода. Помимо правильно подобранного рациона питания и регулярных прогулок, большинство питомцев также нуждаются в прививках и периодическом медицинском осмотре. И, в зависимости от сложности строения организма домашнего питомца, ответственность по уходу за ним возрастает. К примеру, для собак и кошек свойственно появление аллергий, раздражителем которых может выступать рацион питания. В таком случае хозяева обращаются за помощью в ветеринарные клиники, которые проводят ряд анализов и подбирают оптимальный гипоаллергенный рацион. Но не часто такой корм можно встретить у себя возле дома, также возникает вопрос в выборе хорошей ветеринарной клиники. С этой целью люди обращаются за помощью к всевозможным интернет-порталам и форумам, на которых владельцы домашних питомцев делятся опытом обращения в ветеринарную клинику, а также подсказывают местонахождение гипоаллергенного корма. Однако подобный поиск часто отнимает слишком много времени, которое могли потратить, к примеру, уделив больше внимания своему питомцу.

Исходя из этого можно сделать вывод, что пользователям не хватает приложения, которое бы совмещало в себе все самое необходимое: местоположение ветеринарных клиник и зоотоваров. Полезные статьи и совету по уходу за своим питомцем. Также мобильное приложение должно должно отвечать всем требованиям безопасности, без потери данных о пользователе.

## ****1.2 Описание предметной области****

В 2020 году цифровые, мобильные и социальные сети стали неотъемлемой частью жизни людей во всем мире. Количество интернет-пользователей в 2020 году достигло 4,54 млрд человек – в 2019 году эта цифра составляла 4,38 млрд.

Аудитория социальных сетей в 2020 году насчитывает 3,80 млрд человек – по сравнению с 3,484 млрд в прошлом году.

Число пользователей мобильных телефонов составляет 5,19 млрд человек – по сравнению с 5.11 млрд в прошлом году.

Количество активных пользователей социальных сетей стремительно растет – уровень проникновения 49%, что на 4% больше, чем в 2019 году.

В среднем каждый пользователь проводит в сети 6 часов и 43 минуты в день, из них именно в социальных сетях – 2 часа и 24 минуты, что больше чем треть от общего времени в интернете.

Для сравнения, за просмотром ТВ люди проводят 3 часа и 18 минут, за прослушиванием музыки – 1 час 26 минут, за видеоиграми – 1 час 10 минут.

С мобильных телефонов люди находятся в сети больше половины от общего времени онлайн. Доля мобильного трафика не меняется глобально с 2017 года – большинство пользователей все-таки продолжают использовать как смартфоны, так и десктоп для доступа в сеть Интернет. Рост мобильного трафика представлен на рисунке 1.1.

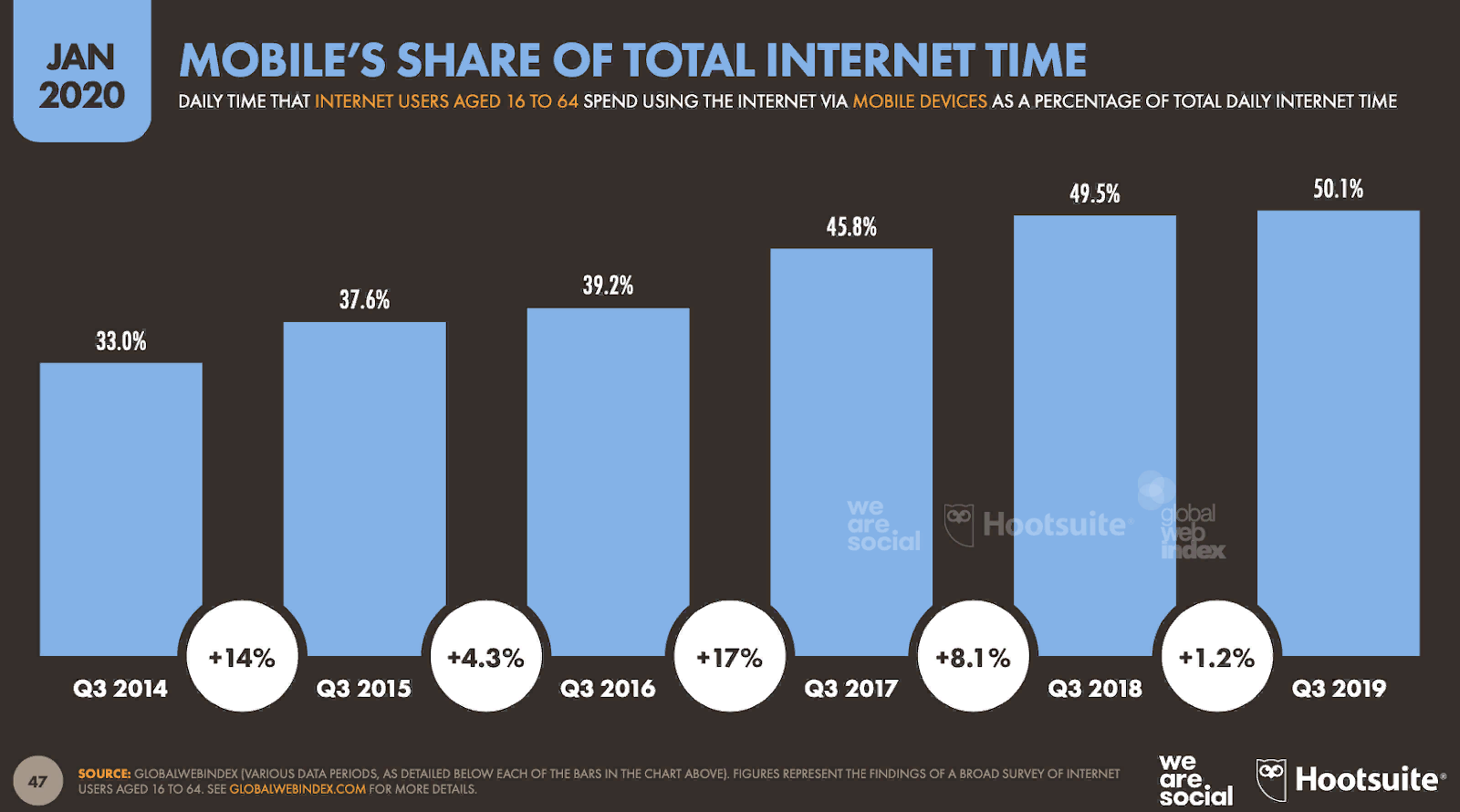


Рисунок 1.1 – Рост доли мобильного трафика с 2014 по 2019 год

Соотношение нахождения в сети с разных устройств:

* доля заходов с мобильных телефонов увеличилась на 8,6% – до 53% пользователей;
* доля заходов с десктопа уменьшилась на 6,8% – теперь компьютером для входа в интернет пользуется 44% пользователей;
* 2.7% пользователей используют для входа в интернет планшет – и это на 27% меньше, чем в 2019 году;
* 0.07% заходят с игровых консолей и других устройств – на 30% меньше, чем в 2019 году.

В мобильных приложениях пользователи проводят больше 90% своего времени – в целом это около 3 часов и 22 минут в день. Половину от общего времени со смартфоном в руках опять же занимают социальных медиа.

Безопасность мобильных приложений. Сегодня безопасность мобильных устройств стала серьезной проблемой как для разработчиков, так и для пользователей. Приложение может стать действительно востребованным только в том случае, если оно может предложить хороший пользовательский опыт, что более важно, безопасный пользовательский интерфейс. Следовательно, обеспечение безопасности мобильного приложения должно быть на первом месте на всех этапах разработки приложений и развертывания приложения для соответствующих мобильных устройств. Обеспечение безопасности на всех этапах разработки мобильных приложений:

* Ранняя интеграция. Большинство уязвимостей безопасности приложений можно предотвратить, плавно интегрируя процессы обеспечения безопасности с самых ранних этапов разработки приложений. Планирование своей первоначальной стратегии разработки приложений, постоянно сохраняющей безопасность, значительно снизит вероятность возникновения рисков безопасности на последующих этапах разработки приложений.
* Предпроектная стадия. Этап включает в себя сбор и анализ данных для разработки приложения. Этот этап также включает понимание документации и других процессов для создания приложения, понимание ОС, для которой разрабатывается приложение.
* Этап разработки приложений. Этап разработки приложения может привести к возникновению множества проблем безопасности. Разумеется, эти проблемы также могут быть рассмотрены относительно легко, когда их заметили на первоначальном этапе. Фактическая проблема, однако, возникает во время реализации дизайна приложения. Проблемы безопасности, возникающие на этом этапе, являются наиболее трудными для определения и устранения. Лучшим способом свести к минимуму фактор риска будет создание списка всех потенциальных ловушек, заблаговременно, а также планирование курса действий, чтобы избежать каждого из них.
* Тестирование и развертывание приложений. Тестирование приложения производится с целью убедиться, что МП полностью соответствует требованиям и полностью защищено.

Дружественный интерфейс мобильных приложений. Разные приложения требуют разных подходов, дизайнерских решений и техник. Тем не менее, сама суть мобильных устройств с сенсорным экраном имеет несколько базовых тенденций:

* Отклик. Это ключевой элемент в мобильном приложении. Приложение всегда должно подтверждать взаимодействие с пользователем. Но важно помнить, что отклик и скорость работы – не одно и то же. Выполнение некоторых операций занимает время, но нужно дать понять пользователю, что задача находится в обработке.
* Область нажатия. Приложения нужно разрабатывать не под указательный, а под большой палец. Область нажатия должна быть не меньше 44px, иначе пользователь может промахнуться и нажать не на ту область, на которую хотел.
* Контент. Сенсорные экраны произвели революцию, поскольку дали возможность напрямую взаимодействовать с контентом. Они устранили такие интерфейсные устройства-посредники как мышь и сенсорная панель, и лучше соответствуют тому, как работает наш мозг.
* Элементы управления. Должны находиться внизу экрана под контентом. Все элементы управления всегда внизу, иначе не будет видно что происходит с контентом во время работы с ним.

Приложения для мобильных устройств предназначены как для работы, так и для развлекательных целей. Одни успешно помогают пользователям контролировать бизнес и вести по нему отчеты, разрабатывать дизайн в оригинальном и фирменном стиле. Другие обеспечивают качественное прослушивание музыки и просмотр фильмов, поддерживают средства общения и выполняют ряд других функций.

На сегодняшний день имеется множество приложений для хозяев и их домашних животных. Они имеют разные назначения. Одни помогают пользователю отслеживать время прогулки и программу питания питомца, другие ведут учет походов к ветеринару и содержат в себе советы по уходу за питомцем. Но это все разные приложения, которые занимают память в мобильном телефоне и затрудняют пользователям их поиск и выбор.

В рамках данного дипломного проекта будет реализовано мобильное приложение под платформу iOS.

Плюсы разрабатываемого приложения заключаются в том, что все необходимые заведения, советы будут находиться в одном приложении, который сможет скачать любой пользователь. Приложение будет бесплатным для пользователя.

## ****1.3 Обзор существующих мобильных платформ****

Наличие операционной системы (ОС) – главная особенность, которая отличает смартфон от обычного мобильного телефона. При выборе конкретной модели телефона или устройства, операционная система часто является определяющим фактором. Наиболее распространенные операционные системы для смартфонов и платформ:

Symbian OS – эта ОС занимала большую часть рынка смартфонов до конца 2010 г. В начале 2010 года на базе данной ОС остается только одна платформа: Series 60, которая используется в основном в устройствах Nokia, а также в некоторых моделях Samsung.

BlackBerry OS – система широко используется в устройствах в первую очередь в Соединенных Штатах, так как спецслужбы некоторых стран не заинтересованы в использовании смартфонов в стране из-за того, что все входящие и исходящие данные зашифрованы с использованием алгоритма шифрования AES.

Windows Mobile и Windows CE – компактная операционная система Microsoft, выпущенные с 1996 года и занимала самый большой сегмент рынка ОС для смартфонов к 2010 году. В настоящий момент не разрабатывается и не поддерживается: в мае 2011 года стало известно о прекращении поддержки разработчиков приложений сервиса Windows Marketplace for Mobile; в июле 2011 года компания сообщила своим клиентам о прекращении работы сервиса 12 My Phone в октябре 2011 года; в мае 2012 года Microsoft закрыла магазин приложений Windows Marketplace for Mobile для владельцев устройств.

Windows Phone – мобильная операционная система, разработанная Microsoft, вышла 11 октября 2010 года. 21 октября начались поставки первых устройств на базе новой платформы. 9 октября 2017 года исполнительный директор Microsoft Джо Бельфиор заявил о прекращении создания новых устройств и обновлений Windows 10 Mobile. В 2019 году компанией Microsoft официально заявлено, что поддержка Windows 10 Mobile будет прекращена 10 декабря 2019 года; владельцам мобильных устройств под управлением этой системы рекомендовано перейти на устройства с Android или iOS.

Palm OS – одна из популярных платформ своего времени, однако в настоящее время мобильные телефоны на базе Palm OS малораспространены и более не выпускаются. Последний смартфон под управлением операционной системы был выпущен в конце 2007 года (Palm Centro).

Linux – широкого распространения эта операционная система на мобильных устройствах не получила, однако ее развитие традиционно считается перспективным направлением. Смартфоны на базе Linux распространяются главным образом в Азии.

Android – портативная (сетевая) операционная система для смартфонов, планшетных ПК, электронные книг, цифровых плееров, часов и нетбуков на базе ядра Linux. Первоначально разработанная Android Inc., которую затем купил Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (OHA), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать приложения на основе Java, который управляют устройством через разработанные Google библиотеки. Android Native Development Kit позволяет системе использовать библиотеки и компоненты приложений, написанных на C и других языках.

ОС iOS (до 24 июня 2010 года – iPhone OS) – это мобильная операционная система, разработанная и изготовленная американской компанией Apple. Она была выпущена в 2007 году; первоначально – для iPhone и iPod Touch, а позже – для таких устройств, как iPad. В 2014 году появилась поддержка автомобильных мультимедийных систем Apple CarPlay. В отличие от Android (Google), выпускается только для устройств, производимых фирмой Apple. В iOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. Ядро iOS почти идентично ядру настольной операционной системы Apple macOS (ранее называвшейся OS X). Начиная с самой первой версии, iOS работает только на планшетных компьютерах и смартфонах с процессорами архитектуры ARM.

## ****1.4 Анализ существующих программных решений****

На сегодняшний день все более актуальными становятся приложения для хозяев и их питомцев. Было найдено несколько вариантов приложений, схожих по функциональным требованиям. Были рассмотрены некоторые из них, которые наиболее схожи с дипломным проектом: «GoDog: Dog Training Clicker», «PetCoach».

**1.4.1** Мобильное приложение GoDog: Dog Training Clicker

Основной функционал GoDog – интерактивные уроки по дрессировке собак – их можно смотреть как в виде обучающей видео-инфографики, так и с помощью технологии AR. Для этого нужно поднести телефон к горизонтальной поверхности и урок перейдет в дополненную реальность.

Для начала работы в приложении нужно зарегистрировать свою собаку (кличка, порода, возраст, добавить фото), после станут доступны разделы приложения – Уроки, Выгул, Чаты, Свисток. В разделе Уроки доступны базовые и продвинутые уроки и игры с собакой. В каждом уроке короткие инструкции-инфографики и количество выполненных подходов, после которых можно отметить урок выполненным. К каждому уроку есть рекомендации при подготовке к уроку.

В разделе Выгул можно отмечать прогулки с собакой: будет отображаться график прогулок, время и расстояние. Таким образом можно посмотреть сколько времени тратится на прогулки и какое расстояние проходит собака. В разделе Чаты можно найти собак, которые гуляют рядом.

Плюсы мобильного приложения:

* современный и простой интерфейс;
* возможность входа и регистрации через социальные сети;
* подсказки при каждом новом действии;
* поддержка версии Android 4.4.

Минусы приложения:

* ограниченный бесплатный функционал.

