Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Факультет инновационных технологий (ФИТ)
Кафедра инновационного обеспечения инновационной деятельности (ИОИД)

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП, зав. каф. ИОИД,
канд. физ.-мат. наук, ст. н. с.
_____ С. Л. Миньков
«22» 06 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Создание CRM-системы для малого бизнеса

по основной образовательной программе подготовки бакалавров
направление подготовки 09.03.03 – Прикладная информатика

Руководитель ВКР
Зав. кафедрой ИОИД
_____ С. Л. Миньков
«22» 06 2020 г.

Автор работы
студент группы №18511
_____ А. В. Пупышев
«___»_____2020 г.

Томск-2020

Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Факультет инновационных технологий

Кафедра информационного обеспечения инновационной деятельности

УТВЕРЖДАЮ
Руководитель ООП, зав. каф. ИОИД,
канд. физ.-мат. наук, ст. н. с.
_____С.Л. Миньков
«___» _____2020 г.

ЗАДАНИЕ

по подготовке выпускной квалификационной работы студенту группы № 18511

Пупышеву Александру Викторовичу

1. Тема ВКР: «Создание CRM-системы для малого бизнеса» (утверждена распоряжением по факультету от 20 мая 2020 г. №252/20).

2. Срок сдачи студентом законченной ВКР:

а) на кафедре 19.06.2020 г.

б) в ГЭК 23.06.2020 г.

3. Исходные данные к ВКР: отчет по научно-исследовательской работе, отчет по преддипломной практике, документация к средствам разработки.

4. Краткое содержание работы: описание предметной области, анализ предметной области, выбор средств разработки, проектирование информационной системы, описание информационной системы.

5. Перечень графического материала: описание проблем, описание аналогов, постановка цели работы, логическая модель базы данных, описание средств разработки, описание интерфейса программы, выводы по работе.

6. Дата выдачи задания: 16 февраля 2020 г.

Руководитель ВКР,
зав. кафедрой ИОИД _____ С. Л. Миньков

Задание принял к исполнению _____А. В. Пупышев

РЕФЕРАТ

ВКР бакалавра содержит: 40 с., 4 таблицы, 32 рис., 10 литературных источников.

Список ключевых слов: БИЗНЕС, CRM-СИСТЕМА, УЧЕТ, БАЗА ДАННЫХ, MYSQL, LINUX, PHP, REACT, DOCKER.

Объектом исследования данной работы является процесс ведения бизнеса с использованием современных информационных систем.

Цель работы – разработка CRM системы управления малым бизнесом и системы бронирований, ведения статистики и унифицирования входной и выходной информации.

Результатом данной выпускной квалификационной работы является программный продукт, позволяющий реализовывать поставленные цели.

Текст ВКР выполнен в текстовом редакторе MS Word 2016. Также использованы программы MS Excel. В качестве графического материала представлена презентация, выполненная в MS PowerPoint 2016.

ABSTRACT

The final Bachelor’s qualifying work contains 40 pages, 32 figures, 4 table, 10 references.

The list of keywords: BUSINESS, CRM-SYSTEM, ACCOUNTING, DATABASE, MYSQL, LINUX, PHP, REACT, DOCKER.

The object of this study is the process of doing business using modern information systems.

The purpose of the work is the development of a CRM system for managing a small business and a reservation system, maintaining statistics and unifying input and output information.

The result of this final qualification work is a software product that allows you to achieve your goals.

The Bachelor’s qualifying work text was made in a text editor MS Word 2016. MS Excel programs were also used. Presentation made in MS PowerPoint 2016 is presented as graphic material.

ОГЛАВЛЕНИЕ

6

ВВЕДЕНИЕ7

1 Анализ существующих систем9

1.1 Envybox9

1.2 Битрикс2410

1.3 Amocrm11

1.4 МойСклад12

1.5 Megaplan13

1.6 Бизнес.Ру14

1.7 WireCRM15

2 Проектирование17

2.1 Проектирование приложения17

2.2 Проектирование архитектуры17

3 Выбор технологий и средств разработки19

3.1 Технологии19

3.2 Средства разработки20

3.3 Проектирование базы данных23

4 Реализация27

4.1 Описание проекта27

4.2 Результат реализации28

ЗАКЛЮЧЕНИЕ39

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ40

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Кроссплатформенность – способность программного обеспечения работать более чем на одной аппаратной платформе и (или) операционной системе.

ORM (Object-RelationalMapping) – реляционная база данных в объектно-ориентированном программировании.

API (Application Programming Interface) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой..

Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

БД – база данных.

ИС – информационная система.

СУБД – система управления базами данных.

Web-приложение – клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

Администратор – пользователь CRM системы.

Гость – лицо, бронирующее место или стол в CRM.

ВВЕДЕНИЕ

Развитие инновационных технологий и информатизация общества напрямую влияет на все сферы любого бизнеса даже на тот, который на первый взгляд никак не связан с информационными технологиями. Сегодня, почти каждый предприниматель знает и активно использует электронный документооборот, сдачу налоговых деклараций онлайн, системы электронных платежей и многое другое. Государство активно внедряет все больше и больше технологий в помощь бизнесу и для упрощения жизни предпринимателей.

Но, к сожалению, не все предприниматели пользуются доступными средствами или пользуются не в полном объеме. Бухгалтера ведут бухгалтерию в тетрадях, учет ведется «на листочках» у некоторых нет единой базы клиентов. Это происходит, как правило, из-за консервативности и ошибочных суждений предпринимателей.

В России есть огромное количество малого бизнеса, которому необходимо вести учёт заказов и бронирований столов, например кафе или кальянным. Как правило, этот учёт ведется следующим образом: Администратор заведения собирает эту информацию по разным источникам, таким как: телефонные звонки и социальные сети, и ведёт запись в тетрадке, либо в электронной таблице. Несмотря на то, что этот подход в целом покрывает потребность ведения данного учёта, данный подход имеет ряд недостатков, например, имеет место быть человеческий фактор, когда администратор совершает ошибку при записи, перепутав день или время, или записав двух людей на одно место. Так же, очень серьезным недостатком является отсутствие возможности проводить анализ этих данных, для проведения успешных маркетинговых кампаний, из-за чего заведение не до конца реализовывает свой потенциал и теряет потенциальную прибыль.

Также, очень часто, малый бизнес не использует системы учёта остатков, что тоже влияет на успешность бизнеса, его эффективности и прибыльности.

Опрос владельцев различных заведений показал, что большинство тех заведений, которые не используют в своей работе системы бронирования и системы учёта хотели бы использовать подобные системы. Из основных преимуществ выделяют следующие пункты:

- удобство, относительно ручного учёта;
- экономия рабочего времени администраторов;
- возможность анализа данных о частоте бронирований в определенные временные отрезки;
- возможность анализа данных о среднем чеке в определенные временные

отрезки;

- возможность проводить информационные смс- и email-рассылки;
- возможность вести систему лояльности для постоянных клиентов.

Цель работы – разработка CRM системы, обеспечивающую возможность онлайн бронирования, учета остатков продукции, возможность проводить рекламные смс и email рассылки, сбор и аналитика данных.

Задачи:

- изучение существующих систем;
- проектирование собственной системы;
- выбор платформ и средств разработки;
- разработка приложения.

1 Анализ существующих систем

Все рассматриваемые аналоги реализованы в виде веб-приложений и использованием разных технологий и языков программирования. Распространяются по разным формам лицензии, и реализованы студиями разработчиков разного масштаба. Сравнение будет проходить по следующим критериям:

- стоимость;
- техническая поддержка пользователей;
- функциональность;
- внешний вид.

Оценивать будем CRM из списка лучших CRM для малого бизнеса в России [1].

