МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра математического и компьютерного моделирования

Разработка платформы для продажи товаров и услуг

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 247 группы

направление 09.04.03 – Прикладная информатика

механико-математического факультета

Косова Сергея Григорьевича

Научный руководитель
проф., д.э.н., проф. Л.В. Кальянов

Зав. кафедрой
зав.каф., д. ф. – м. н., доцент Ю.А. Блинков

Саратов 2019

Введение. Интернет-торговля является следствием развития электронной коммерции. Она включает в себя:

- Передачу информации, продуктов или услуг через онлайн ресурсы;
- Предоставление услуг электронным путем;
- Организация онлайн методами обычной торговли.

Первым примером продаж товара через интернет можно считать систему SABRE (Semi-Automatic Business Research Environment). В 1960 году американские компании American Airlines и IBM приступают к созданию системы автоматизации процедуры резервирования мест на авиарейсы. Система SABRE сделала воздушные перелёты более доступными для рядовых пассажиров, помогая им ориентироваться в тарифах и рейсах, число которых постоянно растёт. За счет автоматизации процесса расчета тарифов, при резервировании мест, снизилась стоимость услуг, выросли пассажироперевозки.

Развитие интернет-торговли в России началось в 2007 году и привело к созданию таких площадок как Авито, Юла и прочие.

Для любого бизнеса важно видеть динамику продаж. Это позволяет выявлять слабые места, а также упрощает планирование. Но существующие площадки сильно ограничивают возможности своих клиентов по анализу продаж и предоставляют довольно скудный функционал по интеграции со сторонними системами.

Целью работы является создание платформы нового поколения, которая упростит клиентам поиск нужного товара и позволит выбрать наиболее выгодное предложение, а владельцы компаний получат удобный сервис для продажи своих товаров, помогающий им автоматизировать часть бизнес-процесса.

Данная платформа позволит избавить малый и средний бизнес от этих проблем, а также предоставит функционал по анализу продаж, предсказанию оттока клиентов и новые пути коммуникации с клиентами.

Все это позволит автоматизировать часть бизнес процессов малого и среднего бизнеса, что в свою очередь приведет к увеличению качества и количества предоставленных услуг.

Работа состоит из трех разделов. Первый раздел посвящен анализу функционала существующих решений и выявлению недостающих функциональ-

ных возможностей. Во втором разделе описаны этапы проектирования и разработки основных функций. Третий раздел будет посвящает описанию возможностей разработанной платформы. Здесь рассмотрены функциональные возможности платформы, способы использования, а также произведено сравнение с аналогами.

Анализ функциональных требований к платформе. Функциональные требования к платформе были сформированы опираясь на функционал существующих решений, а также на выявленные проблемы малого и среднего бизнеса.

Лидерами на российском рынке в данной области являются Авито и ЮЛА. Они предоставляют возможности по размещению товаров, созданию магазина и ведению сильно ограниченной статистики.

Общаясь с представителями малого и среднего бизнеса, была проведена работа по моделированию бизнес процессов, а также были выявлены места, нуждающиеся в оптимизации и автоматизации.

Из представителей среднего бизнеса получилось установить контакт с сотрудниками клуба робототехники и программирования «Код да винтик». Они рассказали о предоставляемых услугах и их жизненных циклах, показали имеющиеся процессы наблюдения за бизнес процессами, а также показали как организована клиентская база. Малый бизнес представлен так называемыми самозанятыми. Бизнес процессы почти у всех одинаковые и спектр услуг невелик, поэтому малый бизнес не будет рассматриваться на конкретных его представителях. В ходе анализа бизнес-процессов были выявлены следующие проблемы:

- Нет статистики.
- Нет истории.
- Нет средств коммуникации с клиентом кроме звонков.
- Нет возможности заказать услугу online.
- сложно оставить отзыв.

Проектирование и разработка. В ходе анализа нефункциональных требований к платформе было принято решение использовать микросервисную архитектуру [1].

Этот тип архитектуры позволяет масштабировать приложения по оси Y «Куба масштабирования» (Scale Cube), описанного в книге «The Art of Scalability» Мартина Эбботта (Martin L. Abbott) и Майкла Фишера (Michael T. Fisher).

В этом случае приложение разбивается на множество небольших сервисов, называемых микросервисами. Каждый микросервис включает в себя бизнес-логику и представляет собой совершенно независимый компонент. Сервисы одной системы могут быть написаны на различных языках программирования и общаться друг с другом, используя различные протоколы [2].

Микросервисная архитектура позволяет с легкостью масштабировать приложение – если вам понадобилось внедрить новую функцию, то достаточно просто написать новый сервис.