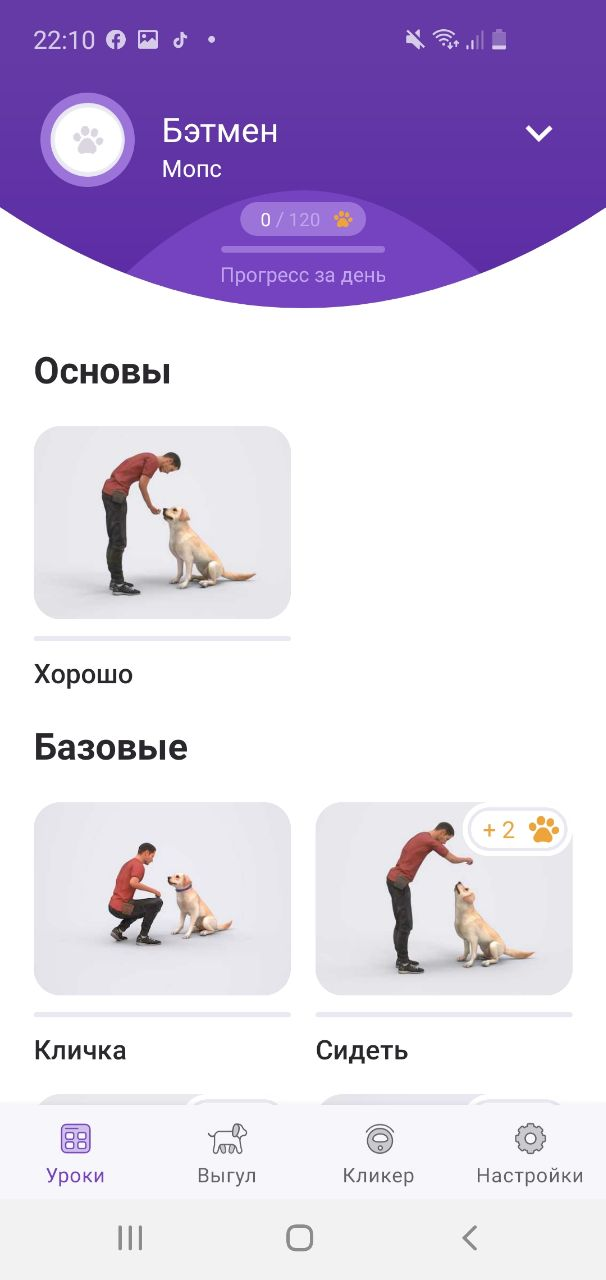


Рисунок 1.2 – Главное окно приложения «GoDog: Dog Training Clicker»

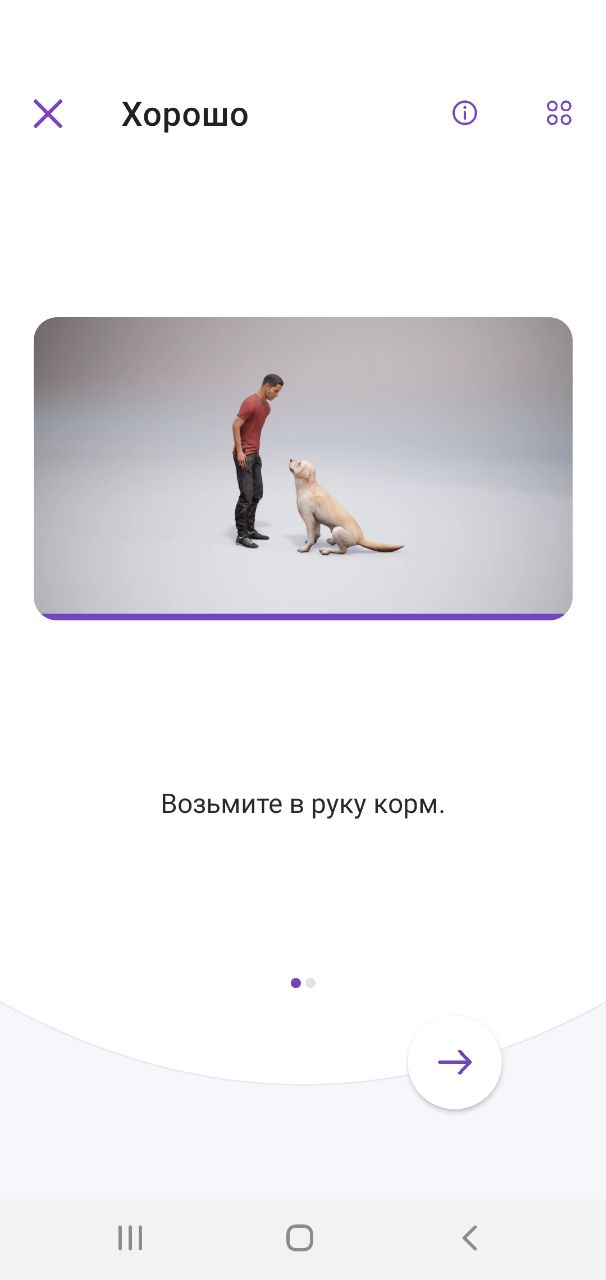


Рисунок 1.3 – Окно с дрессировкой «GoDog: Dog Training Clicker»

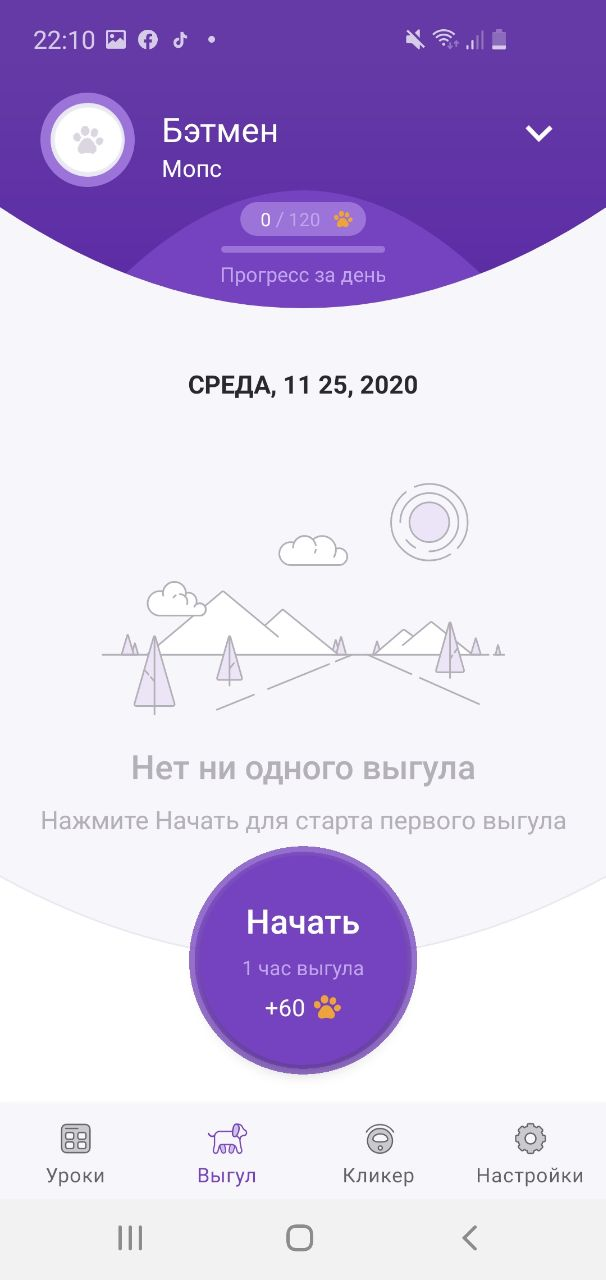


Рисунок 1.4 – Окно с отслеживанием выгула питомца приложения «GoDog: Dog Training Clicker»

**1.4.2** Мобильное приложение PetCoach

В приложении PetCoach присутствуют бесплатные онлайн советы и консультации для владельцев животных от различных специалистов: начиная от ветеринаров и заканчивая дрессировщиками. База содержит множество задаваемых вопросов и познавательных статей с подробными ответами и рекомендациями. Возможно обращение к экспертам в чате программы с изложением сути проблемы в любое время суток. Ответы – подробны, информативны и оперативны. Англоязычный интерфейс, удобная навигация, с гибкими настройками.

Плюсы мобильного приложения:

* удобная навигация с гибкими настройками;
* возможность входа и регистрации через социальные сети;
* современный и простой интерфейс;
* поддержка версии Android 4.4.

Минусы приложения:

* только англоязычный интерфейс;
* ограниченный бесплатный функционал.

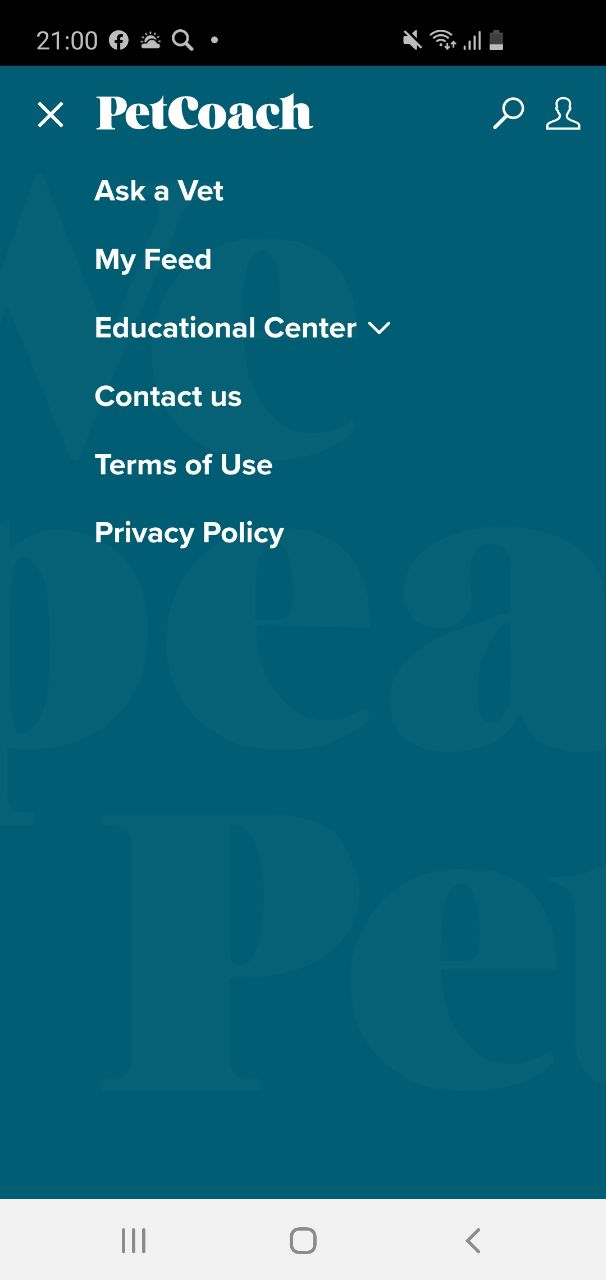


Рисунок 1.5 – Главное меню приложения «PetCoach»

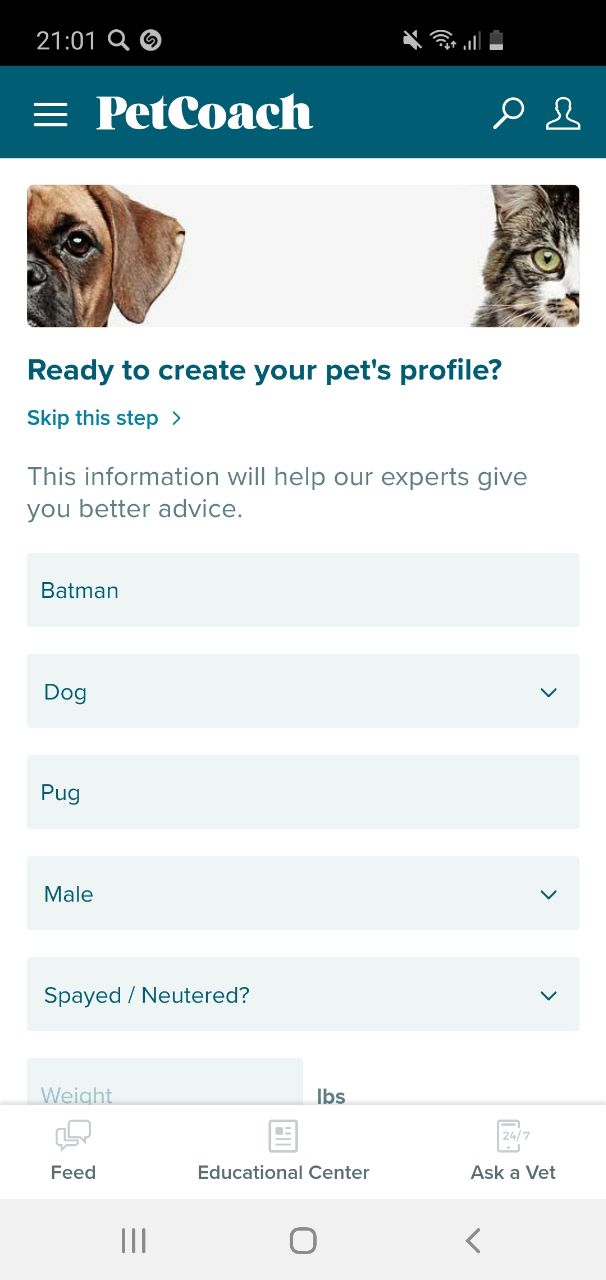


Рисунок 1.6 – Профиль питомца приложения «PetCoach»

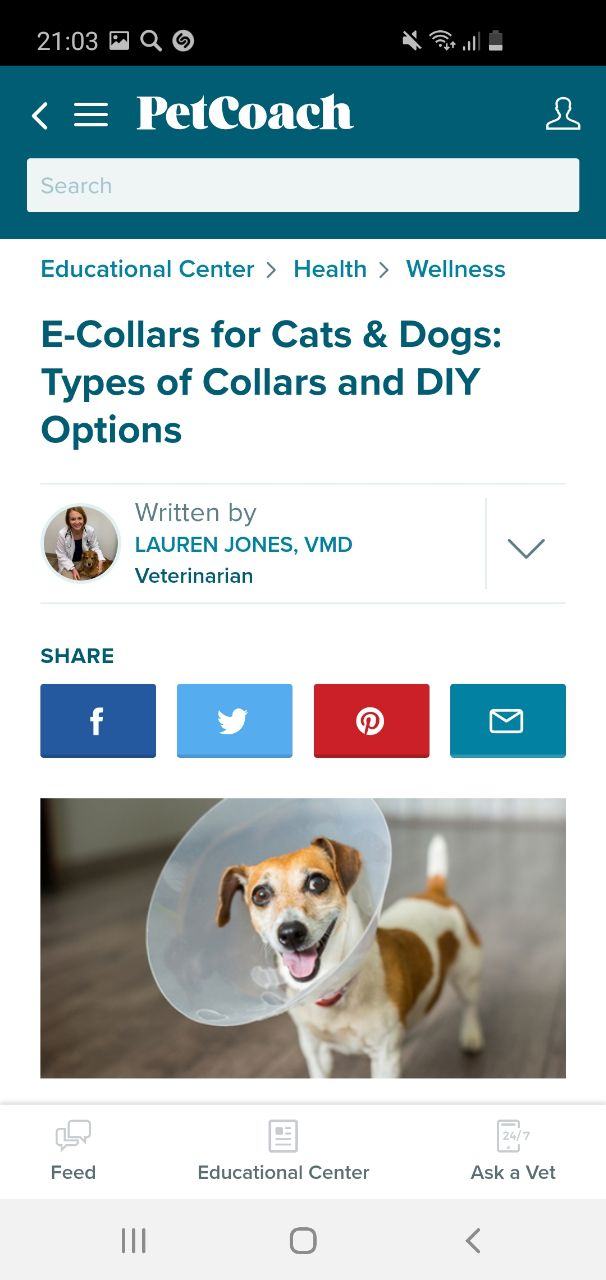


Рисунок 1.7 – Статьи по уходу за питомцем приложения «PetCoach»

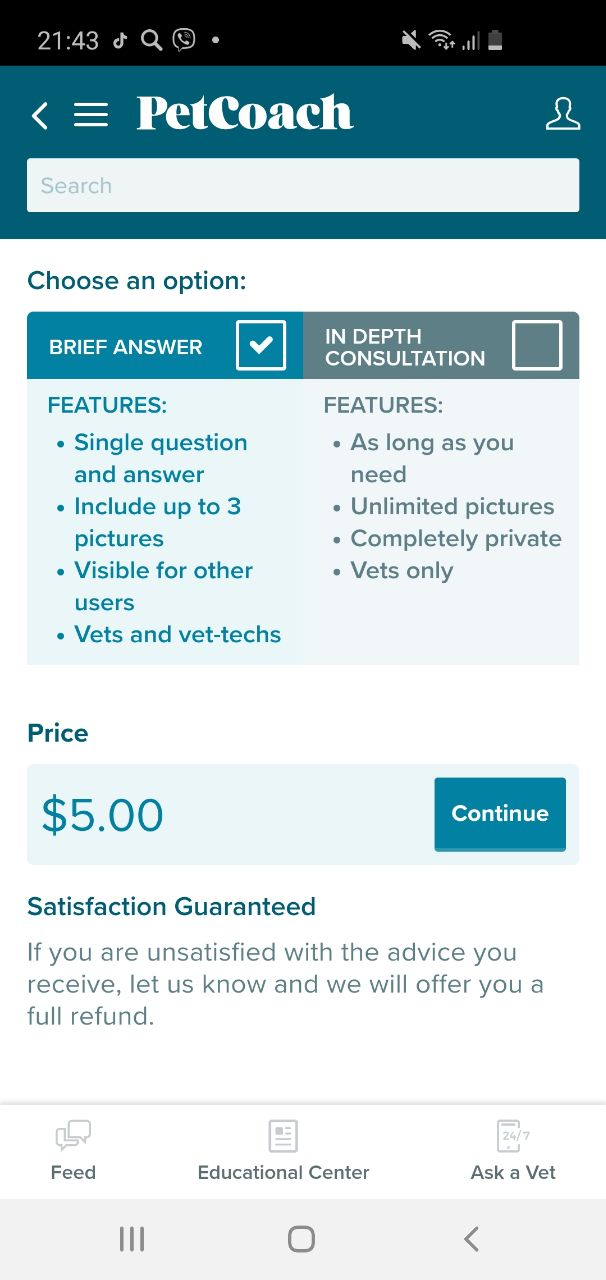


Рисунок 1.8 – Оплата за консультацию ветеринара приложения «PetCoach»

В ходе анализа существующих аналогов были выявлены достоинства и недостатки мобильных приложений. Приложения имеют современный и простой интерфейс, удобный в использовании. Главным минусом приложения является ограниченный бесплатный функционал, также в приложении PetCoach отсутствует русскоязычный интерфейс.