1.1 Envybox

Данная CRM нацелена на управление продажами, поэтому основной упор сделан именно на продажи, основной функционал – это постановка задач менеджерам, виджеты онлайн чата на сайт, но система интегрируется со многими другими популярными сервисами и ip-телефонией.

Стоимость пользования системой в месяц составляет 750 рублей. Доступен пробный период в 7 дней.

Относительно технической поддержки, сервис обещает доступность 24/7, есть федеральный номер помощи и онлайн чат со специалистами технической поддержки. Но на основании договора они не несут ответственность за сбои в работе, а так же не несут материальную ответственность за убытки пользователей, вызванные сбоем в работе их системы.

Так же, от компании Envybox представлен ряд других продуктов, которые тесно связанны между собой.

Пример графического интерфейса CRM представлен на рис.1.

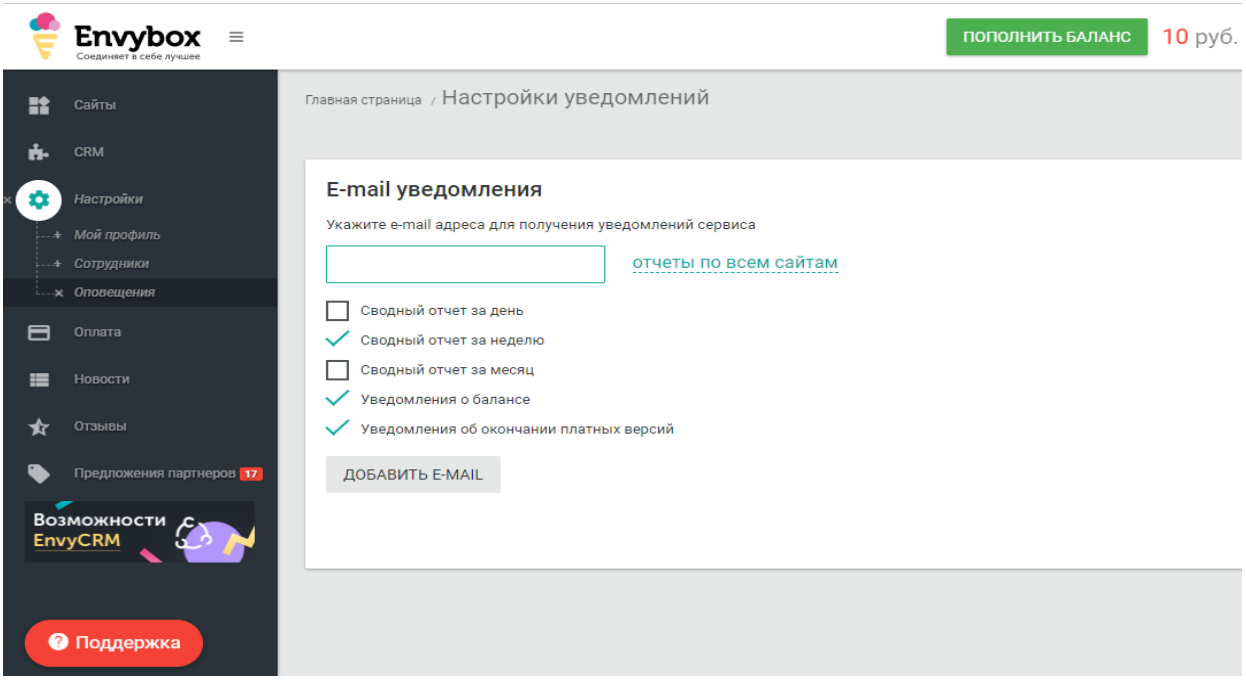


Рисунок 1 – Интерфейс «Envybox CRM»

1.2 Битрикс24

Функциональность Бутрикс24 более обширна по сравнению с Envybox, у них есть CRM-Маркетинг, что включает в себя генератор смс и email рассылок, а так же возможность управлять социальными сетями и email сообщениями через CRM. Преимуществом считается легкая интеграция с платформами 1с.

Стоимость варьируется от 990 рублей в месяц до 11 990 рублей в месяц, при покупке на год, доступна 20% скидка. Так же есть бесплатный план, для знакомства с платформой. Есть возможность купить коробочную версию, цены от 60 тысяч рублей до 1 миллиона рублей.

Техническую поддержку обещают 24/7 есть возможность стать vip клиентом и иметь отдельный, приоритетный, канал поддержки.

Внешний вид представлен на рис. 2.

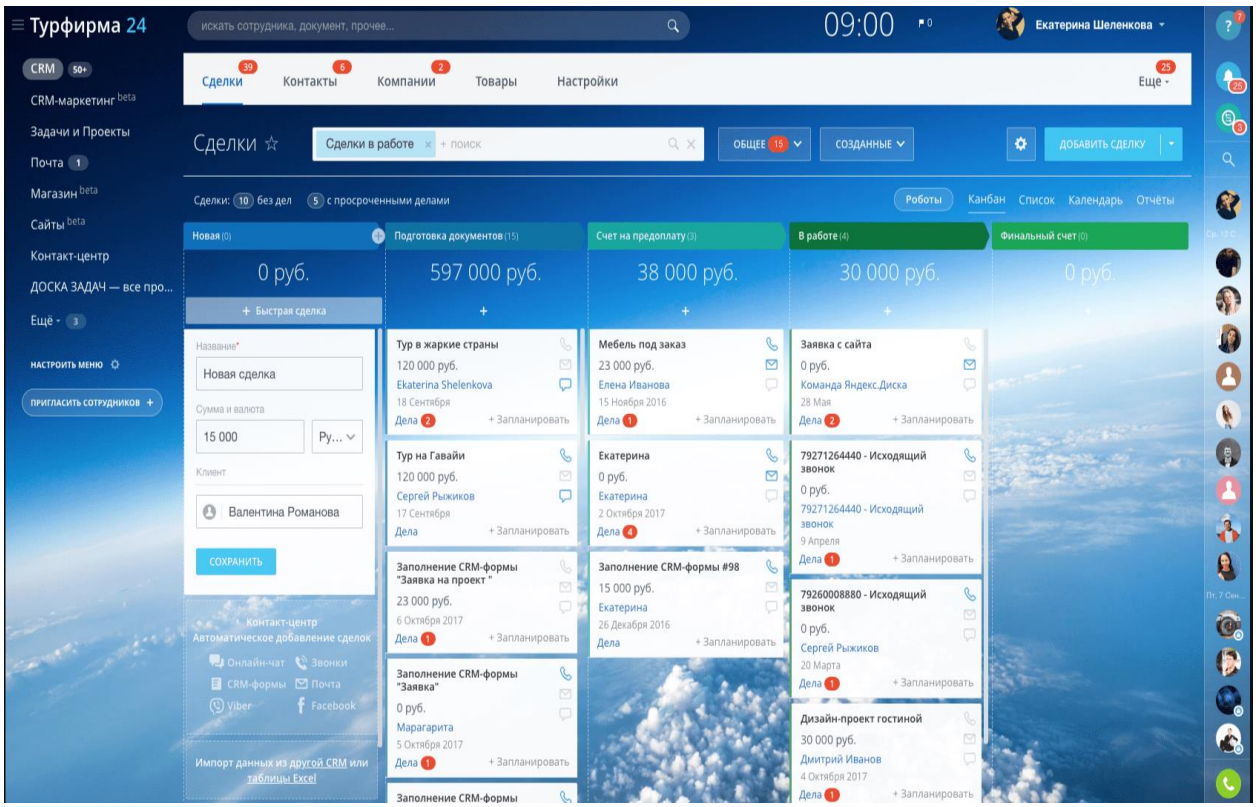


Рисунок 2 – Интерфейс «Битрикс24 CRM»

1.3 Amocrm

Amocrm имеет схожий функционал с Envybox, единственное функциональное отличие, это возможность управления переписками в социальных сетях через CRM.