Очевидным недостатком этого паттерна является необходимость передачи большого количества данных между микросервисами. Если накладные расходы на обмен сообщениями слишком велики, то нужно либо оптимизировать протокол, либо объединить микросервисы.

Далее функционал платформы был декомпозирован на сервисы [3]

- Account Сервис аккаунт реализует логику по сохранению деятельности пользователя в системе – свойства самого пользователя, настройки интерфейса платформы, результаты действий в системе.
- Communacation. Сервис коммуникации является единой точкой входа в интерфейс обмена сообщениями. Основная его задача - сформировать из данных введенных пользователем сообщение в формате системы и положить его в очередь сообщений.
- vk-sender. Этот сервис представляет собой клиент для отправки сообщений в социальной сети ВКонтакте, также это один из способов отправки сообщений внутри платформы.
- internal-sender. Данный сервис представляет собой мессенджер, который работает только внутри платформы. Он отвечает за доставку сообщений пользователей без использования сторонних сервисов. Такой

вариант может быть удобен для пользователей, которые не хотят связывать свой аккаунт в социальной сети со своими покупками.

- market. Этот сервис обладает следующими обязанностями:
  - предоставление информации о товаре (цена, доступность, др характеристики),
  - категоризация товаров,
  - разнообразная фильтрация товаров.
  - осуществление операций над товаром (оформление, отмена, завершение заказа)
- companion-service. Этот сервис осуществляет сбор статистики о товарах, которые были просмотрены в рамках одной сессии, производит агрегацию и выдает статистическую информацию. Основываясь на этой информации система предлагает несколько товаров к товару который сейчас смотрит пользователь.
- prediction-service. Сервис предсказания уменьшения покупательской активности для конкретного товара.

  Основываясь на статистике совершенных сделок, учитывая отмены заказов, опираясь на характеристики товаров, система прогнозирует потерю интереса покупателей к товару.
- auth-service Сервис авторизации занимается авторизацией пользователей и разграничением прав доступа к функционалу.
- CONFIG Конфигурация всех сервисов должна быть в одном месте. Будет крайне неудобно иметь 10 сервисов, каждый из которых нужно настраивать отдельно. Оптимальным будет вариант, когда каждый сервис знает всего одну точку, придя на которую он получает всю конфигурацию, с которой может полноценно функционировать.
- API GATEWAY Все основные сервисы, которые были описаны выше, предоставляют для внешнего пользователя некоторый API.

  Клиентское приложение может запрашивать каждый из сервисов самостоятельно. Но такой подход имеет массу ограничений — необходимость знать адрес каждого эндпоинта, делать запрос за каждым куском информации отдельно и самостоятельно производить слияние результа-

тов. Кроме того, не все приложения на бэкенде могут поддерживать дружественные web протоколы и т.д..

- Service Discovery Service discovery позволяет автоматически определять сетевые адреса для доступных экземпляров приложений, которые могут динамически изменяться по причинам масштабирования, падений и обновлений.
- monitoring Этот сервис инфраструктурный, необходим для изоляции точек доступа к удаленным системам, службам и сторонним библиотекам, предотвращения каскадного сбоя и обеспечения устойчивости в сложных распределенных системах, где сбой неизбежны.

Все сервисы базируются на платформе Java, в качестве языков разработки использовались Java, Scala, Kotlin и Groovy. Для хранения данных использовались два вида баз данных – реляционная [4] (PostgreSQL [5–7]) и документоориентированная (Mongodb и Elasticsearch [8])

Для реализации машинного обучения был использован Spark, для анализа данных ELK стек [9].

Полученное решение. Возможности и способы использования. Архитектура полученного решения схематично визуализирована в соответствии с рисунком 1.

