**2. Исследовательская часть**

# **2.1. Постановка задачи проектирования**

Целью написания данной курсовой работы является разработка web-сервиса «HandyMady», для облегчения рабочего процесса самозанятых лиц, а также реализации товаров и услуг при помощи поиска и рекомендательной системы.

Задача проектирования такой системы может быть решена при помощи следующих средств разработки: средством реализации пользовательских интерфейсов должно быть реализовано на языках HTML и CSS, системы управления базами данных PostgreSQL, Исходные коды должны быть реализованы на языке Python посредством свободного фреймворка для веб-приложений Django.

Структурное проектирование системы осуществляется с помощью составления концептуальной и логической моделей хранимых данных. Описание этапов проектирования архитектуры приложения сопровождается описанием на естественном языке.

# **2.2. Описание предметной области.**

Самозанятость — форма получения вознаграждения за свой труд непосредственно от заказчиков, в отличие от наёмной работы.

Самозанятый — это лицо, у которого нет работодателя и у которого нет наемных работников, то есть это граждане, получающие доход от своей личной трудовой деятельности.

Стать самозанятым могут обычные граждане и индивидуальные предприниматели, которые решили поменять статус. Физические лица и индивидуальные предприниматели, которые перейдут на новый специальный налоговый режим (самозанятые), могут платить с доходов от самостоятельной деятельности только налог по льготной ставке — 4 (при получении денег от физических лиц) или 6% (при получении денег от юридических лиц).

Осуществлять свою рабочую деятельность самозанятые лица могут с помощью следующих интернет площадок:

* Социальные сети — онлайн-платформа, которую люди используют для общения, создания социальных отношений с другими людьми, которые имеют схожие интересы или офлайн-связи. На базе социальных сетей самозанятое лицо создает сообщество с целью продвижения товара и его дальнейшего сбыта. Самозанятое лицо, продавец, ведет каталоги, добавляя новый товар в наличие, ведет переписки с потенциальными покупателями, работает над продвижением своего товара. Оплата в социальных сетях чаще всего производится посредством онлайн переводов при помощи сторонних приложений банков. Это существенно затрудняет ведение финансовой отчетности и порождает проблему мошенничества.
* Интернет-магазин — сайт, торгующий товарами посредством сети Интернет. Позволяет пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ. При этом продажа товаров осуществляется дистанционным способом, и она накладывает ограничения на продаваемые товары. В интернет-магазинах оплата осуществляется при помощи посредников.

Необходимо разработать интернет-магазин, реализующий товары и услуги самозанятых. Система должна обеспечить удобный интерфейс продавцам для создания своего магазина с возможностью добавления товара в каталог и ведение акций, в то время как пользователь получит удобный интерфейс поиска товаров по категориям и рекомендациям.

# **2.2.1 Концептуальная схема данных.**

На основании представленной информации в предметной области можно выделить несколько сущностей (рисунок 1): Пользователь, магазин, товар, акция, заказ и позиция заказа.