Компания обещает техническую поддержку 24/7.

Стоимость пользования от 499 рублей до 1499 рублей за пользователя в месяц, есть возможность покупки пакетов на нескольких пользователей, от 4999 в год. Есть 14 дневный пробный период.

Внешний вид представлен на рис.3.

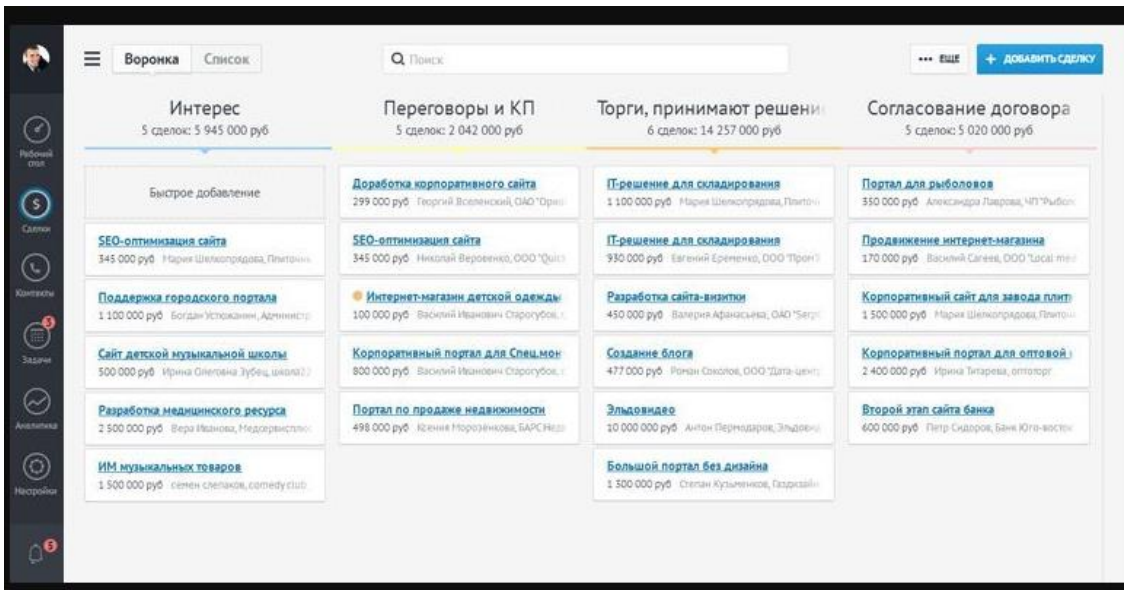


Рисунок 3 – Интерфейс «Amocrm»

1.4 МойСклад

Если предыдущие CRM были нацелены на управление продажами, то данная CRM нацелена на учет остатка товаров. Система предоставляет различный функционал связанный с учетом, формированием документации, а так же ведению бухгалтерии, но, к сожалению, никаких функций для связей с клиентами данная CRM не предоставляет.

Стоимость от 450 рублей до 6900 рублей в месяц в зависимости от количества работников и складов. Есть возможность бесплатного пробного периода. Внешний интерфейс представлен на рисунке 4.

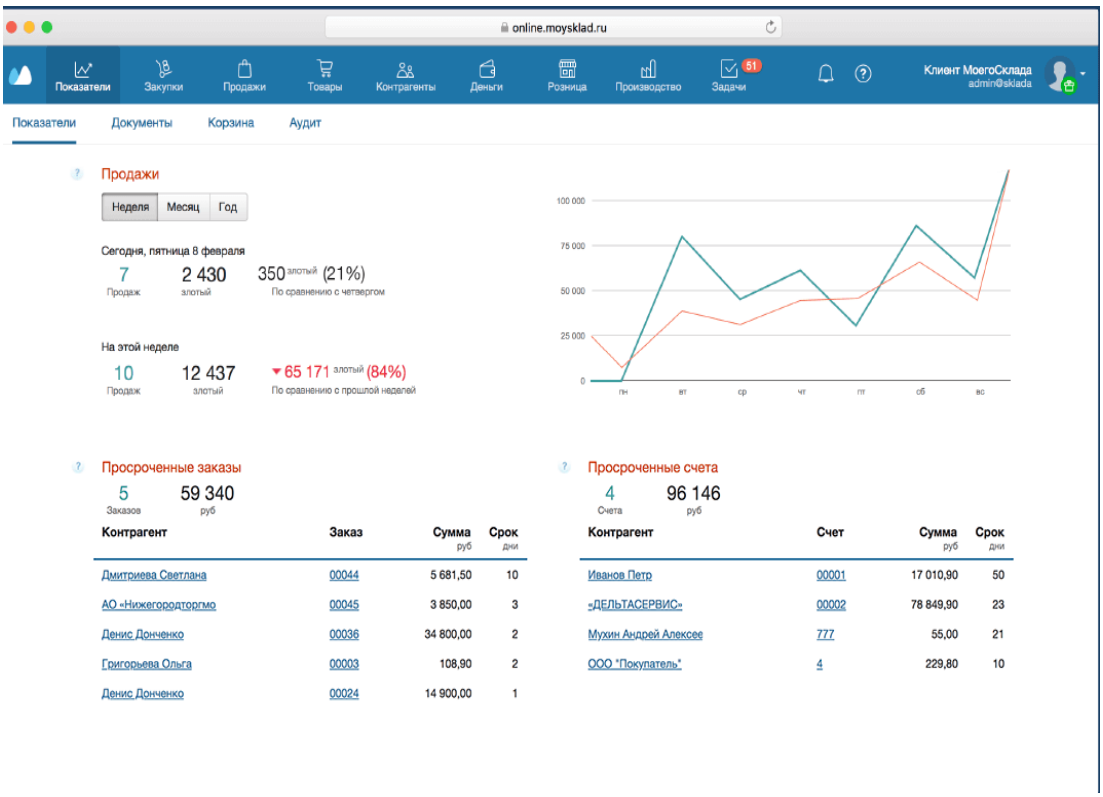


Рисунок 4 – Интерфейс «МойСклад»

1.5 Megaplan

Мегаплан CRM более богата на функционал чем предыдущие система. В системе имеются, управление продажами, учёт остатков товара, финансовый учёт, интеграция с телефонией, управление задачами.

Стоимость от 500 рублей до 1500 рублей за одного пользователя системой. Есть пробный период. Оплата происходит минимум за 3 месяца.

Круглосуточная техническая поддержка, с бесплатным номером 8-800.

Внешний вид представлен на рисунке 5.