## ****1.5 Постановка цели и задач дипломного проекта****

В ходе анализа существующих аналогов были выявлены достоинства и недостатки мобильных приложений. Приложение, которое будет разработано в ходе дипломного проекта, будет собирая в себе все самые необходимые решения для успешного и простого ухода за своим домашним питомцем.

Целью дипломного проекта является создание мобильного приложения для хозяев и их домашних питомцев с современным и понятным интерфейсом. Мобильное приложение должно выполнять следующие задачи:

* определение местоположения пользователя;
* поиск необходимой информации по уходу за питомцем в приложении;
* поиск ветеринарных клиник на картах;
* запись на прием к ветеринару;
* фильтрация и отображение интересующих статей по уходу за питомцем;
* поиск зоомагазинов на картах;
* поиск выставок;
* запись на выставки на определенную дату.

Приложение должно быть современным и простым в понимании, с удобным интерфейсом, решающее проблемы пользователя.

Интерфейс должен быть дружественным. Приложение должно быть простым и логичным, чтобы пользователь сразу мог понять за что отвечает каждый элемент.

Также приложение должно быть безопасным, чтобы данные пользователя не были утеряны.

Версия для установки операционной системы iOS должна быть не ниже 10.0. Приложение должно быть адаптивно под все разрешения экранов.

# ****2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ****

## ****2.1 Описание функциональности программного продукта****

Для представления функциональной модели была выбрана диаграмма вариантов использования UML, которая отражает отношения между актерами и прецедентами и позволяет описать систему на концептуальном уровне. Диаграмма вариантов использования представлен на рисунке 2.1.

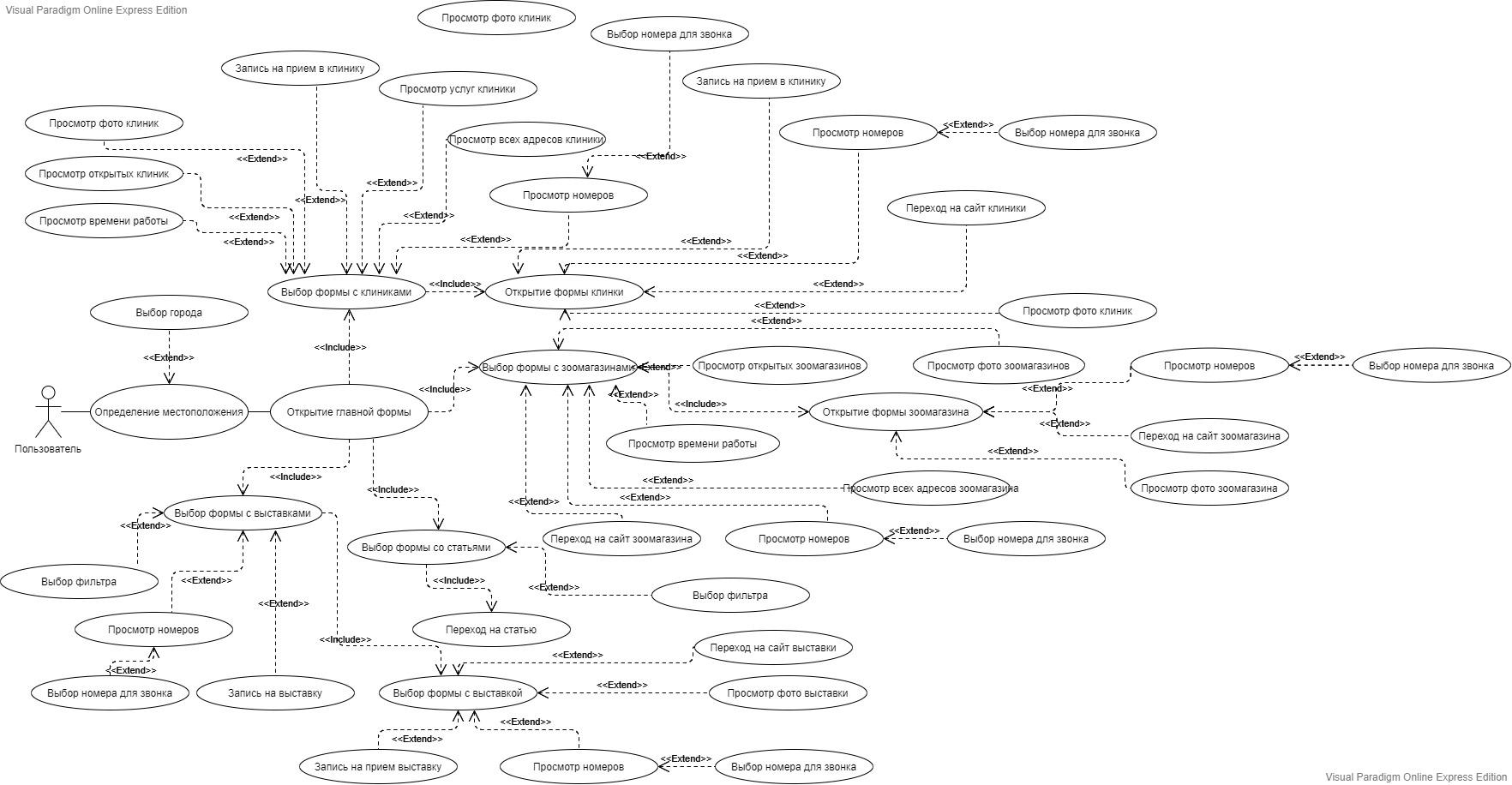


Рисунок 2.1 - Диаграмма вариантов использования

Диаграмма вариантов использования разрабатываемого программного средства в полном развертывании представлена в приложении Б.

На диаграмме можно выделить два основных составляющих элемента – актер и прецедент. Актер – стилизованный человек, обозначающий набор ролей пользователя, взаимодействующего с некоторой сущностью. Прецедент – эллипс с надписью, обозначающий выполняемые системой действия, приводящие к наблюдаемым актером результатам.

Пользователю предоставляются следующие возможности:

* просмотр данных: данные представляет собой информацию, хранящуюся в БД.
* выбор данных: пользователь может просмотреть данные из любой желаемой категории.
* запись: пользователь может записаться на прием в любую интересующую его ветеринарную клинику либо выставку, которая хранится в приложении.

Выборка данных это отбор данных по выбранной категории, с отображением всех аспектов, записанных в базе данных. Так же включает себя выгрузку данных по фильтрам в приложении. Диаграмма вариантов использования представлена в приложении Б.

## ****2.2 Спецификация функциональных требований****

Функциональным назначением разрабатываемого мобильного приложения является предоставление пользователю возможностей для поиска необходимых ветеринарных клиник, зоомагазинов, выставок и статей.

Пользователем в данном случае может выступать любой человек, который имеет мобильный телефон с операционной системой iOS не ниже 10 с доступом к сети Интернет. Для использования мобильного приложения не требуется специальная подготовка или обучение пользователей.

Предполагается возможность одновременной эксплуатации разрабатываемого мобильного приложения большим числом пользователей.

Исходя из предполагаемого использования, можно заключить, что проектируемое МП должно реализовывать следующие группы функций:

* определение местоположения пользователя;
* поиск необходимой информации о клиниках и зоомагазинах в приложении;
* запись на прием к ветеринару;
* фильтрация и отображение интересующих статей по уходу за питомцем;
* поиск выставок;
* запись на выставки на определенную дату.

Версия для установки операционной системы iOS должна быть не ниже 10.0.

## ****2.3 Разработка функциональной модели****

IDEF0 – методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

Она позволяет создать модель, которая будет отражать:

* структуру системы;
* функции.

Модель IDEF0 разворачивается одновременно слева направо и сверху вниз, по диагонали. Объекты, расположенные левее / выше, доминируют над теми, которые находятся правее / ниже. Доминирующие объекты могут включать в себя зависимые: например, доставка заказа – это элемент, входящий в состав более масштабного процесса управления заказами. Также доминирующие объекты могут являться предшествующими этапами для зависимых: получение заявки – согласование заявки.

Графические элементы:

* Прямоугольники – действия или этапы.
* Стрелки – ресурсы, исполнители, необходимые для совершения действия или прохождения этапа.

Главное достоинство IDEF0 – крайне высокая степень детализации, можно создать модель, которая будет учитывать на каждом этапе практически все ресурсы, сотрудников, которые потребуются даже для самых сложных алгоритмов. Недостатком является то, что графическая модель занимает очень много места, её тяжело читать, не имея специальных навыков.

Проектируемое программное средство «Мобильное приложение для владельцев домашних питомцев на платформе iOS MyPet» основные задачи:

* определение местоположения пользователя: позволяет определить город, в котором находится пользователь.
* выбор ветеринарной клиники: позволяет пользователю просматривать сначала ветеринарные клиники, которые находятся поблизости с ним, а после весь список по городу, в котором он находится.
* выбор зоомагазина: позволяет пользователю просматривать зоомагазины, которые находятся поблизости с ним, а после весь список по городу, в котором он находится.
* просмотр выставок: позволяет пользователю просматривать список выставок, которые должны проходить во всех городах.
* поиск статей: пользователь может искать и читать интересующие его статьи.

В качестве входных потоков представлены:

* список ветеринарных клиник;
* список зоомагазинов;
* список выставок;
* статьи по уходу.

В качестве выходных потоков представлены:

* запись на прием к ветеринарному врачу;
* просмотр местоположения зоомагазина;
* запись на выставку;
* чтение статьи.

Функциональную модель программного средства (IDEF0 – диаграмму) и ее декомпозицию можно увидеть на рисунках 2.4 и 2.5.

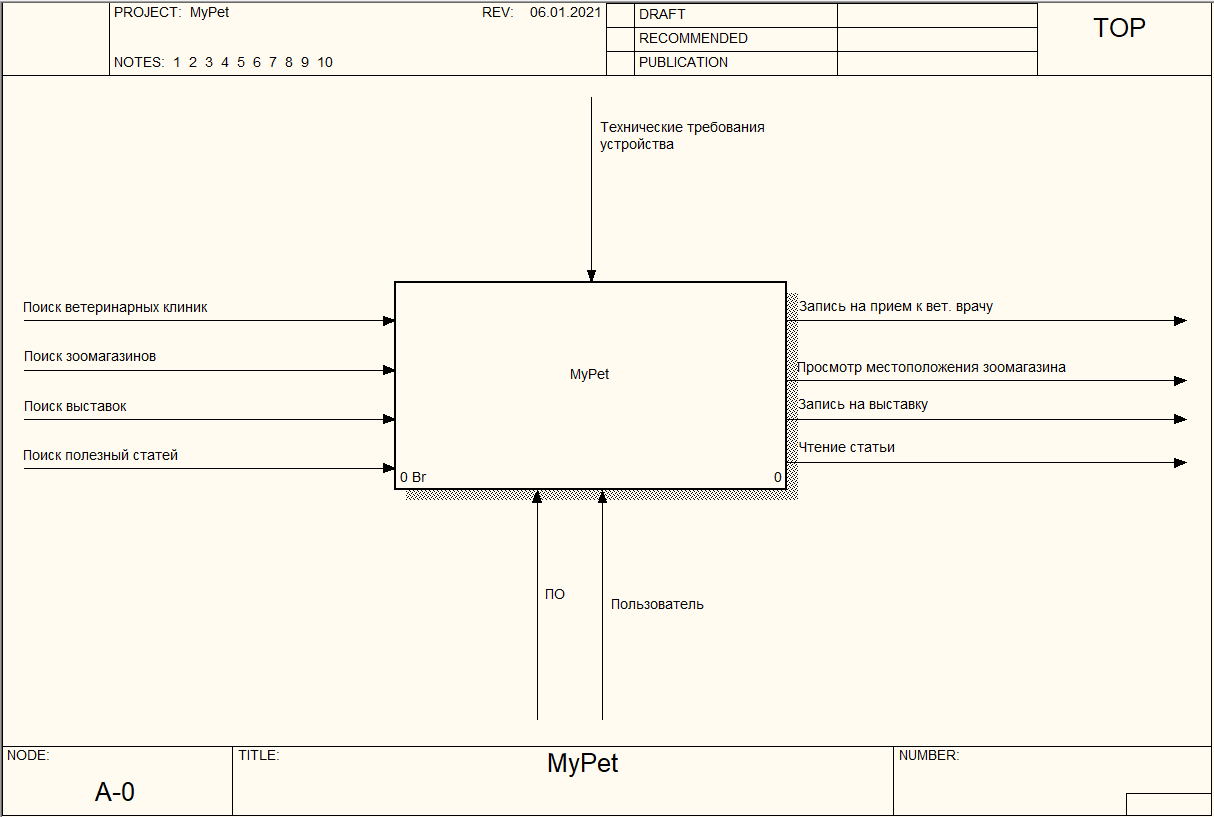


Рисунок 2.4 – Функциональной диаграмма IDEF0

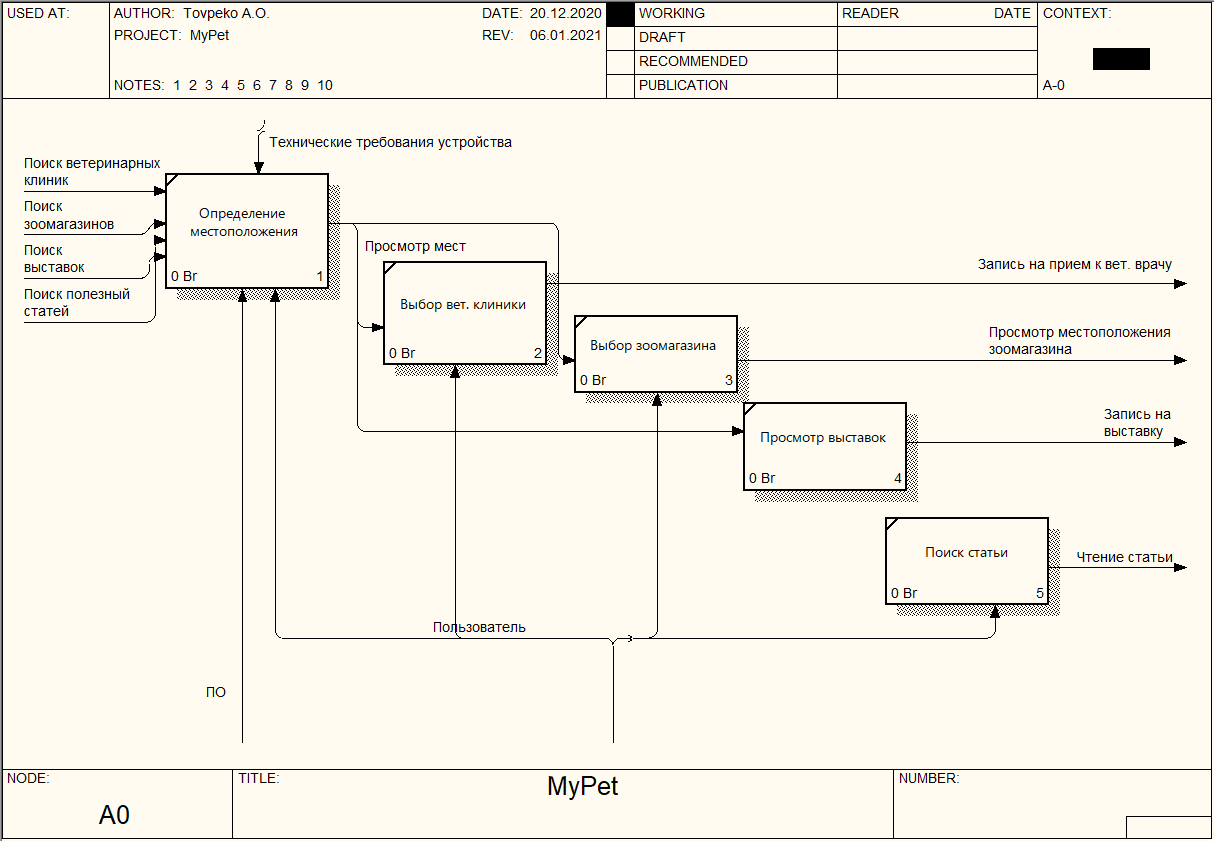


Рисунок 2.5 – Декомпозиция функциональной диаграммы IDEF0

Далее производится функциональная декомпозиция. Второй уровень модели представлен в приложении В. В связи с тем, что само программное средство по будет обладать рядом функций, декомпозируем функциональный блок на пять функциональных блоков.