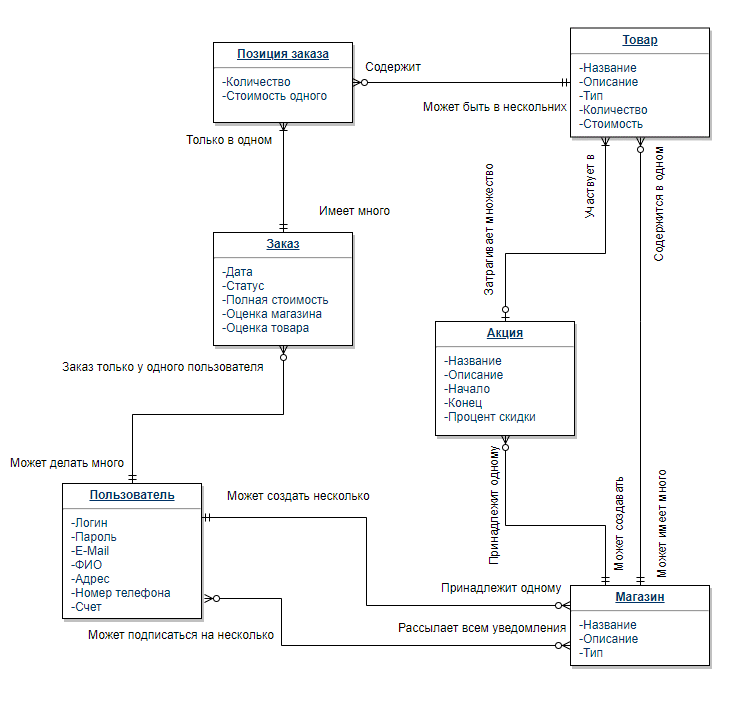


Рисунок 1 – Концептуальная модель данных

На рисунке 1 представлена концептуальная схема данных предметной области. Из данной схемы можно выделить следующие связи: существуют пользователи, которые могут делать много заказов, подписываться на чужие магазины и создавать множество своих. Владельцы магазинов могут создавать акции и добавлять в ассортимент новые товары. В одном заказе может быть множество позиций заказов, содержащие в себе товар и его количество.

# **2.2.2 Сценарии использования.**

После регистрации на сайте пользователь может как делать заказы в магазинах других продавцов, так и создать свои магазины с различными товарами и продавать их. У пользователя есть возможность просмотра каталога товаров по различным категориям. Пользователь как в роли покупателя, так и в роли продавца может создать несколько магазинов с различными категориями товаров. У каждого товара есть свое описание, кратко рассказывающее о том, что представляет из себя продукт, и фотографии, при помощи которых можно построить себе визуальное представление о товаре. Чтобы сделать заказ покупатель должен добавить товары в корзину, указать номер телефона и телефон, если они не написаны в профиле и оплатить заказ. После того как товар был получен покупатель получает право оценить по пятибалльной шкале товар, продавца и магазин. Также пользователь может воспользоваться поиском товаров с различными фильтрами для точности. Для поиска товаров будет реализована поисковая строка. Поиск может проводиться как по названию, так и по различным категориям товаров. Покупатель может зайти на страницу магазина других пользователей, где может подписаться на рассылку новостей о появлении новый товаров и акций. UML диаграмма пользовательских сценариев (рисунок 2) показывает возможные варианты действий покупателя.



Рисунок 2 – UML диаграмма пользовательских сценариев для покупателя

Продавец имеет такие же пользовательские сценарии что и покупатель, но при этом он может заниматься менеджментом своих магазинов. Он может проводить различные акции индивидуально для каждого своего магазина с настройкой описания, даты начала и окончания. В профиле магазина владелец может видоизменить его вид, добавить товары на продажу и купить рекламу для продвижения товаров. UML диаграмма пользовательских сценариев (рисунок 3) показывает возможные варианты действий продавца.

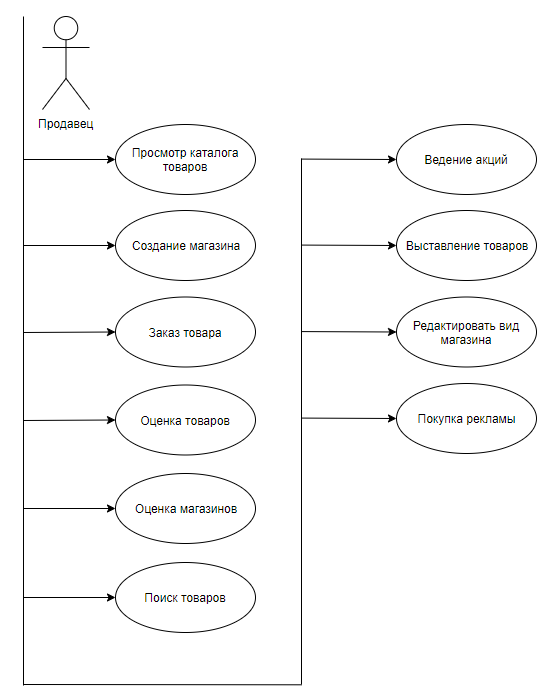


Рисунок 3 – UML диаграмма пользовательских сценариев для продавца

# **2.2.3 Алгоритм системы рейтинга и рекомендаций**

На главной странице система рекомендации подбирает пользователю на основе его предыдущих покупок.