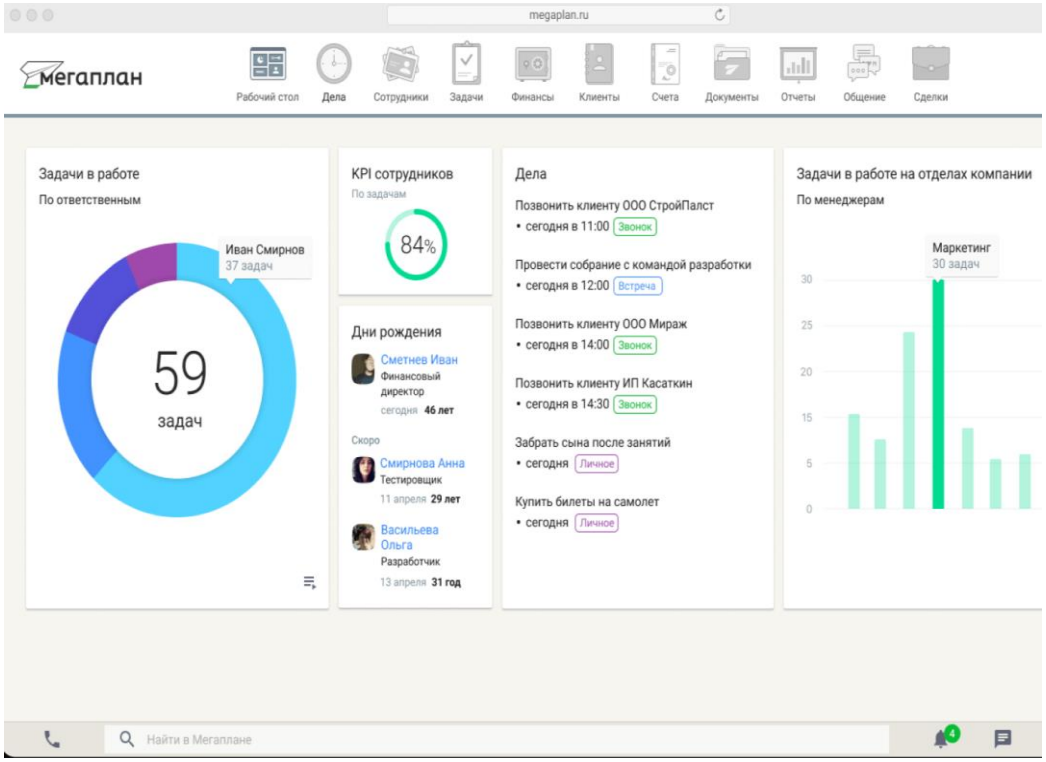


Рисунок 5 – Интерфейс «Megaplan»

1.6 Бизнес.Ру

Выбор функционала в Бизнес.Ру (ранее «Класс365») очень широк, данная CRM предлагает:

- интеграцию с онлайн кассами;
- интеграцию с интернет магазинами;
- автоматические email и смс рассылки;
- формирование бухгалтерских документов;
- учёт остатков;
- управление продажами;

Подсчитать стоимость достаточно сложно, потому что цена формируется в соответствии в выбранными модулями. Минимальная цена 500 рублей в месяц за одного сотрудника.

Круглосуточная техническая поддержка, с бесплатным номером 8-800.

Внешний интерфейс представлен на рисунке 6.

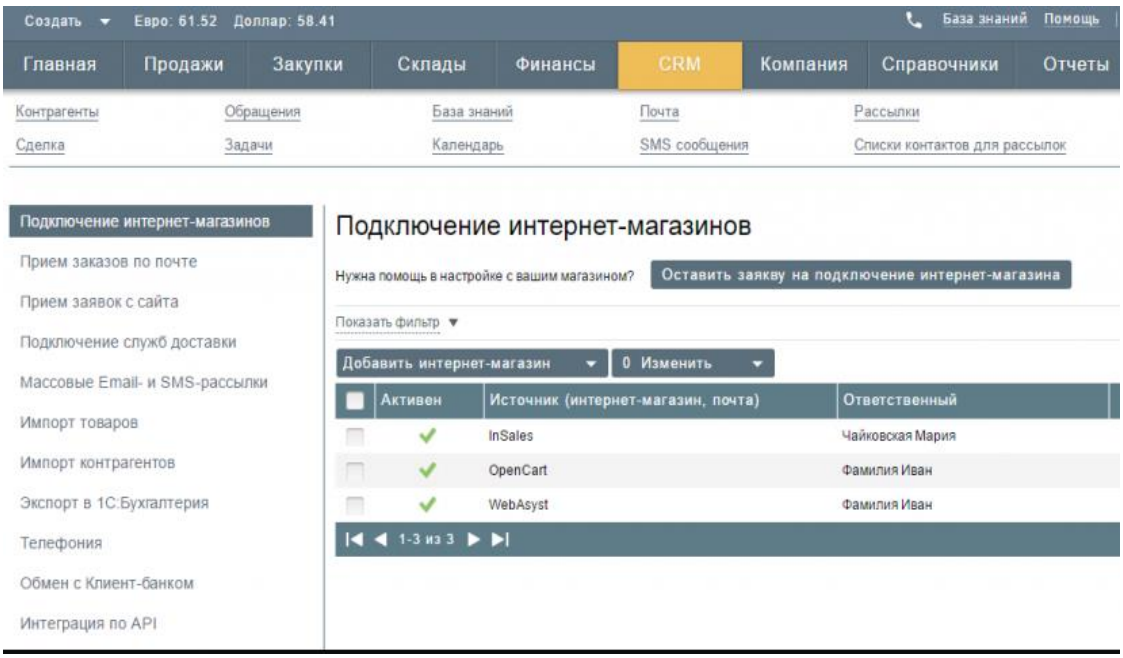


Рисунок 6 – Интерфейс «Бизнес.Ру»

1.7 WireCRM

Данная CRM достаточно простая имеет только один тариф. 400 рублей за одного пользователя. Система предлагает управление продажами, автоматические смс и email рассылки, а так же учёт остатков.

Внешний вид можно увидеть на Рисунке 7.



Рисунок 7 – Интерфейс «WireCRM»

Таблица 1 – Анализ аналогов

Название	Бесплатный период. Дней.	Минимальная цена руб..	Наличие коробочной версии	Круглосуточная техническая поддержка	Бесплатный номер 8-800	Учёт остатков	Автоматические рассылки
Envybox	7	750	-	+	–	-	-
Битрикс24	0	990	+	+	+	+	+
amoCRM	14	499	-	+	-	-	+
МойСклад	14	450	-	+	-	+	-
Мегаплан	14	500	-	+	-	+	+
Бизнес.Ру	0	500	-	+	+	+*	+*
WireCRM	30	400	–	+	-	+	-

*Не во всех тарифных планах.

2 Проектирование

2.1 Проектирование приложения

Для того, что бы начать разработку, необходимо проанализировать сценарий работы администратора и гостя заведения при бронировании и определить доступный функционал обеим ролям.

Функционал для гостя заведения:

- бронировать место/стол в заведении;
- отменить бронирование.

Функционал для администратора CRM:

- добавлять новое бронирование;
- отменить бронирование;
- видеть остаток товаров;
- делать массовые смс или email рассылки;
- видеть данные о частоте бронирований, в определенное время;
- управление настройками системы.

Также необходимо определить основные требования к системе:

- защищенность хранимых данных;
- безопасность хранимых данных от утраты;
- безопасность системы.

Помимо этих качеств, система должна быть способна восстанавливаться за максимально возможное время. И система должна быть способна выдерживать нагрузку до 500 человек online.

2.2 Проектирование архитектуры

Архитектура – это высокоуровневая часть проекта приложения, каркас, состоящий из деталей проекта [2].

Неправильный выбор архитектуры может очень дорого стоить при последующем изменении выбора. Оценив все требования системы, было принято решение реализовывать посредством микросервисной архитектуры.

Микросервисная архитектура — вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей — микросервисов, получивший

распространение в середине 2010-х годов в связи с развитием практик гибкой разработки и DevOps.

Если в традиционных вариантах сервис-ориентированной архитектуры модули могут быть сами по себе достаточно сложными программными системами, а взаимодействие между ними зачастую полагается на стандартизованные тяжеловесные протоколы (такие, как SOAP, XML-RPC), в микросервисной архитектуре системы выстраиваются из компонентов, выполняющих относительно элементарные функции, и взаимодействующие с использованием экономичных сетевых коммуникационных протоколов (в стиле REST с использованием, например, JSON, Protocol Buffers, Thrift). За счёт повышения гранулярности модулей архитектура нацелена на уменьшение степени зацепления и увеличение связности, что позволяет проще добавлять и изменять функции в системе в любое время [3].

Спроектированная архитектура схематично представлена на рисунке 8.