Функциональный блок №1 – Определение местоположения. Данный функциональный блок описывает логическое представление дальнейшей реализации функции программного средства – определения местоположения.

В функциональном блоке №1 сторона управления включает: технические требования устройства.

Входными значениями стороны входа являются данные пользователя: поиск необходимых мест, статей.

После действий над входными параметрами, выходные значения будут представлены информацией о местах поблизости, полезных статьях и ближайших выставках.

Механизм представлен в виде одного параметра – пользователь.

Функциональный блок №2 – Выбор вет. клиник. Данный функциональный блок описывает выбор ветеринарной клиники исходя из местоположения пользователя, также пользователь может выбрать фильтр по времени и открытым клиникам в текущий момент.

Входным значением является список ветеринарных клиник.

После действий над входными параметрами, выходная информация имеет следующее значения: запись на прием к ветеринарному врачу.

Механизм представлен в виде одного параметра – пользователь.

Функциональный блок №3 – Выбор зоомагазина. Данный функциональный блок описывает выбор зоомагазина исходя из местоположения пользователя, также пользователь может выбрать фильтр по времени и открытым зоомагазинам в текущий момент.

В функциональном блоке №3 входным значением просмотр списка зоомагазинов. После действий над входными параметрами будет отображаться информация и передана на экран мобильного телефона. Механизм представлен в виде одного параметра – пользователь.

Функциональный блок №4 – Просмотр выставок. Данный функциональный блок позволяет пользователю ознакомиться со списком выставок во всех городах Беларуси, а также записаться на участие. Входным значением стороны входа является просмотр мест. После действий над входными параметрами будет отображаться выбранная информация на экран телефона.

Механизм представлен в виде одного параметра – пользователь.

Функциональный блок №5 – Поиск статей. Данный функциональный блок позволяет пользователю ознакомиться с полезными статьями по уходу и содержанию своего питомца. Пользователю будет отображаться выбранная информация на экран мобильного телефона.

Механизм представлен в виде одного параметра – пользователь.

Функциональной диаграмма IDEF0 представлена в приложении Д.

## ****2.4 Разработка информационной модели****

На основании функциональной модели была разработана информационная модель программного решения.

На рисунке 2.6 представлена информационная модель проектируемой системы.

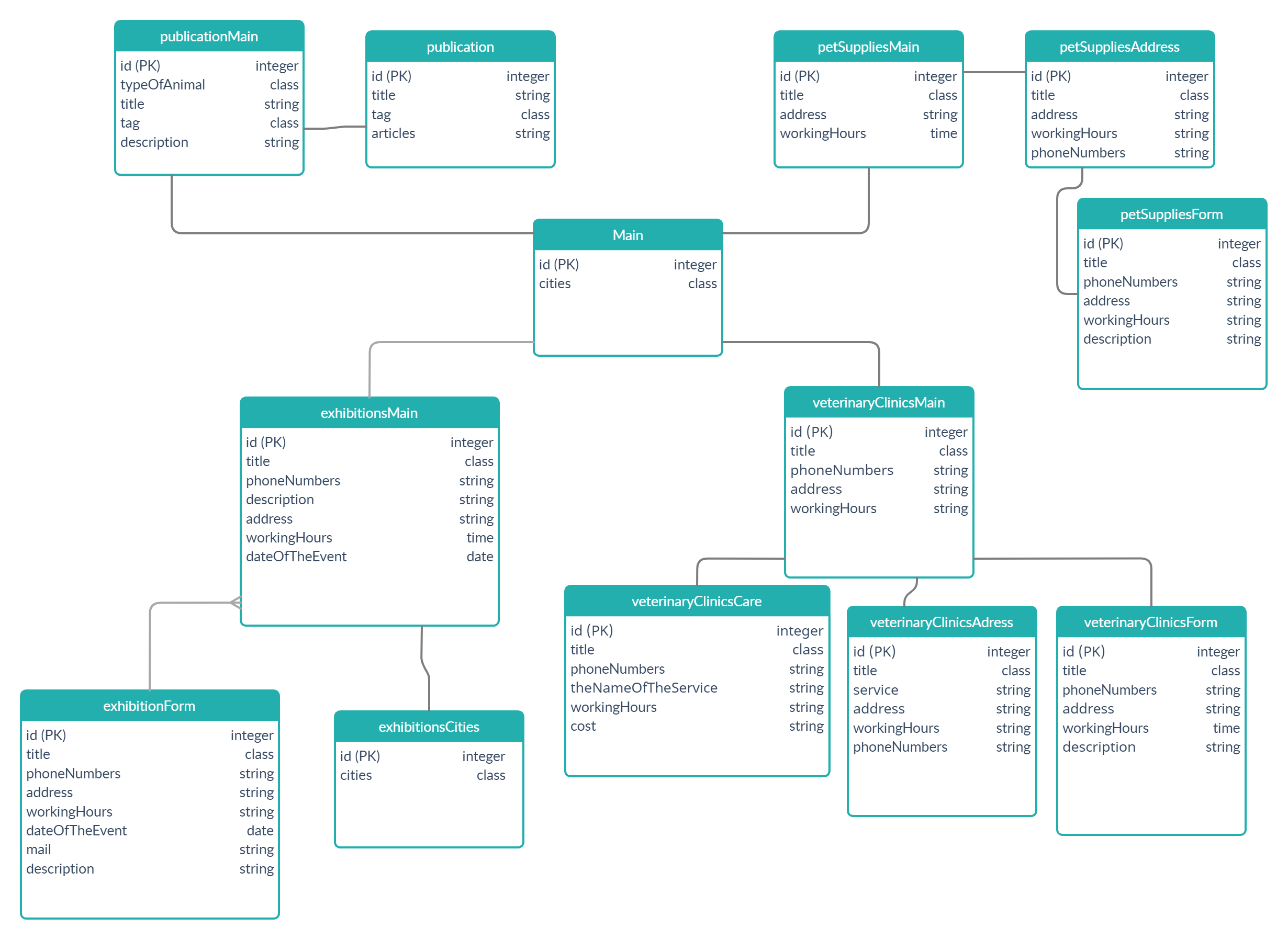


Рисунок 2.6 – Информационная модель проектируемой системы

В процессе анализа предметной области были выделены следующие типы сущностей:

Main – сущность, хранящая в себе главный экран приложения и определение геолокации пользователя. Атрибут Id – идентификатор города, cities – список городов.

veterinaryClinicsMain – сущность, содержащая список ветеринарных клиник определяемых по геолокации пользователя. Сущность обладает следующими атрибутами: id – идентификатор клиники, title – название клиники, phoneNumbers – номера телефонов, address – адрес ветеринарной клиники, workingHours – время работы ветеринарной клиники.

veterinaryClinicsCare – сущность, которая содержит весь перечень услуг и тем выбранной ветеринарной клиники. Атрибут Id – идентификатор услуги, title – название ветеринарной клиники, phoneNumbers – номера телефонов, theNameIfTheService – название предоставляемых услуг, workingHours – время работы, cost – стоимость услуг.

veterinaryClinicsAddress – сущность, содержащая все адреса ветеринарных клиник. Атрибут Id – идентификатор адреса клиники, title – название ветеринарной клиники, address – адрес ветеринарной клиники, workingHours – время работы, phoneNumbers – номера телефонов.

veterinaryClinicsForm – сущность, отображающая страницу выбранной ветеринарной клиники. Атрибут Id – идентификатор страницы клиники, title – название ветеринарной клиники, phoneNumbers – номера телефонов, address – адрес ветеринарной клиники, workingHours – время работы, description – описание клиники.

exhibitionsMain – сущность, содержащая в себе список всех выставок домашних питомцев в Беларуси. Атрибут Id – идентификатор выставки, title – название выставки, phoneNumbers – номера телефонов, description – описание выставки, address – адрес проведения выставки, workingHours – время работы, dateOfTheEvent – дата проведения выставки.

exhibitionsForm – сущность, отображающая страницу выбранной выставки. Атрибут Id – идентификатор страницы выставки, title – название выставки, phoneNumbers – номера телефонов, address – адрес проведения выставки, workingHours – время работы, dateOfTheEvent – дата проведения выставки, email – почтовый адрес, description – описание выставки.

exhibitionsCities – каждый пользователь может выбрать город проведения выставки. Атрибут Id – идентификатор города.

petSuppliesMain – сущность, содержащая список зоомагазинов, определяемых по геолокации пользователя. Атрибут Id – идентификатор зоомагазина, title – название зоомагазина, address – адрес зоомагазинов, workingHours – время работы.

petSuppliesAddress – сущность, содержащая все адреса зоомагазинов. Атрибут Id – идентификатор зоомагазина, title – название зоомагазина, address – адрес зоомагазинов, workingHours – время работы, phoneNumbers – номера телефонов.

petSuppliesForm – сущность, отображающая страницу выбранного зоомагазина. Атрибут Id – идентификатор страница зоомагазина, title – название зоомагазина, phoneNumbers – номера телефонов, address – адрес зоомагазинов, workingHours – время работы, description – описание зоомагазина.

publicationMain – сущность, отображающая страницу со всеми статьями в приложении. Атрибут Id – идентификатор статьи, typeOfAnimal – вид домашнего животного, title – название статьи, tag – тег статьи, description – описание статьи.

publication – сущность, отображающая страницу выбранной статьи. Атрибут Id – идентификатор страницы статьи, title – название статьи, tag – тег статьи, articles – текст статьи.

Информационная модель проектируемой системы представлен в приложении Е.

# ****3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА****

## ****3.1 Разработка модели взаимодействия пользователя с интерфейсом****

В процессе анализа программных решений в области дипломного проектирования, отмечалась необходимость обеспечения высокого уровня интерактивности при взаимодействии с пользователем. На рисунке 2.7 изображена модель взаимодействия пользовательского интерфейса, управляющей логики и данных программного средства, именующаяся MVC.

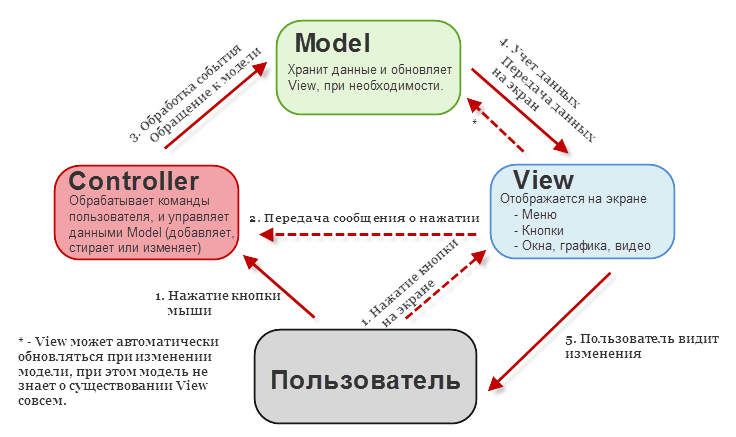


Рисунок 3.1 – Схема MVC

MVC – это паттерн проектирования веб-приложений, который включает в себя несколько более мелких шаблонов. При использовании MVC на три отдельных компонента разделены модель данных приложения, пользовательский интерфейс и логика взаимодействия пользователя с системой, благодаря чему модификация одного из этих компонентов оказывает минимальное воздействие на остальные или не оказывает его вовсе.

Основная цель применения MVC состоит в разделении данных и бизнес-логики от визуализации. За счет такого разделения повышается возможность повторного использования программного кода: добавление представления данных какого-либо существующего маршрута не только в виде HTML, но и в форматах JSON, XML, PDF, XLSX становится очень просто и не требует изменений слоя бизнес-логики исходного маршрута. Также упрощается и сопровождение программного кода: внесение изменений во внешний вид, например, не отражаются на бизнес-логике, а изменения бизнес-логики не затрагивают визуализацию.

Концепция MVC разделяет данные, представление и обработку действий пользователя на компоненты:

* Модель / Model – представляет собой объектную модель некой предметной области, включает в себя данные и методы работы с этими данными, реагирует на запросы из контроллера, возвращая данные и/или изменяя своё состояние. При этом модель не содержит в себе информации о способах визуализации данных или форматах их представления, а также не взаимодействует с пользователем напрямую.
* Представление / View – отвечает за отображение информации (визуализацию). Одни и те же данные могут представляться различными способами и в различных форматах. Например, коллекцию объектов при помощи разных представлений можно представить на уровне пользовательского интерфейса как в табличном виде, так и списком; на уровне API можно экспортировать данные как в JSON, так в XML или XSLX.
* Контроллер / Controller – обеспечивает связь между пользователем и системой, использует модель и представление для реализации необходимой реакции на действия пользователя. Как правило, на уровне контроллера осуществляется фильтрация полученных данных и авторизация – проверяются права пользователя на выполнение действий или получение информации.

## ****3.2 Разработка алгоритма поиска ветеринарных клиник****

Поиск пользователем всех ветеринарных клиник осуществляется через определение его геолокации. Алгоритм поиска ветеринарных клиник представлен на рисунке 3.2.

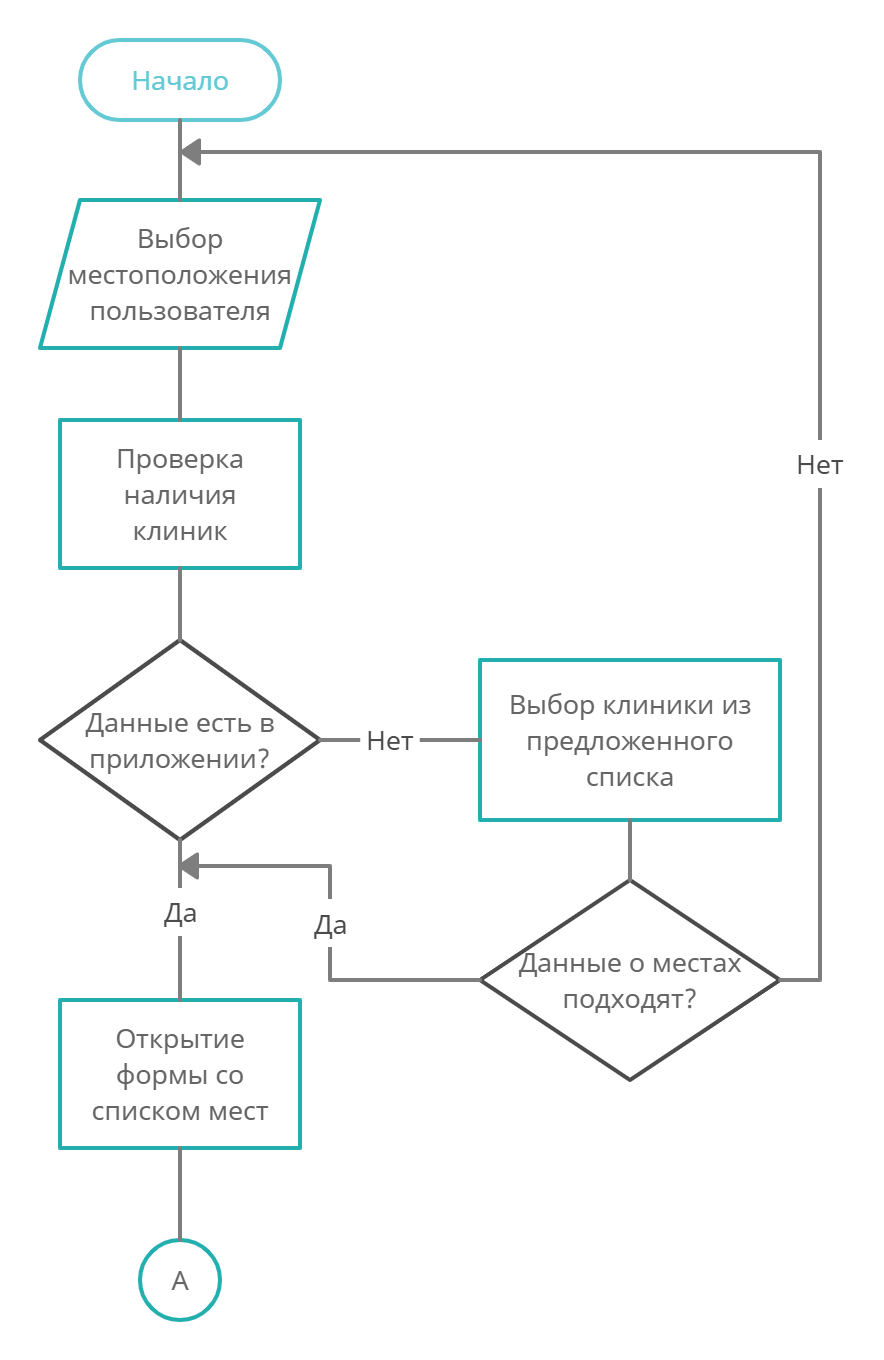


Рисунок 3.2 – Схема алгоритма поиска

Данный алгоритм реализует функцию поиска ближайших ветеринарных клиник к пользователю.