Рейтинг составляется из выставленной оценке по пятибалльной шкале пользователями. Итоговая оценка формируется из суммы всех оценок поделенного на их количество.

# **2.3. Перечень задач, подлежащих решению в процессе разработки**

Для создания системы необходимо:

* Составить техническое задание.
* Провести исследование предметной области;
* Сформировать концептуальную и логическую модели хранимых данных;
* Подготовить перечень необходимого функционала;
* Оформить описание процессов, протекающих в системе;
* Спроектировать макет сайта;
* Организовать структуру хранения данных и реализовать тестовое наполнение базы данных,
* Реализовать систему:
* Произвести тестирование и отладку;
* Оформить конструкторскую документацию;

Перечень реализуемых функций:

* Реализация поиска товаров по названию, категориям, цене, оценкам и авторам.
* Реализация ведения списков желаемого.
* Система рейтинга, базирующаяся на отзывах пользователей.
* Предоставлять пользователю информацию об выбранном товаре.
* Функция подписок на магазины с возможностью уведомления о поступлении новых товарах и акциях.
* Возможность уведомления о скидках на товары из списка желаемого.

# **2.4. Обоснование выбора инструментов и платформы для разработки**

PostgreSQL - это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных.

* Надежность PostgreSQL.
* Производительность PostgreSQL основывается на использовании индексов, интеллектуальном планировщике запросов, тонкой системы блокировок, системе управления буферами памяти и кэширования, превосходной масштабируемости при конкурентной работе.
* Поддержка SQL
* PostgreSQL имеет очень богатый набор встроенных функций и операторов для работы с данными, полный список которых можно посмотреть в документации.
* Простота использования всегда являлась важным фактором для разработчиков.
* pgAdmin (GNU Artistic license) предоставляет удобный интерфейс для работы с базами данных PostgreSQL.

HTML (HyperText Markup Language — язык гипертекстовой разметки) — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML до 5-й версии определялся как приложение SGML (стандартного обобщённого языка разметки по стандарту ISO 8879). Спецификации HTML5 формулируются в терминах DOM (объектной модели документа).

Язык XHTML является более строгим вариантом HTML, он следует синтаксису XML и является приложением языка XML в области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

CSS (Cascading Style Sheets) — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

CSS поддерживает таблицы стилей для конкретных носителей, поэтому авторы могут адаптировать представление своих документов к визуальным браузерам, слуховым устройствам, принтерам, брайлевским устройствам, карманным устройствам и т.д.

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

Django — это программный каркас с богатыми возможностями, подходящий для разработки сложных сайтов и веб-приложений, написанный на языке программирования Python.

Django — фреймворк для веб-приложений на языке Python. Один из основных принципов фреймворка — DRY (don't repeat yourself). Веб-системы на Django строятся из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из заметных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Также, в отличие от многих других фреймворков, обработчики URL в Django конфигурируются явно (при помощи регулярных выражений), а не автоматически задаются из структуры контроллеров.

Django проектировался для работы под управлением Apache (с модулем mod\_python) и с использованием PostgreSQL в качестве базы данных. В настоящее время, помимо PostgreSQL, Django может работать с другими СУБД: MySQL (MariaDB), SQLite, Microsoft SQL Server, DB2, Firebird, SQL Anywhere и Oracle. Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Первоначально разработка Django велась для обеспечения более удобной работы с новостными ресурсами, что достаточно сильно отразилось на архитектуре: фреймворк предоставляет ряд средств, которые помогают в быстрой разработке веб-сайтов информационного характера. Например, разработчику не требуется создавать контроллеры и страницы для административной части сайта, в Django есть встроенное приложение для управления содержимым, которое можно включить в любой сайт, сделанный на Django, и которое может управлять сразу несколькими сайтами на одном сервере. Административное приложение позволяет создавать, изменять и удалять любые объекты наполнения сайта, протоколируя все совершённые действия, и предоставляет интерфейс для управления пользователями и группами (с пообъектным назначением прав).