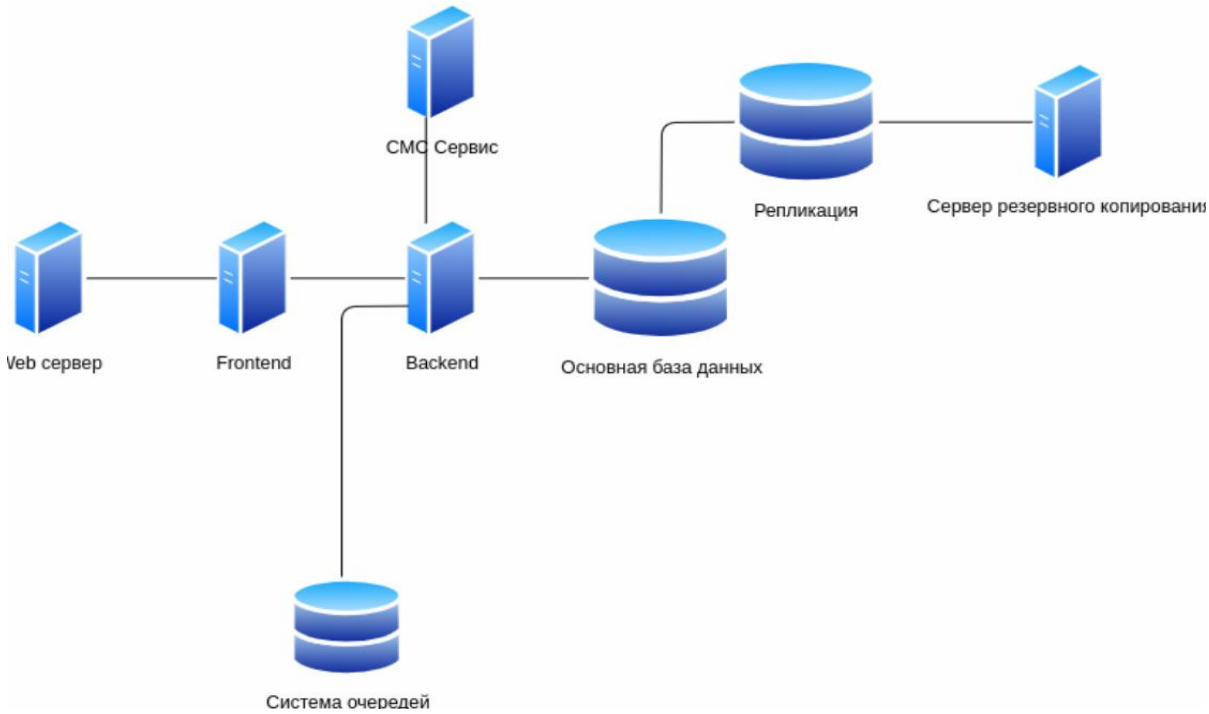


Рисунок 8 – Архитектура приложения

3 Выбор технологий и средств разработки

3.1 Технологии

Определившись со структурой архитектуры, необходимо выбрать конкретные технологии, с помощью которых будет реализована система.

Веб-сервер – Nginx, он полностью отвечает необходимым требованиям, способен выдерживать заданные нагрузки, а так же легок в настройке.

Frontend – на языке JavaScript, с использованием библиотеки ReactJS. Данная библиотека полностью подходит под поставленные задачи, а также у разработчика есть опыт работы с данной библиотекой.

Backend – на языке PHP с использованием фреймворка Laravel. Этот веб-ориентированный язык, позволяет создавать веб приложения любой сложности, а фреймворк позволит ускорить разработку за счёт реализации в себе большой части необходимых функций, что в свою очередь позволит разработчику не тратить время на реализацию этих функций, а использовать готовые библиотеки функций.

Сервер баз данных – Postgresql, современный SQL-сервер, предназначен для работы с большими объемами данных.

Сервер очередей – RabbitMq, основной аналог этого сервера – KafkaMq, но в данном случае RabbitMq больше подходит, так как в KafkaMq достаточно много функций, которые не будут использованы в данной реализации.

Смс-сервер – на языке golang, по причине того, что этот язык отлично работает с большим потоком процессов, что позволит асинхронно совершать большое количество отправок смс сообщений.

Архитектура бэкенда построена в соответствии с объектно-ориентированным подходом и принципами SOLID [4].

Принципы SOLID содержат в себе 5 принципов, по одному на начальную букву.

- Single Responsibility Principle (Принцип единственной ответственности). Класс должен иметь только единую зону ответственности.
- Open-Closed Principle (Принцип открытости-закрытости). Программные сущности (классы, модули, функции) должны быть открыты для расширения, но не для модификации.
- Liskov Substitution Principle (Принцип подстановки Барбары Лисков). Необходимо, чтобы подклассы могли бы служить заменой для своих суперклассов.
- Interface Segregation Principle (Принцип разделения интерфейса). Этот

принцип направлен на устранение недостатков, связанных с реализацией больших интерфейсов.

- Dependency Inversion Principle (Принцип инверсии зависимостей). Модули верхних уровней не должны зависеть от модулей нижних уровней. Оба типа модулей должны зависеть от абстракций.

Также будет использован MVC-подход, но роль view в данном подходе будет выполнять сервер фронтенда. Также данный подход расширен: добавлен ещё один уровень — services, в которых хранится вся бизнес-логика, это позволит не нарушать принципы DRY [5], (don`t repeat yourself), избежание дублирования своего кода. В работе использованы лучшие практики и популярные паттерны программирования.

- Factory [6] (Фабрика), этот паттерн проектирования, который задаёт общий интерфейс для создания объектов в суперклассе, позволяя подклассам изменять тип создаваемых объектов.
- Facade [7] (Фасад), этот паттерн, предоставляет простой интерфейс к сложной системе классов, библиотеке или фреймворку.
- Adapter [8] (Адаптер), паттерн проектирования, позволяющий объектам с несовместимыми интерфейсами работать вместе.
- State (Состояние), патерн позволяет объектам менять поведение в зависимости от своего состояния. Извне создаётся впечатление, что изменился класс объекта.

3.2 Средства разработки

До начала разработки приложения, необходимо полностью настроить рабочее окружение, для удобной и комфортной работы.

Данная работа выполнена в операционной системе Linux Manjaro, потому что существует огромное количество программ, облегчающих работу по разработке web приложений именно для *unix систем, это связанно с тем, что, как правило, все сервера web приложений развёрнуты именно на системах семейства unix. Также, некоторые программы не имеют графического интерфейса, и все взаимодействия с ними происходят с помощью командной строки, система Manjaro предоставляет очень удобный и расширяемый терминал. Система распространяется по открытой лицензии, и доступна для любого пользователя.

В первую очередь, один из главных инструментов разработчика – это IDE (англ. Integrated Development Environment). Разработано много различных IDE, в данной работе

была использована IDE PhpStorm, потому что у разработчика большой опыт работы с данным инструментом. Данная IDE предоставляет обширный функционал: СУБД, встроенный терминал, анализатор кода, режим отладки, генератор http-запросов, интерфейс для работы с системами контроля версий. Интерфейс программы представлен на рис. 9.