Изначально приложение определяет местоположение пользователя, для того, чтобы в последующем предлагать ему необходимые места поблизости. После определения местоположения программа предоставляет список всех ветеринарных клиник. Если пользователь не нашел ветеринарную клинику в своем городе, то он может посмотреть клиники поблизости к нему. Алгоритм поиска ветеринарных клиник представлен в приложении Ж.

## ****3.3 Разработка алгоритма работы мобильного приложения****

После запуска мобильного приложения появляется сообщение с предложением автоматического определения геолокации пользователя для того, чтобы предлагать искать места поблизости. Пользователь может отклонить сообщение и после выбрать город из списка самостоятельно. Далее пользователь попадает на главный экран приложения, на котором он может выбрать экраны с ветеринарными клиниками, зоомагазинами, выставками и полезными статьями.

После нажатия кнопки «Ветеринарные клиники» пользователь попадает на экран, где предоставлен список всех ветеринарных клиник поблизости. Пользователь может воспользоваться фильтром, выбрав время работы клиники. После того, как пользователь выбрал нужную ему клинику, он переходит на экран с выбранной клиникой. На ней он может записаться на прием, просмотреть все адреса, выбрать номер телефона и позвонить в клинику, перейти на сайт, ознакомиться с информацией и временем работы и дополнительной информацией.

При нажатии на кнопку «Зоомагазины» пользователь попадает на экран со списком зоомагазинов по выбранной геолокации. Он может ознакомиться со всем списком, выбрать зоомагазин удобному для него местоположению. После перехода на экран выбранного зоомагазина пользователь может позвонить в зоомагазин, просмотреть все адреса данной сети, перейти на сайт магазина. Также он может ознакомиться с дополнительной информацией.

После перехода по кнопке «Выставки» пользователю предоставляется информация по всем выставкам, которые должны состояться. Пользователь может выбрать по фильтру самостоятельно дату выставки, выбрать город, в котором будет проходить выставка. После перехода в карточку выбранной выставки пользователь может отправить заявку на участие, просмотреть номера телефонов для связи с организаторами, просмотреть адрес выставки, ознакомиться с дополнительной информацией.

При нажатии на кнопку «Для хозяев» пользователь переходит на экран со статьями по уходу за питомцем. Пользователь может выбрать фильтр, в котором будет указан вид животного, про которого он хочет эту статью прочитать. Выбрать категорию ухода за животным. После выбора статьи пользователь попадает на экран со статьей. Схема алгоритма работы программного средства представлена на рисунке 3.3.

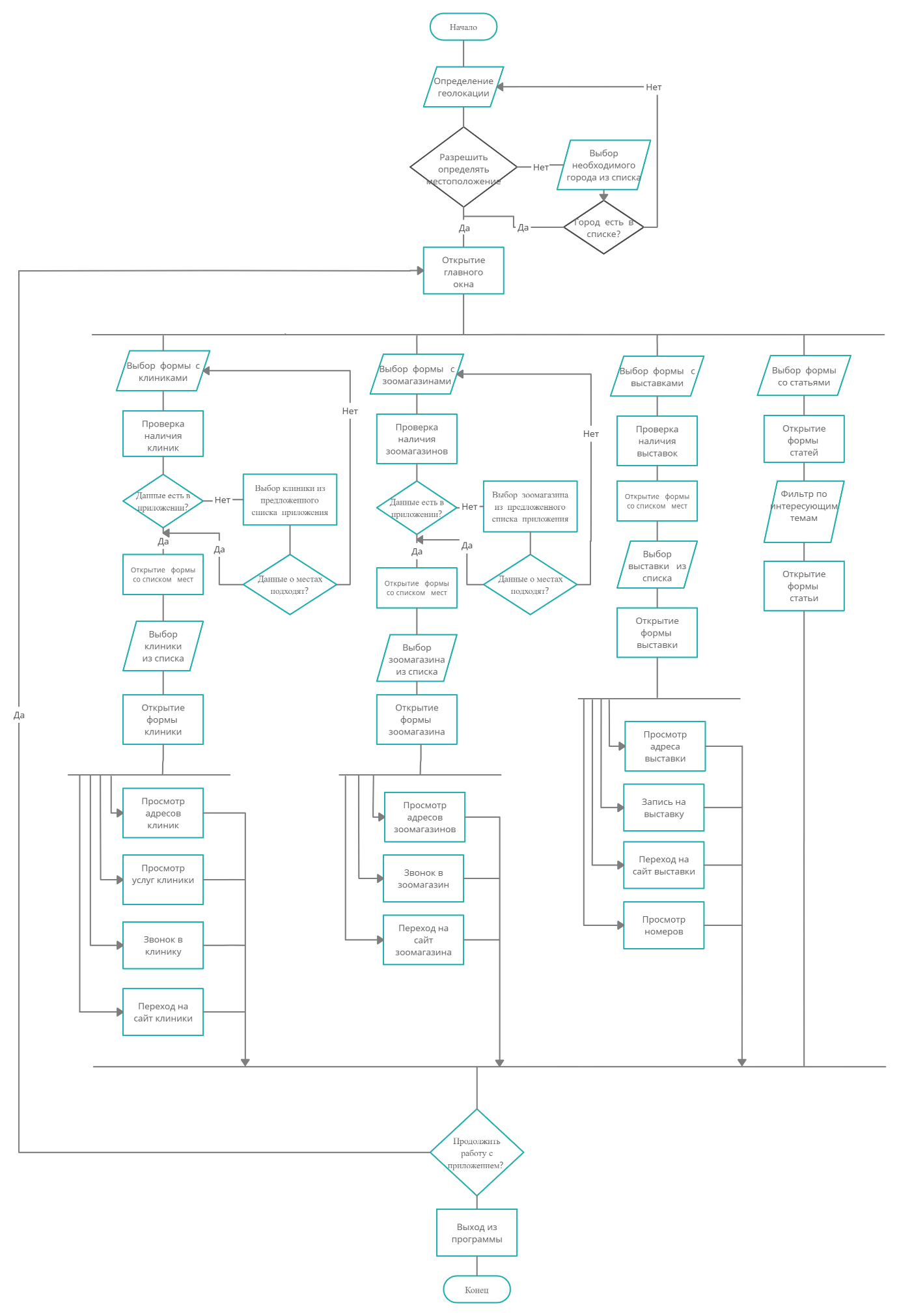


Рисунок 3.3 – Схема алгоритма работы программного средства

После совершения над данными необходимых действий у пользователя есть выбор выйти из программного средства, либо продолжить работу с мобильным приложением.

Алгоритм работы мобильного приложения представлен в приложении З.

## ****3.4 Выбор и обоснование языка и среды разработки программного продукта****

В качестве языка программирования для реализации дипломного проекта был выбран Swift, а среда разработки Xcode и Realm.

В 2014 году компания Apple представила новый язык программирования – Swift. Он стал самым быстрорастущим языком программирования в истории. 3 декабря 2015 года компания открыла исходный код Swift. Всемирное сообщество разработчиков – от создателей приложений до компаний и учебных заведений – может развивать Swift и оптимизировать язык, чтобы сделать его доступным на новых вычислительных платформах.

Swift – язык программирования для разработки приложений под iOS, macOS, watchOS и tvOS.

Swift предоставляет свои собственные версии фундаментальных типов C и Objective-C, включая Int для целых чисел, Double и Float для значений с плавающей точкой, Bool для булевых значений, String для текста. Swift также предоставляет мощные версии трех основных типов коллекций, Array, Set и Dictionary, как описано в разделе Типы коллекций.

Swift использует переменные для хранения и обращения к значениям по уникальному имени. Swift также использует переменные, значения которых не могут быть изменены. Они известны как константы. Константы используются в Swift повсеместно, чтобы сделать код безопаснее и чище в случаях, когда ведется работа со значениями, которые не должны меняться.

В дополнение к знакомым типам, Swift включает расширенные типы. К ним относятся кортежи, которые позволяют создавать и передавать группы значений. Кортежи могут возвращать несколько значений из функции как одно целое значение.

Swift также включает опциональные типы, которые позволяют работать с отсутствующими значениями. Опциональные значения говорят либо «здесь есть значение, и оно равно х», либо «здесь нет значения вообще». Опциональные типы подобны использованию nil с указателями в Objective-C, но они работают со всеми типами, не только с классами. Опциональные значения безопаснее и выразительнее чем nil указатели в Objective-C, и находятся в сердце многих наиболее мощных особенностей Swift.

Swift – язык типобезопасный, что означает, что Swift помогает понять, с какими типами значений код может работать. Если кусок кода ожидает String, безопасность типов не даст вам передать ему Int по ошибке. Кроме того, безопасность типов не позволит случайно передать опциональный String куску кода, который ожидает неопциональный String. Безопасность типов позволяет вам улавливать и исправлять ошибки как можно раньше в процессе разработки [5].

Xcode – это интегрированный программный продукт компании Apple для разработки программного обеспечения для платформ: iOS, macOS, WatchOS и tvOS. Xcode уникален и поддерживает целый ряд технологий, он содержит все, что необходимо разработчику: интуитивно-понятный редактор кода с подсветкой синтаксиса, расширенные возможности отладки программ, простой, но многофункциональный интерфейс и многое другое.

Xcode поддерживает языки программирования: C, C++, Objective-C, Ja-va, AppleScript, Python, Ruby, ResEdit и Swift, с различными моделями программирования, включая, но не ограничиваясь, Cocoa, Carbon и Java. Сторонние разработчики добавили поддержку GNU Pascal, Free Pascal, Ada, C#, Perl и D.

Среда разработки Xcode обеспечивает эффективность работы как небольших, так и крупных команд разработчиков. В Xcode IDE используется схема разделения данных приложения Model-View-Controller (Модель-Представление-Контроллер или MVC) для сегментации каждого слоя приложения. Так проще вносить изменения в код.

С Xcode могут работать индивидуальные разработчики. Программный код проверяется в репозитории Git, после чего им можно делиться с другими. Поддерживается концепт непрерывной интеграции и инструменты тестирования. В текущей версии Xcode также присутствует инструмент Test Assistants — обеспечивает корректность кода и тестов; инструмент для тестирования Test Navigator; поддержка для ботов в Xcode Server, которые запускаются после проверки кода в элементе, есть средства проверки производительности, асинхронности и UI-тестов.

Возможности Xcode:

* Интерфейс единственного окна. Интерфейс Xcode интегрирует редактирование кода, проект пользовательского интерфейса, управление активами, тестирование и отладку в единственном окне рабочей области. Окно реконфигурирует свое содержание во время работы. Например, при выборе файл в одной области соответствующий редактор открывается в другой области. При выборе символа или объекта пользовательского интерфейса его документация появляется в соседней области.
* Редактирование исходного кода. Xcode проверяет исходный код. Когда Xcode замечает ошибку, редактор исходного кода выделяет ее и, когда возможно, предлагает фиксировать. Xcode ускоряет ввод с интеллектуальным завершением кода и предоставляет возможность сократить свой ввод далее с готовыми к использованию фрагментами кода и шаблонами исходного файла.
* Графический проект UI. Интерфейсный разработчик является редактором визуального проектирования, который он интегрируется в Xcode. Использование интерфейсного разработчика позволяет создавать пользовательский интерфейс iOS или приложений Mac путем сборки окон, представлений, средств управления, меню и других элементов от библиотеки конфигурируемых объектов. Функция Auto Layout позволяет определить ограничения для объектов так, чтобы они автоматически корректировались к размеру экрана, размеру окна и локализации. С классами размера можно настроить свой мобильный UI для любой комбинации размера экрана и ориентации: настройка автоматических ограничений макета, добавление или удаление представлений и даже изменение шрифта.
* Интегрированная отладка. При запуске Xcode сразу же запускается сеанс отладки. Можно отладить приложение непосредственно в редакторе исходного кода. В то время, как разработчик исследуете код, область отладки и навигатор отладки позволяют тщательно управлять выполнением своего приложения. Для более удобного управления консоль предоставляет доступ командной строки к отладчику. Приборы отладки выводят на экран потребление ресурсов приложения, чтобы помочь идентифицировать проблемы до того, как это заметят пользователи.
* Тестирование и непрерывная интеграция. Xcode включает в себя платформу для тестирования производительности и функционирования приложения. Он дает возможность писать тесты и использовать тестовый навигатор для того, чтобы отслеживать результат. Тесты производительности помогают определить, что важные части приложения выполняют свою работу правильно [3].

База данных Realm спроектирована, как объектно-ориентированная база данных. Realm отличается от других аналогичных библиотек тем, что она рассматривает объекты данных как живые объекты – это значит, что объекты обновляются синхронно, мгновенно реагируя на изменения и легко их сохраняя. Мобильная база данных Realm распространяется с открытым исходным кодом с 2016 года и бесплатна для разработчиков.

Мобильная база данных Realm предоставляет особые преимущества уже готовые к использованию:

* Производительность и безопасность потока. Благодаря использованию собственного движка, обеспечивающего высокую скорость работы и простоту в применении, мобильная база данных Realm выполняет запросы и синхронизирует объекты значительно быстрее, чем Core Data и SQLite, и осуществляет параллельный доступ к данным без проблем. Это значит, что несколько источников могут получить доступ к одному и тому же объекту без необходимости управлять блокировкой или каких-либо проблем с несогласованностью данных. Данная особенность является серьезным конкурентным преимуществом. Практика показывает, что в большинстве случаев Realm значительно превосходит в скорости не только SQLite, но и другие альтернативные ORM для Android, такие как ORMLite и Greendao.
* Шифрование. Мобильная база данных Realm предлагает службы шифрования для защиты базы на диске с помощью AES-256 + SHA2 64-разрядного шифрования.
* Кроссплатформенность. В отличие от основных данных, мобильная база данных Realm действительно кроссплатформенная и поддерживает iOS, Android, JavaScript веб-приложения и Xamarin [6].

## ****3.5 Безопасность операционной системы iOS****

Концепция безопасности системы дополняет уникальные возможности оборудования Apple, обеспечивая максимальную безопасность операционных систем на устройствах Apple без ущерба для удобства пользователей. Средства безопасности системы охватывают процесс загрузки, обновление программного обеспечения и текущую работу операционной системы.

В основе безопасной загрузки лежит аппаратное обеспечение, с которого начинается цепочка доверия, объединяющая все модули программного обеспечения. Прежде чем передать управление следующему звену, каждое звено убеждается в том, что следующее звено функционирует должным образом. Эта модель безопасности поддерживает не только стандартную загрузку устройств Apple, но и различные режимы восстановления и обновления на устройствах iOS, iPadOS и macOS.

Самые новые версии iOS, iPadOS и macOS являются самыми безопасными. Механизм обновления программного обеспечения не только следит за своевременным обновлением устройств Apple, но и гарантирует, что будет установлено только проверенное программное обеспечение Apple. Система обновления также предотвращает атаки через понижение версии, так что злоумышленник не может откатить устройство до более старой версии операционной системы (которую он умеет взламывать) для кражи пользовательских данных.

Безопасная последовательность загрузки, а также функции обеспечения безопасности системы и приложений помогают следить за тем, чтобы на устройстве запускались только надежные приложения и фрагменты кода. В устройствах Apple также реализованы дополнительные функции шифрования для защиты данных пользователей даже в случае компрометации других частей системы безопасности (например, в случае пропажи устройства или запуска ненадежного кода). Все эти функции имеют большое значение для пользователей и ИТ-администраторов, поскольку они обеспечивают постоянную защиту личной и корпоративной информации, а также предоставляют средства для мгновенного и полного удаленного стирания в случае потери или кражи устройства.

На устройствах iOS и iPadOS используется технология шифрования данных, также именуемая технологией защиты данных, а для защиты данных на компьютерах Mac используется технология шифрования тома под названием FileVault. Основой иерархии ключей в обеих моделях является специализированная микросхема Secure Enclave (на устройствах с процессором SEP). Кроме того, в обеих моделях используется выделенный AES-модуль, который поддерживает шифрование на полной скорости и избавляет от необходимости предоставлять ключи шифрования ядру операционной системы или центральному процессору (где они могут быть скомпрометированы).