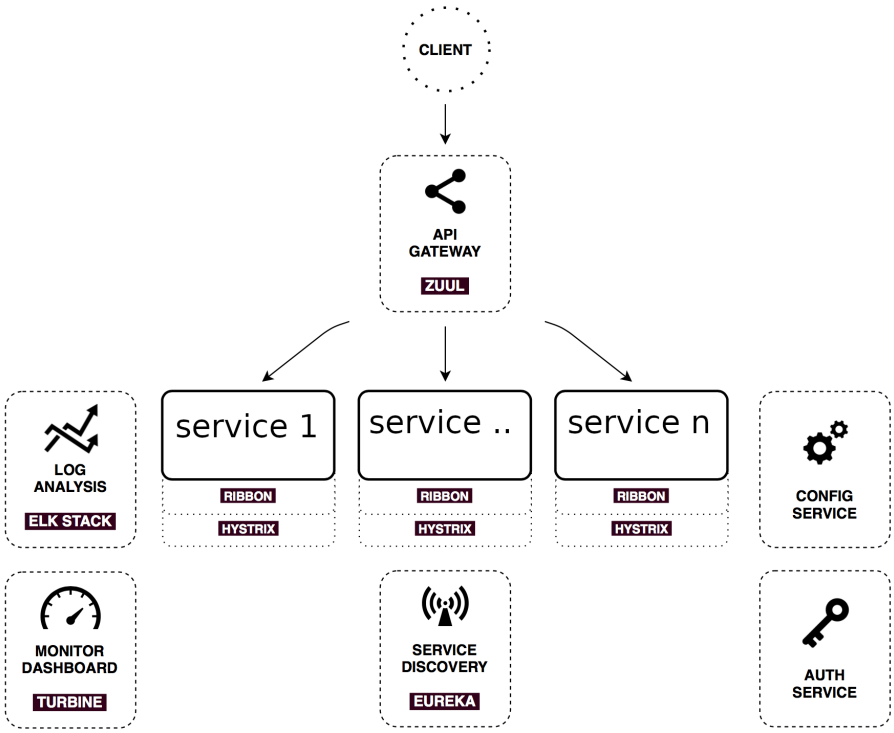

Рисунок 1 — Схематичное представление архитектуры ИС на уровне сервисов

Как видно из схемы, ИС имеет единую точку входа (API Gateway), набор сервисов (service 1..n), а также набор вспомогательных компонент, которые призваны обеспечить стабильную работу системы.

Система состоит из четырнадцати сервисов и восьми сторонних компонент, для десяти из четырнадцати сервисов необходима база данных.

Платформа обладает всеми заявленными функциональными требованиями. Для продажи товаров реализован минималистичный интерфейс пользователя. Для анализа продаж – легко настраиваемые визуализации агрегаций. Обученная модель для анализа оттока клиентов выдает результаты с точностью 80-85%.

Платформа может быть использована как отдельной компанией, желающей построить на ее основе свой бизнес, так и несколькими компаниями, которые регистрируют в платформе свои магазины независимо друг от друга, но при этом пользуются всем функционалом платформы. Благодаря архитектуре платформы она может выдержать поток пользователей сопоставимый с популярными интернет площадками такими как авито или юла.

Заключение. Цель работы – создание платформы, которая упростит клиентам поиск нужного товара и позволит выбрать наиболее выгодное предложение, а владельцы компаний получат удобный сервис для продажи своих товаров, помогающий им автоматизировать часть бизнес – достигнута.

Был произведен анализ предметной области, затем анализ потребностей малого и среднего бизнеса. Далее был произведен анализ существующих решений. Основываясь на этих данных была реализована платформа, позволяющая автоматизировать бизнес-процесса малого и среднего бизнеса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Yang, H. Software Reuse in the Emerging Cloud Computing Era / Yang, H. – M. : Information Science Reference, 2012. – 54 с.

2 Немного об архитектурах программного обеспечения. [Электронный ресурс] : [Сайт]. URL: https://habrahabr.ru/company/it-grad/blog/276297/ (дата обращения: 23.02.2018). – Загл. с экрана. – Яз. рус.

3 Фаулер, Мартин. Архитектура корпоративных программных приложений / Фаулер, Мартин. – М : New York TODS, 2006. – 544 с.

4 Кодд, Е.Ф. Реляционная модель данных для больших совместно используемых банков данных / Кодд, Е.Ф. – М. : Москва, 2007. – 344 с.

5 Филипп Дельгядо — СУБД: индивидуальный пошив и подгонка по фигуре. [Электронный ресурс] : [Видео]. URL: https://www.youtube.com/watch?v=l4l5pLlC40U (дата обращения: 13.02.2018). – Загл. с экрана. – Яз. ru.

6 PostgreSQL: Documentation: 9.5: JSON Functions and Operators. [Электронный ресурс] : [Сайт]. URL: https://www.postgresql.org/docs/9.5/functions-json.html (дата обращения: 12.02.2018). – Загл. с экрана. – Яз. en.

7 Postgres vs Mongo / Олег Бартунов (Postgres Professional). [Электронный ресурс] : [Видео]. URL: https://www.youtube.com/watch?v=SNzOZKvFZ68 (дата обращения: 14.02.2018). – Загл. с экрана. – Яз. ru.

8 MongoDB vs. Elasticsearch: The Quest of the Holy Performances. [Электронный ресурс] : [Сайт]. URL: https://blog.quarkslab.com/mongodb-vs-elasticsearch-the-quest-of-the-holy-performances.html (дата обращения: 14.02.2019). – Загл. с экрана. – Яз. en.

9 How to monitor Elasticsearch performance. [Электронный ресурс] : [Сайт]. URL: https://www.datadoghq.com/blog/monitor-elasticsearch-performance-metrics (дата обращения: 1.04.2018). – Загл. с экрана. – Яз. en.