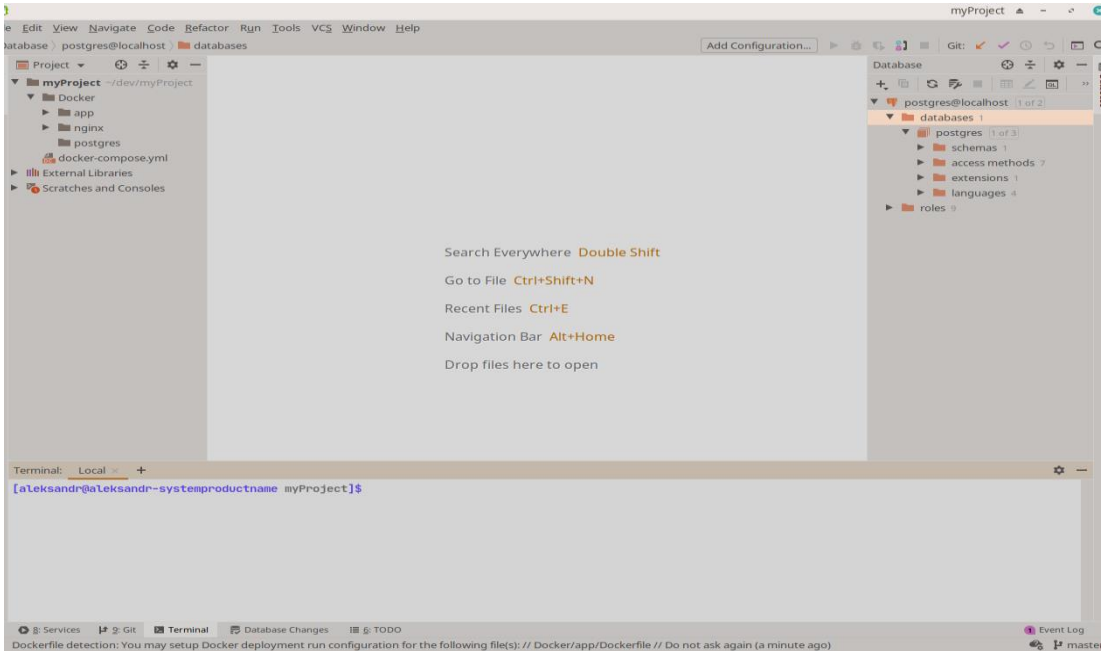


Рисунок 9 – Интерфейс IDE «PHP STORM»

Программа распространяется по платной подписке.

Главная проблема, возникающая при разработке микросервисной архитектуре, как воспроизвести всю архитектуру на одной машине? Можно использовать виртуальные машины, но это слишком дорого обойдётся с точки зрения вычислительных ресурсов. Более легкий вариант – это воспользоваться контейнерами, грубо говоря, контейнер, это облегченный вариант виртуальной машины. Для реализации системы контейнеров отлично подходит программа Docker [9] и утилита docker-compose [10], с помощью которой осуществляется манипуляция контейнерами.

Одним из главных преимуществ docker-compose является соблюдение принципа «Инфраструктура как код». Суть этого принципа состоит в том, что в отдельном файле описываются все необходимые параметры будущего контейнера, а докер уже собирает контейнер из образа. Это позволяет быть уверенным, что вся разработка ведётся в одном и том же окружении и его легко переносить на любой другой компьютер и развернуть проект там.

Заключительным этапом настройки окружения будет подключение системы контроля версий GIT. Для этого был создан репозиторий на GitHub (рис. 10).

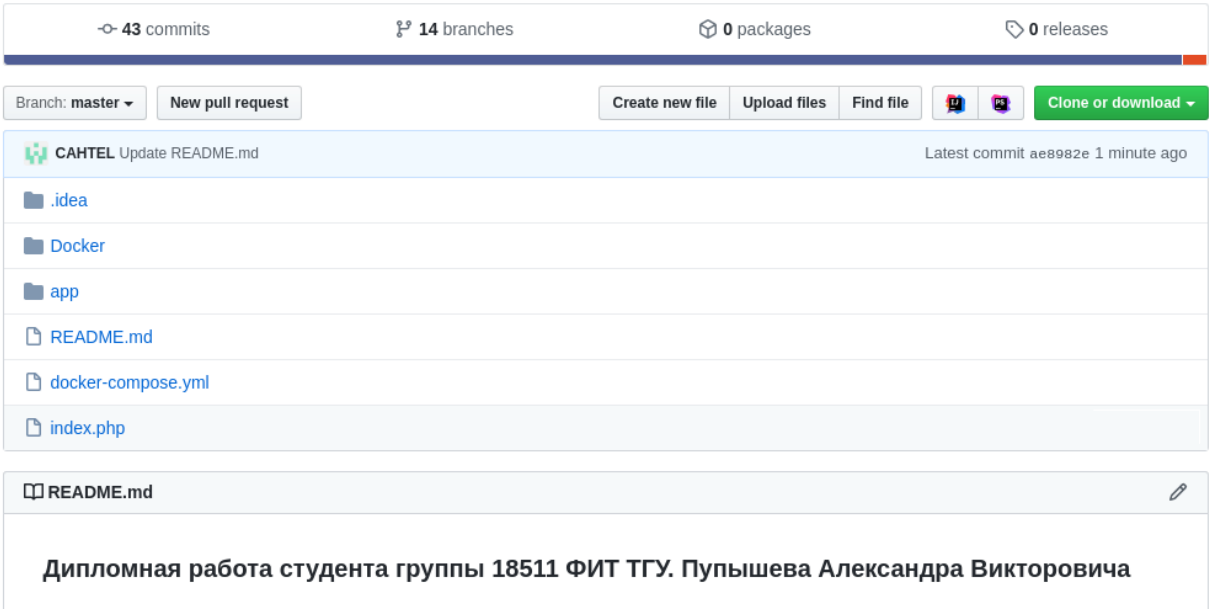


Рисунок 10 – Репозиторий проекта на GitHub

3.3 Проектирование базы данных

Была спроектирована база данных, которая представлена на рис. 11.