Кроме того, ядра операционной системы обеспечивают принудительное использование средств контроля доступа, чтобы предотвратить несанкционированный доступ к данным. Эти средства контроля доступа зачастую представляют собой песочницы для приложений (этот механизм ограничивает доступ приложения к данным) и обеспечивают использование механизма Data Vault. Механизм Data Vault полностью противоположен песочницам. Вместо того, чтобы ограничивать запросы, которые может делать приложение, механизм Data Vault ограничивает доступ к защищенным данным (опять же посредством ядра, независимо от файлового шифрования). При этом не имеет значения, проходил ли изначальный процесс в песочнице или нет.

Приложения являются одним из самых важных элементов современной архитектуры безопасности. Приложения не только значительно повышают продуктивность работы пользователей, но и, если не принять должных мер, могут негативно сказываться на безопасности системы, ее стабильности и пользовательских данных.

Поэтому Apple реализует многоуровневую систему защиты для гарантии того, что приложения не содержат известного вредоносного программного обеспечения и не были подделаны. Дополнительные меры защиты обеспечивают тщательный контроль доступа приложений к пользовательским данным, выполняемого при посредничестве системы. Эти средства создают стабильную, безопасную платформу для работы приложений, позволяя тысячам разработчиков предлагать сотни тысяч приложений для iOS, iPadOS и macOS без ущерба для целостности системы. А пользователи могут получать доступ к этим приложениям со своих устройств Apple, не опасаясь вирусов, вредоносного кода или атак.

Для обеспечения самого жесткого контроля все приложения на iPhone, iPad и iPod touch могут быть получены только из App Store – и каждое работает в своей изолированной среде.

Многие приложения на Mac доступны через App Store, однако пользователи Mac также загружают и используют приложения из интернета. Для безопасной загрузки из интернета в macOS применяются дополнительные меры защиты. Во-первых, по умолчанию в macOS 10.15 и новее для запуска на Mac все приложения должны быть заверены компанией Apple. Благодаря этому требованию гарантируется, что приложения не содержат известного вредоносного программного обеспечения, а разработчики получают возможность распространять такие приложения не только через App Store. Кроме того, macOS включает передовые средства защиты от вирусов, предусматривающие блокирование и удаление вредоносного ПО.

Механизм песочниц, применяемый на всех платформах, помогает дополнительно защитить пользовательские данные от несанкционированного доступа со стороны приложений [2].

## ****3.6 Проектирование и разработка программного интерфейса****

Интерфейс пользователя (user interface или сокращенно UI) – это интерфейс, с помощью которого человек может управлять программным обеспечением или аппаратным оснащением. UI должны быть удобными в использовании, чтобы взаимодействие с ними происходило на максимально интуитивном уровне. Интерфейсы программного обеспечения и сервисов также называют графическими пользовательскими интерфейсами (graphical user interface или GUI).

Опыт пользователя (user experience или сокращенно UX) – это восприятие и ответные действия пользователя, возникающие в результате использования и (или) предстоящего использования продукции, системы или услуги.

Опыт пользователя включает все эмоции, убеждения, предпочтения, ощущения, физические и психологические реакции пользователя, поведение и достижения, которые возникают до, во время и после использования системы. Опыт пользователя сочетает образ торговой марки, способ представления, функциональность, производительность системы, интерактивное поведение и вспомогательные возможности интерактивной системы, физическое и психологическое состояние пользователя, являющееся результатом предшествующего опыта, привычек, навыков и индивидуальности.

Основное достоинство хорошего интерфейса пользователя заключается в том, что пользователь всегда чувствует, что он управляет программным обеспечением, а не программное обеспечение управляет им. Для создания у пользователя такого ощущения «внутренней свободы» интерфейс должен обладать целым рядом свойств:

Естественность интерфейса. Естественный интерфейс – такой, который не вынуждает пользователя существенно изменять привычные для него 38 способы решения задачи. Это, в частности, означает, что сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений. Целесообразно также сохранить систему обозначений и терминологий, используемые в данной предметной области.

Согласованность интерфейса. Согласованность позволяет пользователям переносить имеющиеся знания на новые задания, осваивать новые аспекты быстрее, и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на уяснение различий в использовании тех или иных элементов управления, команд. Обеспечивая преемственность полученных ранее знаний и навыков, согласованность делает интерфейс узнаваемым и предсказуемым.

Дружественность интерфейса. Пользователи обычно изучают особенности работы с новым программным продуктом методом проб и ошибок. Эффективный интерфейс должен принимать во внимание такой подход. На каждом этапе работы он должен разрешать только соответствующий набор действий и предупреждать пользователей о тех ситуациях, где они могут повредить системе или данным; еще лучше, если у пользователя существует возможность отменить или исправить выполненные действия.

Принцип обратной связи. Необходимо всегда обеспечивать обратную связь для действий пользователя. Каждое действие пользователя должно получать визуальное, а иногда и звуковое подтверждение того, что программное обеспечение восприняло введенную команду; при этом вид реакции, по возможности, должен учитывать природу выполненного действия.

Простота интерфейса. Интерфейс должен быть простым. При этом имеется в виду не упрощенчество, а обеспечение легкости в его изучении и в использовании. Кроме того, он должен предоставлять доступ ко всему перечню функциональных возможностей, предусмотренных данным приложением.

Гибкость интерфейса. Гибкость интерфейса – это его способность учитывать уровень подготовки и производительность труда пользователя. Свойство гибкости предполагает возможность изменения структуры диалога и (или) входных данных.

Эстетическая привлекательность. Проектирование визуальных компонентов является важнейшей составной частью разработки программного интерфейса. Корректное визуальное представление используемых объектов обеспечивает передачу весьма важной дополнительной информации о поведении и взаимодействии различных объектов. В то же время следует помнить, что каждый визуальный элемент, который появляется на экране, потенциально требует внимания пользователя, которое, как известно, не безгранично. Необходимо обеспечить формирование на экране такой среды, которая не только содействовала бы пониманию пользователем представленной информации, но и позволяла бы сосредоточиться на наиболее важных ее аспектах.

Интерфейс пользователя необходимо проектировать и разрабатывать как отдельный компонент создаваемого приложения. Необходимо учитывать возможности и особенности аппаратно-программных средств, на базе которых реализуется интерфейс. Целесообразно учитывать особенности и традиции той предметной области, к которой относится создаваемое приложение. Процесс разработки интерфейса должен носить итерационный характер, его обязательным элементом должно быть согласование полученных результатов с потенциальным пользователем. Средства и методы реализации интерфейса должны обеспечивать возможность его адаптации к потребностям и характеристикам пользователя.

# 4 ****ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА****

## 4.1 ****Методы и типы тестирования****

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестирование – важный этап разработки любого программного обеспечения. В ходе этого процесса проверяется соответствие реального состояния ПС с ожидаемым (или установленным), основываясь на наборе тест-кейсов.

Можно определить такие основные цели тестирования программного обеспечения:

– предоставление информации о качестве ПО конечному заказчику;

– повышение качества ПО;

– предотвращение появления дефектов.

Цели тестирования могут отличаться, в зависимости от этапа разработки ПО, на котором оно проводится. К примеру, на этапе кодирования целью тестирования будет вызов как можно большего количества сбоев в работе программы, что позволит локализовать и исправить дефекты. В то же время, при приемочном тестировании необходимо показать, что система работает правильно. В период сопровождения, тестирование в основном необходимо для того, чтобы убедится в отсутствии новых багов, появившихся во время внесения изменений.

Модульное тестирование, иногда блочное тестирование или юнит-тестирование – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы, наборы из одного

или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.

Интеграционное тестирование – одна из фаз тестирования программного обеспечения, при которой отдельные программные модули объединяются и тестируются в группе. Обычно интеграционное тестирование проводится после модульного тестирования и предшествует системному тестированию.

[Системное тестирование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)  – это уровень тестирования, который проверяет законченный и полностью интегрированный программный продукт. Целью системного теста является оценка сквозных технических характеристик системы.

Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного программного обеспечения (ПО) исходным функциональным требованиям компании клиента. То есть проведение функционального тестирования позволяет проверить способность информационной системы в определенных условиях решать задачи, нужные пользователям.

## 4.2 ****Тест-кейсы****

Тест-кейс – это профессиональная документация тестировщика, последовательность действий, направленная на проверку какой-либо функциональности, описывающая как прийти к фактическому результату. Тест-кейсы необходимы для проверки продукта без ознакомления с всей документацией.

Каждый тест-кейс обязательно должен содержать следующие атрибуты:

− уникальный идентификатор;

− наименование;

− описание;

− шаги проверки;

− ожидаемый результат.

Каждый выполненный тест-кейс может иметь один из трех результатов:

− положительный результат, если фактический результат равен ожидаемому результату;

− отрицательный результат, если фактический результат не равен ожидаемому результату (была обнаружена ошибка);

− выполнение теста блокировано, если после одного из шагов продолжение теста невозможно (найдена ошибка).

Таблица 4.1 – Тест-кейс модуля «Запуск мобильного приложения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Главный экран приложения. Запуск приложения | 1. Переход в меню приложений  2. Сделать клик по иконке приложения. | 1. Приложение запустилось | Тест успешно пройден |
| 2 | Главный экран приложения. Выбор города | 1. Открытие главного экрана приложения.  2. Появление сообщения с предложением автоматического определения геолокации. | 1. 1. Корректное определение геолокации пользователя | Тест успешно пройден |

Продолжение таблицы 4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 3 | Главный экран приложения. Выбор города | 1. Выбрать город из списка  2. Нажать кнопку «Применить» | 1. 1. Отображение списка с городами, сохранение выбранных данных | Тест успешно пройден |

Таблица 4.2 – Тест-кейс модуля «Список ветеринарных клиник»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Главный экран приложения. Переход на экран «Ветеринарные клиники» | 1. Нажатие на кнопки «Вет.клиники» | 1. Переход на экран со списком ветеринарных клиник | Тест успешно пройден |
| 2 | Экран со списком ветеринарных клиник. Выбор фильтра по времени работы | 1. Клик на фильтр «Время работы»  2. Выбор времени работы из появившегося списка. | 1. 1. Отображение ветеринарных клиник по времени работы | Тест успешно пройден |
| 3 | Экран со списком ветеринарных клиник. Выбор фильтра «Открыто» | 1. Клик на фильтр «Открыто» | 1. 1. Отображение ветеринарных клиник, открытых в текущий момент | Тест успешно пройден |
| 4 | Экран со списком ветеринарных клиник. Кнопка «Позвонить» | 1. Клик на иконку с изображением кнопки. | 1. 1. Отображение номеров телефонов ветеринарной клиники | Тест пройден успешно |

Продолжение таблицы 4.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 5 | Экран со списком ветеринарных клиник. Кнопка «Записаться» | 1. Клик на кнопку «Записаться». | 1. 1. Переход на форму записи на прием в ветеринарную клинику | Тест пройден успешно |
| 6 | Экран со списком ветеринарных клиник. Кнопка «Услуги» | 1. Клик на кнопку «Услуги». | 1. 1. Переход на форму со списком услуг ветеринарной клиники | Тест пройден успешно |
| 7 | Экран со списком ветеринарных клиник. Кнопка «Все адреса» | 1. Клик на кнопку «Все адреса». | 1. 1. Переход на форму со списком всех адресов ветеринарной клиники | Тест пройден успешно |

Таблица 4.3 – Тест-кейс модуля «Услуги ветеринарной клиники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Экран с услугами ветеринарной клиники. Кнопка «Записаться» | 1. Клик на кнопку «Записаться». | 1. 1. Переход на форму записи на прием в ветеринарную клинику | Тест пройден успешно |
| 2 | Экран с услугами ветеринарной клиники. Кнопка «Позвонить» | 1. Клик на иконку с изображением кнопки. | 1. 1. Отображение номеров телефонов ветеринарной клиники | Тест пройден успешно |

Продолжение таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 3 | Экран с услугами ветеринарной клиники.  Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт ветеринарной клиники | Тест пройден успешно |

Таблица 4.4 – Тест-кейс модуля «Все адреса ветеринарной клиники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | Экран со всеми адресами ветеринарной клиники. Кнопка «Записаться» | 1. Клик на кнопку «Записаться». | 1. 1. Переход на форму записи на прием в ветеринарную клинику | Тест пройден успешно |
| 2 | Экран со всеми адресами ветеринарной клиники. Кнопка «Позвонить» | 1. Клик на иконку с изображением кнопки. | 1. 1. Отображение номеров телефонов ветеринарной клиники | Тест пройден успешно |
| 3 | Экран со всеми адресами ветеринарной клиники. Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт ветеринарной клиники | Тест пройден успешно |
| 4 | Экран со всеми адресами ветеринарной клиники. Кнопка «Услуги» | 1. Клик на кнопку «Услуги». | 1. 1. Переход на форму со списком услуг ветеринарной клиники | Тест пройден успешно |

Таблица 4.5 – Тест-кейс модуля «Экран ветеринарной клиники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | Экран ветеринарной клиники. Кнопка «Записаться» | 1. Клик на кнопку «Записаться». | 1. 1. Переход на форму записи на прием в ветеринарную клинику | Тест пройден успешно |
| 2 | Экран ветеринарной клиники. Кнопка «Позвонить» | 1. Клик на иконку с изображением кнопки. | 1. 1. Отображение номеров телефонов ветеринарной клиники | Тест пройден успешно |
| 3 | Экран ветеринарной клиники. Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт ветеринарной клиники | Тест пройден успешно |

Таблица 4.6 – Тест-кейс модуля «Список зоотоваров»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Главный экран приложения. Переход на экран «Зоотовары» | 1. Нажатие на кнопки «Зоотовары» | 1. Переход на экран со списком зоотоваров | Тест успешно пройден |
| 2 | Экран со списком зоотоваров. Выбор фильтра по времени работы | 1. Клик на фильтр «Время работы»  2. Выбор времени работы из появившегося списка. | 1. 1. Отображение ветеринарных клиник по времени работы | Тест успешно пройден |

Продолжение таблицы 4.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 3 | Экран со списком зоотоваров. Выбор фильтра «Открыто» | 1. Клик на фильтр «Открыто» | 1. 1. Отображение зоотоваров, открытых в текущий момент | Тест успешно пройден |
| 4 | Экран со списком зоотоваров. Кнопка «Позвонить» | 1. Клик на иконку с изображением кнопки. | 1. 1. Отображение номеров телефонов зоотоваров | Тест пройден успешно |
| 5 | Экран со списком зоотоваров. Кнопка «Все адреса» | 1. Клик на кнопку «Все адреса». | 1. 1. Переход на форму со списком всех адресов зоотоваров | Тест пройден успешно |
| 6 | Экран со списком зоотоваров. Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт зоотоваров | Тест пройден успешно |

Таблица 4.7 – Тест-кейс модуля «Все адреса зоотоваров»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | Экран со всеми адресами зоотоваров. Кнопка «Позвонить» | 1. Клик на кнопку «Позвонить». | 1. 1. Отображение номеров телефонов зоотоваров | Тест пройден успешно |
| 2 | Экран со всеми адресами зоотоваров. Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт зоотоваров | Тест пройден успешно |

Таблица 4.8 – Тест-кейс модуля «Экран зоотоваров»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | Экран зоотоваров. Кнопка «Позвонить» | 1. Клик на кнопку «Позвонить». | 1. 1. Отображение номеров телефонов зоотоваров | Тест пройден успешно |
| 2 | Экран зоотоваров. Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт зоотоваров | Тест пройден успешно |
| 3 | Экран зоотоваров. Просмотр фото | 1. Открытие фото на странице зоотоваров | 1. 1. Фото открываются | Тест пройден успешно |

Таблица 4.9 – Тест-кейс модуля «Экран статей»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | Главный экран приложения. Переход на экран «Для хозяев» | 1. Клик на кнопку «Для хозяев». | 1. 1. Открытие формы со статьями | Тест пройден успешно |
| 2 | Экран «Для хозяев». Фильтр | 1. Клик на фильтр по нужным категориям | 1. 1. Отображение статей по выбранному фильтру | Тест пройден успешно |
| 3 | Экран «Для хозяев».  Переход к выбранной статье | 1. Клик на интересующую статью | 1. 1. Открытие формы со статьей | Тест пройден успешно |

Таблица 4.10 – Тест-кейс модуля «Список выставок»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Главный экран приложения. Переход на экран «Выставки» | 1. Нажатие на кнопки «Выставки» | 1. Переход на экран со списком выставок | Тест успешно пройден |
| 2 | Экран со списком выставок. Выбор фильтра по дате проведения выставки | 1. Клик на фильтр «Дата»  2. Выбор даты проведения из появившегося списка. | 1. 1. Отображение выставок по выбранной дате | Тест успешно пройден |
| 3 | Экран со списком выставок. Выбор фильтра по городу проведения выставки | 1. Клик на фильтр «Город»  2. Выбор города проведения из появившегося списка. | 1. 1. Отображение выставок по выбранному городу | Тест успешно пройден |
| 4 | Экран со списком выставок. Выбор фильтра по виду животного | 1. Клик на фильтр «Собаки» | 1. 1. Отображение выставок для собак | Тест пройден успешно |
| 5 | Экран со списком выставок. Кнопка «Записаться» | 1. Клик на кнопку «Записаться». | 1. 1. Переход на форму записи на выставку | Тест пройден успешно |

Продолжение таблицы 4.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 6 | Экран со списком выставок. Кнопка «Позвонить» | 1. Клик на иконку с изображением кнопки. | 1. 1. Отображение номеров телефонов выставок | Тест пройден успешно |

Таблица 4.11 – Тест-кейс модуля «Экран выставки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № тест-кейса | Тестируемая функциональность | Последовательность действий | Ожидаемый  Результат | Полученный результат |
| 1 | Экран выставки.Кнопка «Позвонить» | 1. Клик на кнопку «Позвонить». | 1. 1. Отображение номеров телефонов выставки | Тест пройден успешно |
| 2 | Экран выставки.Кнопка «На сайт» | 1. Клик на кнопку «На сайт». | 1. 1. Переход на сайт выставки | Тест пройден успешно |
| 3 | Экран выставки. Кнопка «Записаться» | 1. Клик на кнопку «Записаться». | 1. 1. Переход на форму записи на выставку | Тест пройден успешно |

Приложение также проходило тестирование с помощью прямого взаимодействия с пользователем по приведенным тест-кейсам, где при каждом «прогоне» тест-кейсов наблюдалось соответствие ожидаемому результату.

Для проведения тестирования были выбраны следующие мобильные устройства: iPhone 8 Plus, iPhone 5s, iPhone 11 Pro Max. Также были рассмотрены следующие версии ОС iOS: v10, v11, v12, v13, v14.

## ****4.3 Результаты тестирования****

В результате тестирования были обнаружены следующие ошибки:

– некорректное отображение данных по фильтрам;

– ошибка подключения к БД;

Для устранения найденных ошибок были созданы проверки на действие пользователя, на корректные выбор данных и определения местоположения.

Тестирование является основным методом отладки, измерения качества и определения реальных характеристик программ и информации баз данных на любых этапах их жизненного цикла.

Ошибка подключения к базе данных. Ошибка подключения к базе данных представлена на рисунке 4.1. Использование необъявленного типа" RLMObjectSchema ".

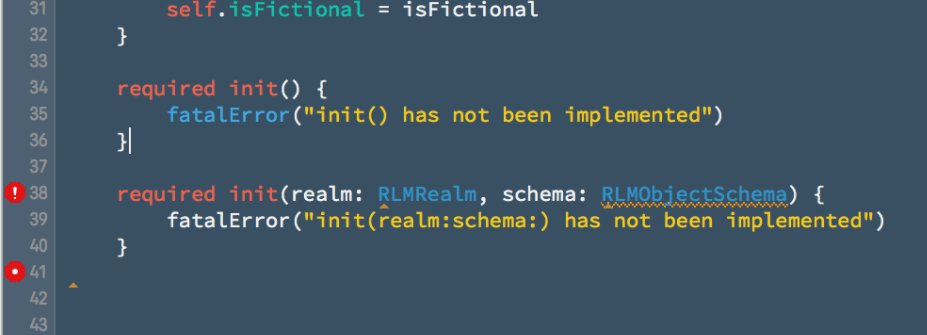


Рисунок 4.1– Ошибка подключения к БД

В результате данной ошибки необходимо создать инициализатор удобства для каждого элемента.

Некорректное отображение данных по фильтрам представлена на рисунке возникает в том случае, если тег в статье, по которому происходит выборка данных, указан не верно.

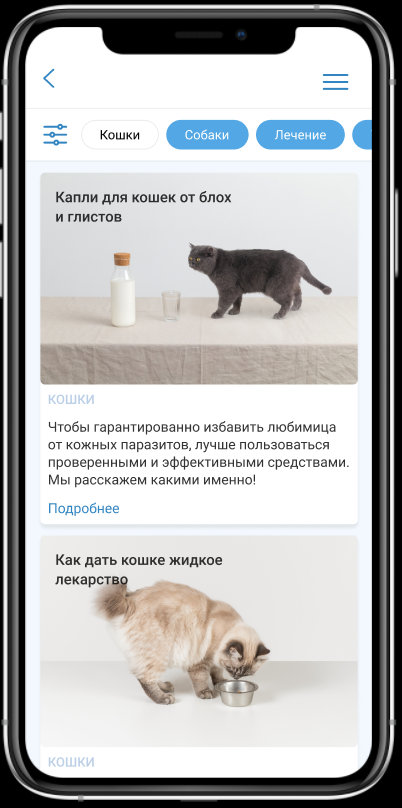


Рисунок 4.2 – Некорректное отображение данных

В случае этой ошибки необходимо перераспределить теги в базе данных.

Результатом описанных выше этапов тестирования явились ошибки. Каждой ошибке был присвоен свой уровень важности, в зависимости от влияния на работу приложения. Ошибки с более высоким приоритетом были исправлены в первую очередь.

Результаты тестирования приложения, а также статистика найденных и исправленных ошибок представлена в таблице 4.12.

Таблица 4.12 – Результаты тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Критических | Важных | Средних | Незначительных | Всего |
| Найдено ошибок | 0 | 1 | 1 | 0 | 2 |
| Исправлено ошибок | 0 | 1 | 1 | 0 | 2 |

По результатам тестирования можно сделать вывод, что полученные характеристики разработанного программного средства приемлемыми и созданное программное средство работает корректно. Все выявленные ошибки были проанализированы и успешно устранены.

# ****5 РУКОВОДСТВО ПО УТСАНОВКЕ И ИСПОЛЬЗОВАНИЮ****

MyPet – это мобильное приложение для операционной системы iOS, которое позволяет получать данные пользователю о местоположении ближайших к нему ветеринарных клиники, зоомагазинов, выставках и читать полезные статьи по уходу и содержанию своего питомца.

## ****5.1 Требования к программным и аппаратным средствам****

Приложение предназначено для работы на мобильных устройствах под

управлением операционной системы iOS не ниже 10 версии.

## ****5.2 Работа с приложением «MyPet»****

**5.2.1** Главное окно приложения

После запуска приложения появляется сообщение с предложением об автоматическом определении геолокации пользователя. Экран с определением геолокации показан на рисунке 5.2

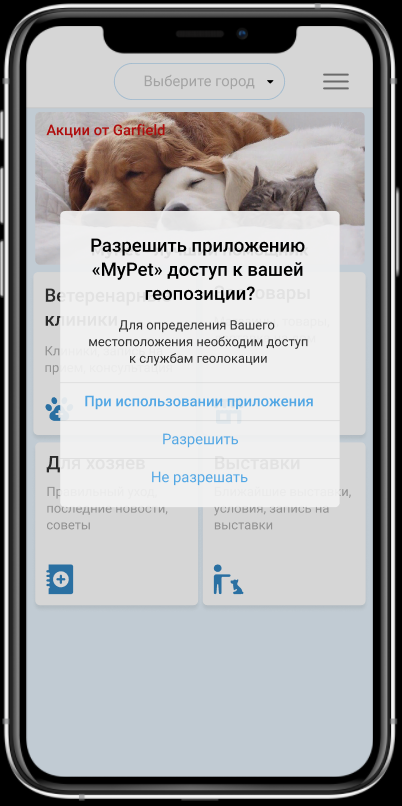


Рисунок 5.2 – Экран с определением геолокации

После разрешения/запрета определения геолокации пользователь попадает на главный экран приложения (рисунок 5.3). Также пользователь может сам на главном экране выбрать город, в котором будет просматривать интересующие его места (рисунок 5.4).

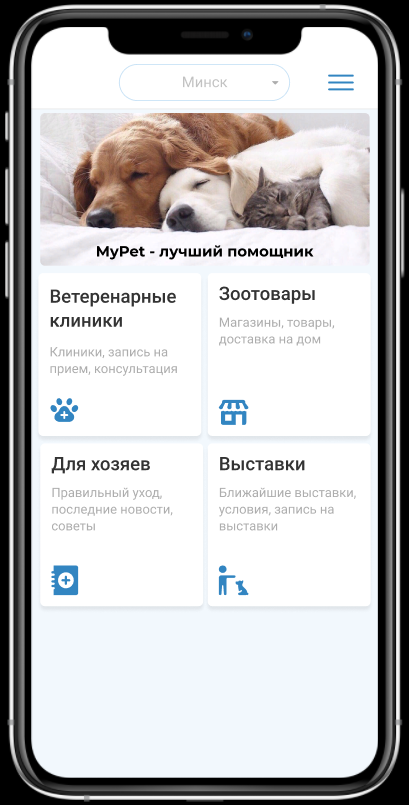


Рисунок 5.3 – Главный экран приложения

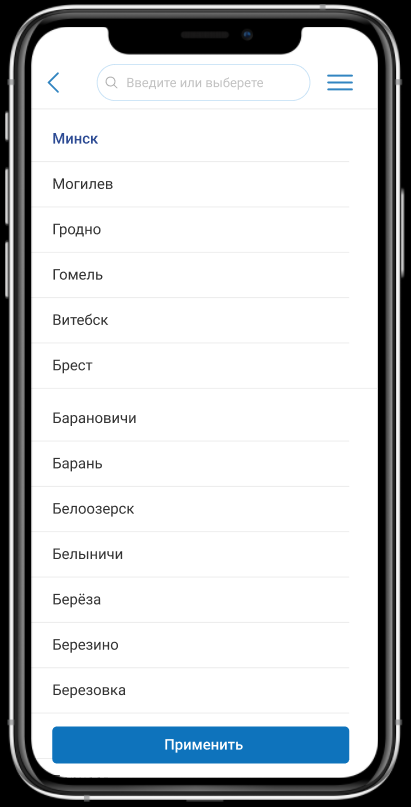


Рисунок 5.4 – Экран с поиском города

**5.2.2** Форма «Ветеринарные клиники»

После перехода на экран со списком ветеринарных клиник пользователь может выбрать удобную клинику по своему текущему местоположению. Ознакомиться с информацией, просмотреть фото и перейти на экран с выбранной клиникой. Также пользователь может выбрать фильтр по времени работы ветеринарной клиники, либо же выбрать клинику, которая открыта в данный момент. Экран со списком ветеринарных клиник представлен на рисунке 5.5.

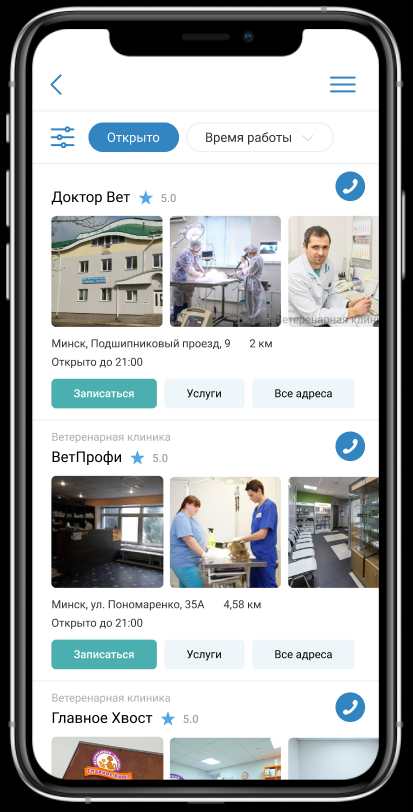


Рисунок 5.5 – Экран со списком ветеринарных клиник

Пользователь может просмотреть услуги, которые предоставляет данная ветеринарная клиника, посмотреть все адреса клиник, выбрать номер телефона для звонка. Также пользователь может записаться на прием в клинику. Данные экраны представлены на рисунках 5.6, 5.7, 5.8.

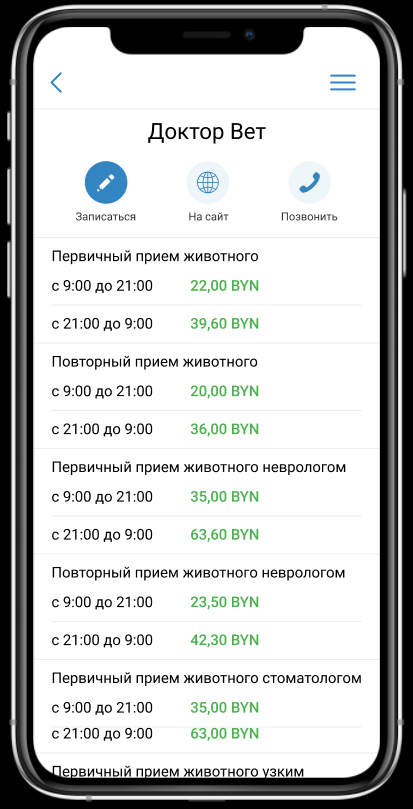


Рисунок 5.6 – Экран с услугами ветеринарной клиники

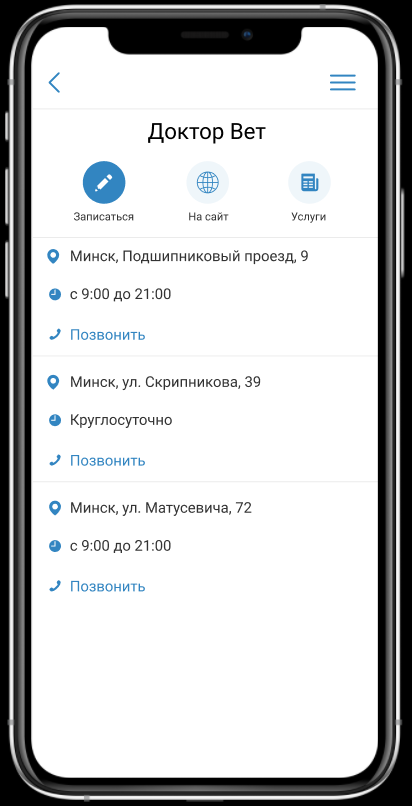


Рисунок 5.7 – Рисунок со всеми адресами ветеринарной клиники

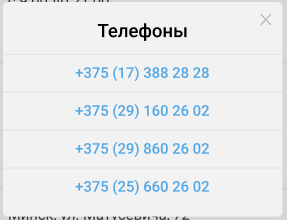


Рисунок 5.8 – Номера ветеринарной клиники

**5.2.3** Форма «Экран ветеринарной клиники»

После просмотра списка клиник пользователь может выбрать самую подходящую и перейти на экран клиники. На выбранном экране пользователь может ознакомиться более подробно с информацией о клинике, просмотреть фотографии, время работы, адрес. Пользователь также может перейти на форму записи на прием, перейти на сайт клиники и посмотреть номера для звонка. Экран ветеринарной клиники представлен на рисунке 5.9.

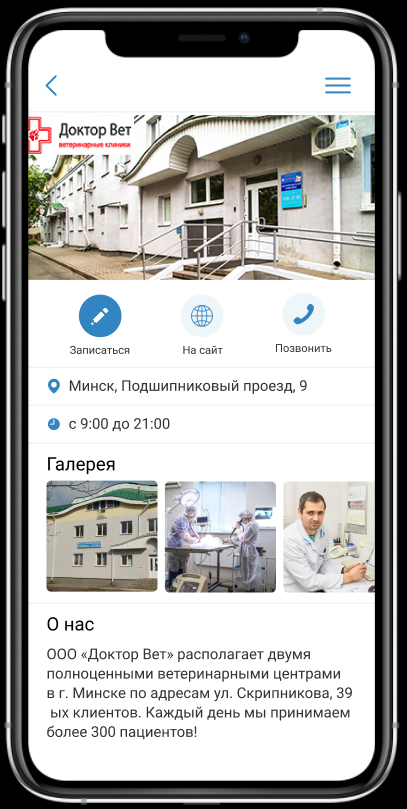


Рисунок 5.9 – Экран ветеринарной клиники

**5.2.4** Форма «Экран со списком зоомагазинов»

После перехода на экран со списком зоомагазинов пользователю предоставляется выбор из мест, которые находятся к нему ближе всего. Пользователь может ознакомиться с информацией о зоомагазине, просмотреть фото и перейти на экран с выбранным зоомагазином. Также пользователь может выбрать фильтр по времени работы ветеринарной клиники, либо же выбрать клинику, которая открыта в данный момент. Экран со списком зоомагазинов представлен на рисунке 5.10.