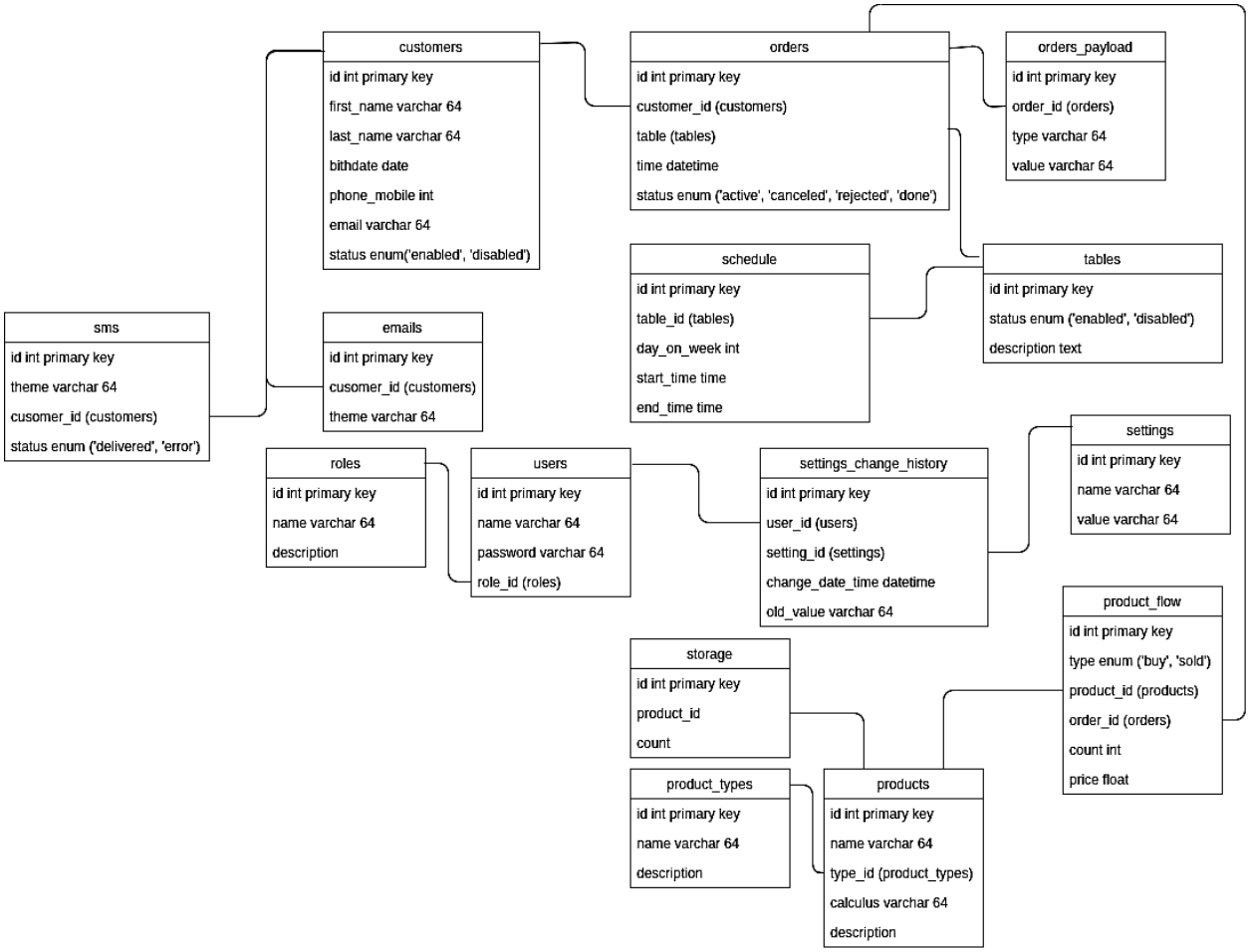


Рисунок 11 – Представление базы данных

Таблица 2.1 – Сущности информационной системы

Имя сущности	Описание
customers	Информация о гостях заведения
orders	Информация о заказах гостей
orders_payload	Дополнительная информация об заказах.
sms	Информация об отправленных смс сообщениях
emails	Информация об отправленных email сообщениях.
tables	Информация о столах в заведении.
schedule	Информация о рабочем времени столов.
users	Информация о пользователях системы (Администраторов заведения)
roles	Информация о ролях пользователей
settings	Информация о настройках

settings_change_history	Историческая информация о смене настроек
product	Информация о продуктах
product_types	Информация о типе продуктов
product_flow	Информация о движениях продуктов со склада и на склад
storage	Информация об остатках продуктов на складе

Таблица 2.2 – Сведения о типах связей

Тип сущности	Название связи	Тип сущности	Кардиналь-ность
customers	Связан с	sms	1:M
customers	Связан с	emails	1:M
customers	Связан с	orders	1:M
orders	Связан с	orders_payload	1:1
orders	Связан с	product_flow	1:1
orders	Связан с	tables	1:1
tables	Связан с	schedule	1:M
roles	Связан с	users	1:M
users	Связан с	settings_change_history	M:M
settings	Связан с	settings_change_history	1:M
product	Связан с	product_types	1:1
product	Связан с	storage	1:1
product	Связан с	product_flow	1:M

Таблица 2.3 – Атрибуты сущностей

Сущность	Атрибут сущности	Домен атрибута
customers	id	Ключ
	first_name	Текстовый
	last_name	Текстовый
	bithdate	Текстовый
	phone_mobile	Целочисленный
	status	Текстовый
orders	id	Ключ
	customer_id	Целочисленный

	table_id	Целочисленный
	time	Дата и время
	status	Текстовый
orders_payload	id	Ключ
	order_id	Целочисленный
	type	Текстовый
	value	Текстовый
sms	id	Ключ
	theme	Текстовый
	customer_id	Целочисленный
	status	Текстовый
emails	id	Ключ
	theme	Текстовый
	customer_id	Целочисленный
tables	id	Ключ
	status	Текстовый
	description	Текстовый
schedule	id	Ключ
	table_id	Целочисленный
	day_of_week	Целочисленный
	start_time	Временной
	end_time	Временной
users	id	Ключ
	name	Текстовый
	password	Текстовый
	role_id	Целочисленный
roles	id	Ключ
	name	Текстовый
	description	Текстовый
settings	id	Ключ
	name	Текстовый

	value	Текстовый
	id	Ключ
	user_id	Целочисленный
settings_change_history	setting_id	Целочисленный
	change_date_time	Дата и время
	old_value	Текстовый
	id	Ключ
	name	Текстовый
products	type_id	Целочисленный
	calculus	Текстовый
	description	Текстовый
	id	Ключ
product_types	name	Текстовый
	description	Текстовый
	id	Ключ
	type	Текстовый
	product_id	Целочисленный
product_flow	order_id	Целочисленный
	count	Целочисленный
	price	Числовой
	id	Ключ
storage	name	Текстовый
	count	Целочисленный

4 Реализация

4.1 Описание проекта

Проанализировав аналоги и возможности выбранных инструментов, выведем критически важные характеристики реализуемого приложения:

- web-приложение адаптивное под мобильные устройства;
- возможность ведения онлайн-бронирований и учёта остатков товаров;
- хранение и защита данных пользователей;
- гибкость настройки приложения для каждого заведения.

Предлагаются следующие этапы развития приложения:

- создание первой предварительной версии приложения;
- подключение к приложению двух заведений;
- доработка приложения, выпуск второй предварительной версии;
- подключение к приложению десять заведений;
- сбор обратной связи;
- выпуск финальной версии приложения.

В списке предусмотрено два этапа бета-тестирования. Это сделано для получения обратной связи, потому что приложение в первую очередь делается для людей, поэтому получить обратную связь на раннем этапе очень важно для определения вектора движения дальнейшей разработки.

На 15 июня 2020 года, приложение прошло первые три этапа, т.е. первые два заведения уже смогли оценить приложение и дать обратную связь, после получения которой, приложение было доработано.

Список доработок:

- добавлена настройка, отвечающая за автоматическое блокирование недобросовестных пользователей, которые бронируют столик и не посещают заведение;
- добавлена настройка, регламентирующая количество запросов смс подтверждения на один номер телефона;
- добавлена возможность сохранять счёт клиента за одно бронирование;
- добавлена возможность указывать имя клиента при sms- рассылке.

4.2 Результат реализации

После подключения заведения к системе, становятся доступны два интерфейса: общий интерфейс для записи (рис. 12), который доступен любому пользователю, и панель администратора, которая доступна только представителям заведения после авторизации.