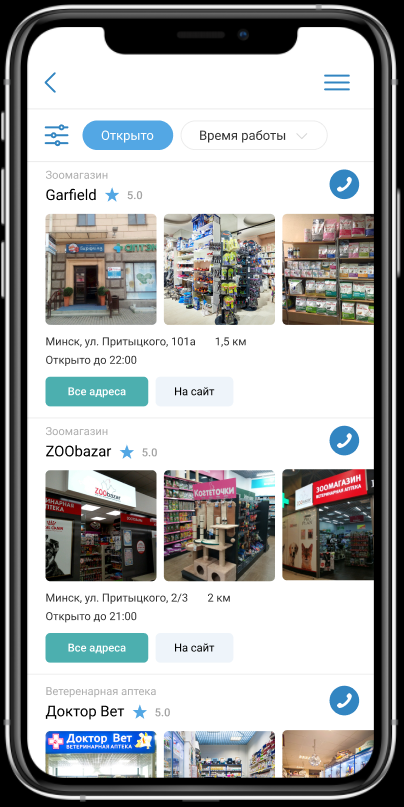


Рисунок 5.10 – Экран со списком зоомагазинов

После просмотра списка зоомагазинов пользователь может перейти на экран магазина, который он выбрал (рисунок 5.11).

Также пользователь может перейти на сайт зоомагазина, просмотреть все адреса, просмотреть номера телефонов и позвонить по ним. Данные экраны представлены на рисунках 5.12, 5.13.



Рисунок 5.11 – Экран зоомагазина

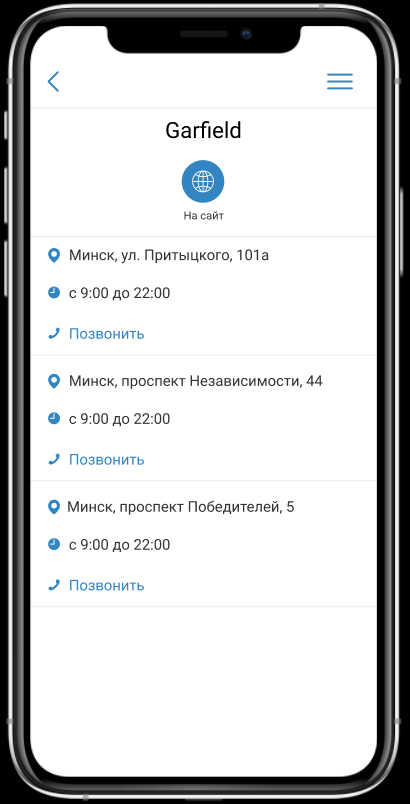


Рисунок 5.12 – Экран со всеми адресами зоомагазина

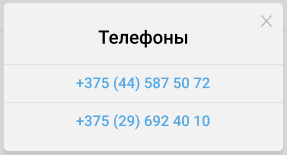


Рисунок 5.13 – Номера телефонов зоомагазина

**5.2.5** Форма «Экран со списком выставок»

После перехода на экран со списком выставок пользователю предоставляется список всех выставок. Пользователь может ознакомиться с информацией о выставках, просмотреть фото и перейти на экран с выбранной выставкой. Также пользователь может выбрать фильтр по времени проведения выставки, города, и вида животного. Экран со списком выставок представлен на рисунке 5.14.



Рисунок 5.14 – Экран со списком выставок

Пользователь может выбрать выставку и записаться на нее. На выбранном экране пользователь может более подробно ознакомиться с информацией о выставке, просмотреть фотографии, время работы, адрес, контакты для связи. Пользователь также может перейти на форму записи на участие в выставке, перейти на сайт выставки и посмотреть номера для звонка. Экраны представлены на рисунках 5.15.

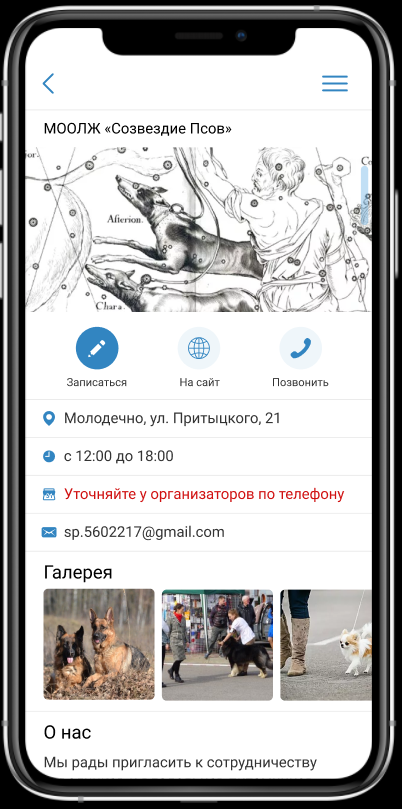


Рисунок 5.15 – Экран выставки

**5.2.6** Форма «Экран со списком статей»

После перехода на экран со списком статей пользователь может с помощью фильтра выбрать статьи по уходу за определенным питомцем. Также пользователь может выбрать вид статьи, которую хочет прочитать. 11111Экран представлен на рисунке 5.16. На рисунке 5.17 изображен экран статьи.

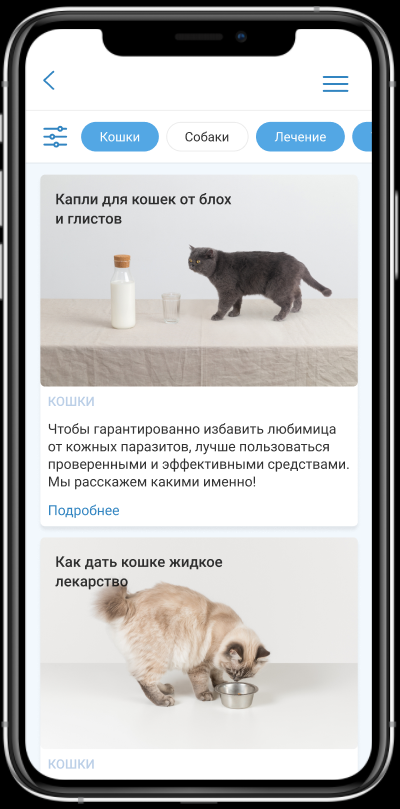


Рисунок 5.16 – Экран со списком статей

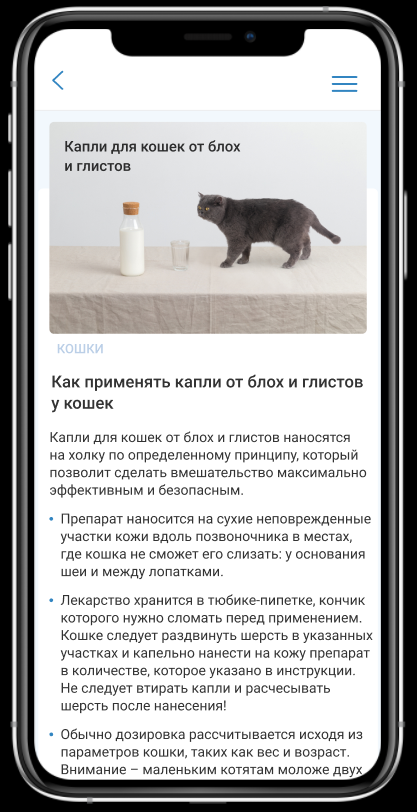


Рисунок 5.17 – Экран статьи

Описание всех элементов интерфейса описаны в приложение Ж.

вывод

# ****6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «MYPET»****

## ****6.1 Описание функций, назначения и потенциальных пользователей МП****

Мобильное приложение разрабатывается от имени ООО "ФордэКонсалтинг".

Мобильное приложение позволяет выполнять следующие функции:

* возможность авторизации через социальные сети;
* определение местоположения пользователя;
* поиск необходимой информации по уходу за питомцем в приложении;
* поиск ветеринарных клиник на картах;
* запись на прием к ветеринару;
* фильтрация и отображение интересующих статей по уходу за питомцем;
* поиск зоомагазинов на картах;
* поиск выставок;
* запись на выставки на определенную дату.

Разработка программного средства была осуществлена по индивидуальному заказу.

## ****6.2 Расчет затрат на разработку ПС****

Затраты на основную заработную плату команды разработчиков определены исходя из состава и численности команды, размеров месячной заработной платы каждого из участников команды, а также общей трудоемкости разработки программного обеспечения.

Расчет величины основной заработной платы участников команды осуществляется по формуле

где n – количество исполнителей, занятых разработкой конкретного ПС;

ЗЧi – часовая заработная плата i-го исполнителя (руб.);

ti – трудоемкость работ, выполняемых i-м исполнителем (ч).

Данные по заработной плате предоставлены ООО "ФордэКонсалтинг" на 30.11.2020 г.

Расчетная норма рабочего времени принята равной 168 часам.

Расчет основной заработной платы представлен в таблице 6.1.

Таблица 6.1 **–** Расчет основной заработной платы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Участник команды | Месячная заработная плата, руб. | Часовая заработная плата, руб. | Трудоемкость работ, часов | Основная заработная плата, руб. |
| 1 | Бизнес-аналитик | 2826 | 16,82 | 24 | 403,68 |
| 2 | DevOps инженер | 3854 | 22,94 | 16 | 367,04 |
| 3 | UI/UX-дизайнер | 2569 | 15,29 | 40 | 611,6 |
| 4 | Инженер-программист | 4111 | 24,47 | 112 | 2740,64 |
| 5 | Тестировщик | 2312 | 13,76 | 24 | 330,24 |
| ПРЕМИЯ, 40% | | | | | 1781,28 |
| Итого затраты на основную заработную плату, руб. | | | | | 6234,48 |

Затраты на дополнительную заработную плату команды разработчиков включают выплаты, предусмотренные законодательством о труде, и определяется по формуле

где Зо – затраты на основную заработную плату команде разработчиков с учетом премии (руб.);

Нд – норматив дополнительной заработной платы команды разработчиков (10%).

Дополнительная заработная плата составит:

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле

где – норматив отчислений в фонд социальной защиты населения и на обязательное страхование (34,6%).

Расчет прочих затрат осуществляется в процентах от затрат на основную заработную плату команды разработчиков с учетом премии по формуле

где Нпз – норматив прочих затрат (120%).

Прочие затраты составят:

Общая сумма затрат на разработку программного средства находится путем суммирования всех статей затрат. Ее расчет представлен в таблице 6.2.

Таблица 6.2 – Затраты на разработку программного обеспечения

|  |  |
| --- | --- |
| Статья затрат | Сумма, руб. |
| Основная заработная плата команды разработчиков | 6234,48 |
| Дополнительная заработная плата команды разработчиков |  |
| Отчисления на социальные нужды | 2372,84 |
| Прочие затраты | 7481,4 |
| Общая сумма затрат на разработку | 16 712,17 |

## ****6.3 Оценка эффекта от продажи МП****

Экономический эффект организации-разработчика представляет собой  
прибыль, полученную от разработки программного продукта под заказ  
сторонней организации. Поскольку в данном случае мобильное приложение является уникальным, т.е. создается под нужды и требования конкретного заказчика, то его цена определяется в процессе переговоров между последним и разработчиком. В качестве договорной цены установлена сумма в 25300 рублей. Стоимость аналогичных программных средств находится в диапазоне от 25000 до 35000 рублей.

Поскольку разработчик освобожден от уплаты налога на прибыль, так как резиденты ПВТ освобождены от уплаты налога на прибыль и НДС, земельного налога и налога на недвижимость, тогда расчет прибыли осуществляется по формуле:

где Ц – цена реализации ПС заказчику (руб.);

Зр – сумма расходов на разработку ПС (руб.);

П – прибыль, получаемая организацией-разработчиком от реализации данного ПС (руб.);

В соответствии с формулой (6.5) сумма прибыли составит:

Проект будет экономически эффективным, если рентабельность затрат на разработку программного средства будет не меньше средней процентной ставки по банковским депозитным вкладам. Рентабельность затрат на разработку ПС рассчитывается по формуле

Р = П / \* 100% (6.6)

где П – прибыль, получаемая организацией-разработчиком от  
реализации данного ПС (руб.);

– общая сумма затрат на разработку ПС (руб.);

Рентабельность составит:

Р = 16712,17/ \* 100% = 51 (%)

По результатам проведенных расчетов затраты на разработку составляют рублей, при этом чистая прибыль составляет рублей, а рентабельность составляет 51%

# ****ЗАКЛЮЧЕНИЕ****

В рамках дипломного проекта было необходимо создать мобильное приложение «MyPet» под операционную систему iOS, которое предоставляет пользователю информацию о местоположении ветеринарных клиник, зоомагазинов, выставок и чтение статей по уходу за питомцем. Приложение позволяет пользователю записываться на прием в ветеринарную клинику, просматривать контакты, фотографии и узнавать более подробную информацию о выбранном месте. Также для удобства пользователя были созданы фильтры для выбора необходимой информации, что значительно упрощает поиск необходимой информации.

Для достижения поставленной задачи был осуществлен обзор существующих аналогов на платформе AppStore, который показал, что главным недостатком приложений является ограниченный бесплатный функционал.

Изучена предметная область, рассмотрены, описаны и выбраны среда разработки, язык разработки мобильного приложения.

Был разработан понятный и дружественный интерфейс приложения, который позволяет быстро ориентироваться в приложении, тем самым не затрудняя использование МП пользователя.

Проведено тестирование программного продукта, в рамках которого были выявлено и исправлено некорректное поведение системы. Разработаны тест-кейсы, в рамках которых производится тестирование мобильного приложения перед каждым его релизом.

Была разработана информационная модель проектируемой системы для описания параметров и переменных величины объекта, связи между ними, входы и выходы объекта. Разработана функциональная модель, которая отражает формализацию и описывает бизнес-процессы МП. Также была создана диаграмма вариантов использования, которая отражает отношения между актерами и прецедентами, и позволяет описать систему на концептуальном уровне. Был разработан алгоритм работы мобильного приложения.

Уделено внимание технико-экономическому обоснованию. Разработано руководство пользователя с подробным описанием использования программного продукта.

Разработано мобильное приложение «MyPet» для операционной системы iOS в соответствии с предъявляемыми требованиями к данному программному продукту, включая требования сертификации в Apple.

В рамках дипломного проекта были углублены знания по разработке мобильных приложений, использованию сервисов по аналитике сторонних приложений, существующих платформ, а также по проектированию пользовательского интерфейса.

# ****СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ****

1. Дипломные проекты (работы) общие требования СТП 01–2010 [Электронный ресурс]: стандарт предприятия / БГУИР – Электронные данные. – Режим доступа: СТП П2010 бгуир.pdf. Дата доступа: 25.11.2020.
2. Официальный сайт. Статья «Безопасность платформы Apple» [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://support.apple.com/ru-ru/guide/security/sece3bee0835/1/web/1>. Дата доступа: 25.11.2020.
3. Официальный сайт. Статья «Обзор Xcode: о Xcode» [Электронный ресурс]. Электронные данные. – Режим доступа: <http://spec-zone.ru/RU/XCode/documentation/ToolsLanguages/Conceptual/Xcode_Overview/index.html>. Дата доступа: 25.11.2020.
4. Официальный сайт. Swiftbook. Руководство Swift [Электронный ресурс]. Электронные данные. – Режим доступа: <https://swiftbook.ru/content/languageguide/basics/>. Дата доступа: 25.11.2020
5. Официальный сайт. Статья «Swift. Язык программирования с открытым кодом» [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.apple.com/ru/swift/>. Дата доступа: 25.11.2020.
6. Официальный сайт. Статья «MongoDB Realm Has Arrived» [Электронный ресурс].–Электронные данные. – Режим доступа: https://realm.io/. Дата доступа: 25.11.2020.
7. Буч, Г. Введение в UML от создателей языка / Гради Буч. -

М.: ДМК Пресс, 2010. – 693 c.

1. Вернон, В. Реализация методов предметно-ориентированного проектирования / Вон Вернон. - М.: Вильямс, 2015. - 688 c.
2. Мандел, Т. Разработка пользовательского интерфейса. / Т. Мандел. Издательство: ДМК Пресс, 2008. – 412 с.
3. Рэшка, Д. Тестирование программного обеспечения / Д. Рэшка, Д. Пол, Д. Элфрид – М.: Лори, 2014. – 568 с.
4. Сорока, О.Г. Официальный сайт учреждения образования / О.Г.

Сорока, И.Н. Васильева // Пачатковае навучанне: сям'я, дзіцячы сад, школа: Университет педагогического самообразования. – 2014. – № 10. – С. 1 – 8.

1. Тамре, Л. Тестирование программного обеспечения / Л. Тамре – М.: Вильямс, 2003. – 368с.
2. Усов, В. Swift. Основы разработки приложений под iOS, iPadOS и macOS. – 3-е изд. – СПб.: Питер, 2017. – 368 с.
3. Хелм, Р. Приемы объектно-ориентированного проектирования.

Паттерны проектирования / Р. Хелм. – Питер, 2020. – 368 с.

1. Холл, Г. М. Адаптивный код: гибкое кодирование с помощью

паттернов проектирования и принципов SOLID / Г.М. Холл – М.: Вильямс, 2019. – 448 с.

# ПРИЛОЖЕНИЕ А

**(**обязательное**)**

Исходный текст программы