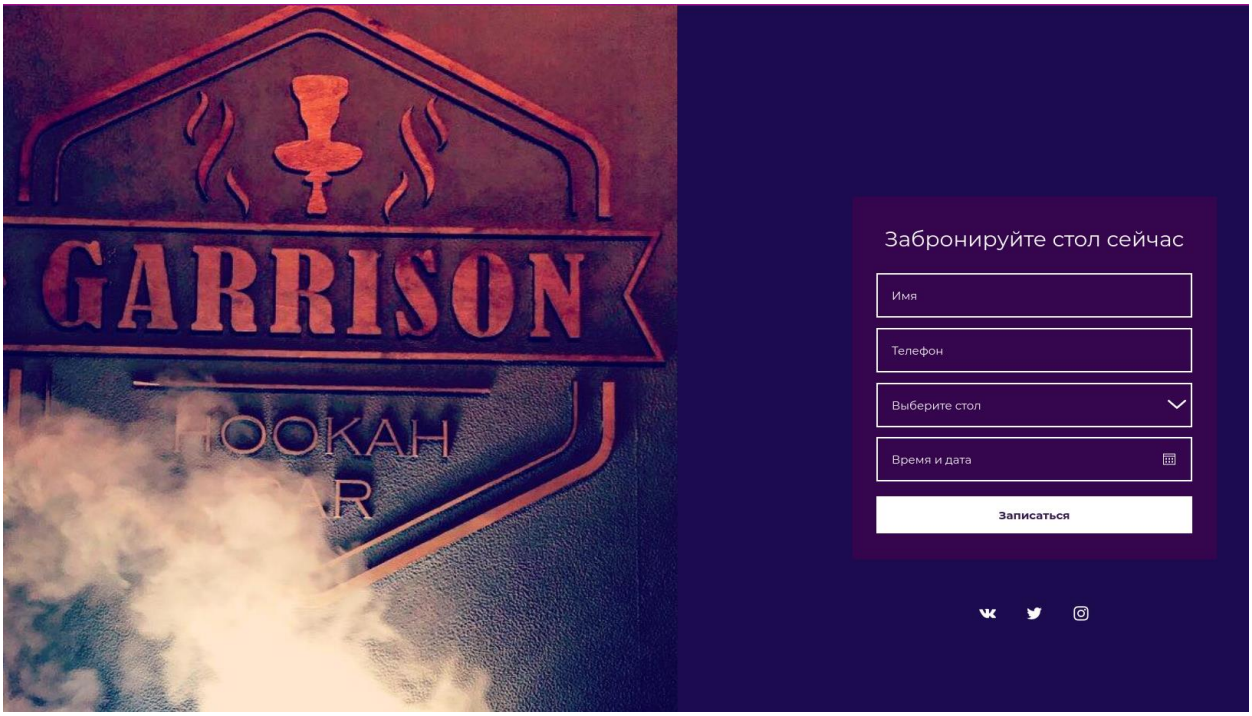


Рисунок 12 – Интерфейс онлайн бронирования

Все заявки со страницы бронирований поступают на панель администратора. Эти два компонента находятся на разных машинах, связанных между собой с помощью RestAPI. Поэтому сбой в работе на одной из них никак не повлияет на работоспособность второй, что значительно повысит отказоустойчивость и скорость восстановления.

REST – (сокр. от англ. Representational State Transfer — «передача состояния представления») архитектурный стиль взаимодействия компонентов распределённого приложения в сети по модели клиент-сервер. Был разработан как альтернатива SOAP, когда запрос клиента несет в себе исчерпывающую информацию о желаемом ответе сервера и сервер не обязан сохранять сессию взаимодействия с клиентом.

Подробнее процесс бронирования выглядит следующим образом.

Пользователь заполняет форму бронирования и отправляется запрос на бэкенд сайта, в случае, если бронирование данного стола на данное время доступно, бэкенд делает запрос на отправку смс-сообщения с кодом подтверждения, и система ожидает от пользователя ввод смс-кода для записи бронирования. В случае, если данный стол недоступен для бронирования в данное время, то пользователю вернется ошибка.

Ограничение на количество ввода смс и на повторный запрос формируется администратором заведения на странице настроек.

4.3 Панель администратора

При попадании на панель управления ранее неавторизированным пользователем система предложит авторизоваться, введя логин и пароль.(рис. 13).

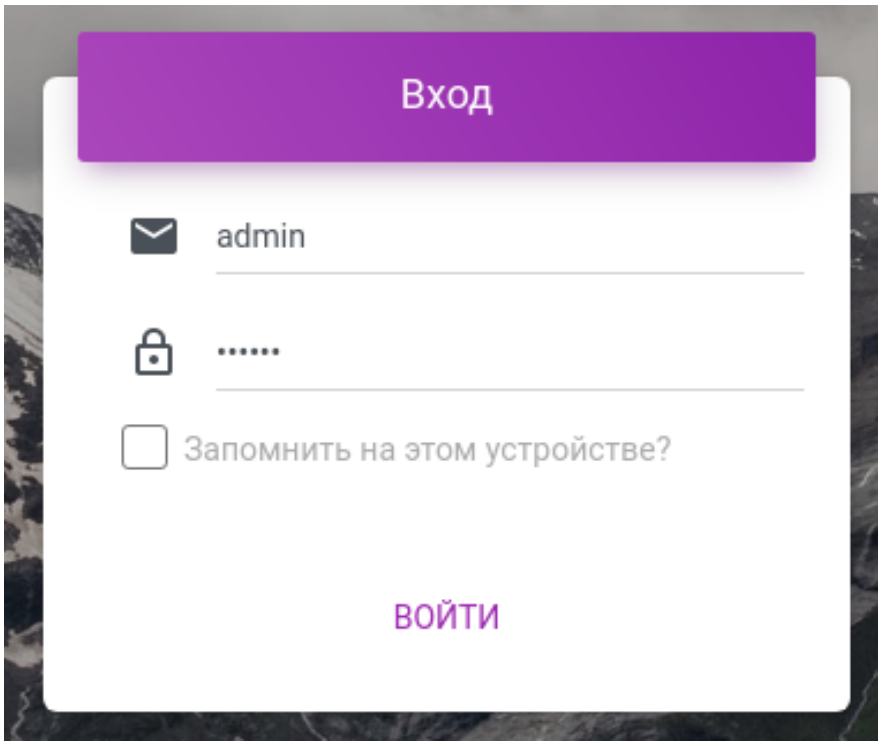


Рисунок 13 – Окно входа в панель управления

Введя логин и пароль, пользователь попадает на главную страницу панели управления - «Рабочий стол» (рис. 14).

Там расположены:

- графики отображающие информацию по бронированиям, отправленным смс и среднему чеку за неделю (рис. 15);
- в центре расположены две таблицы, с текущими (рис. 16) и ближайшими бронированиями (рис.17).
- под таблицами расположена кнопка, вызывающая форму добавления бронирования (рис. 18).

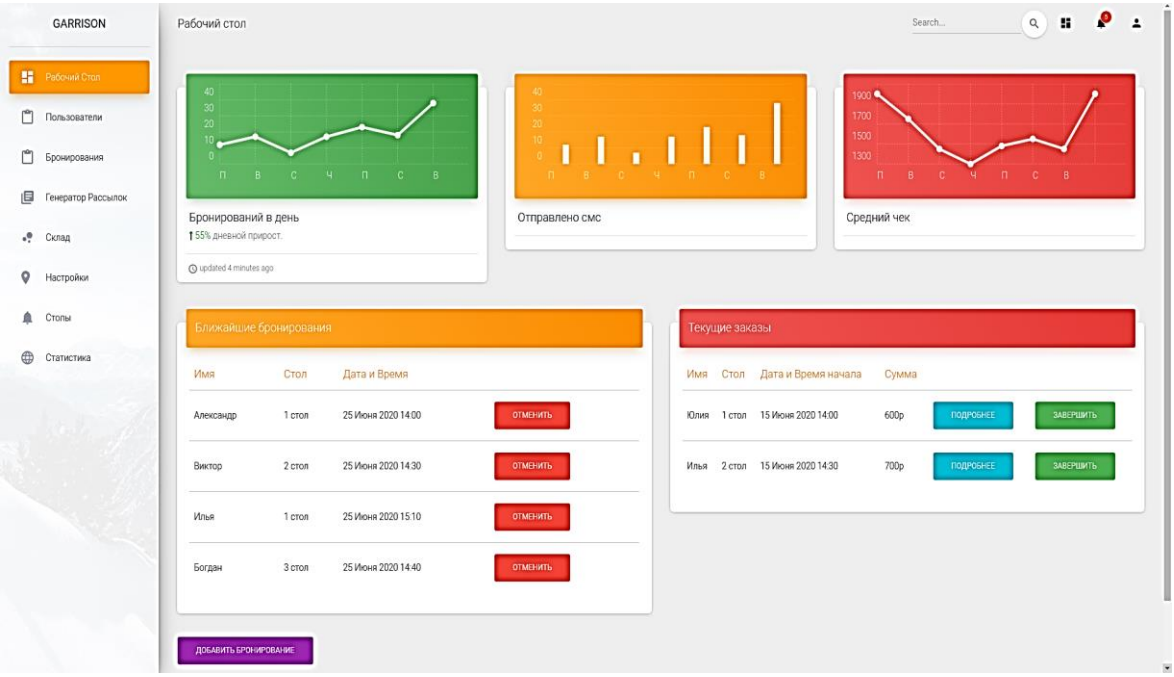


Рисунок 14 – Страница «Рабочий стол»

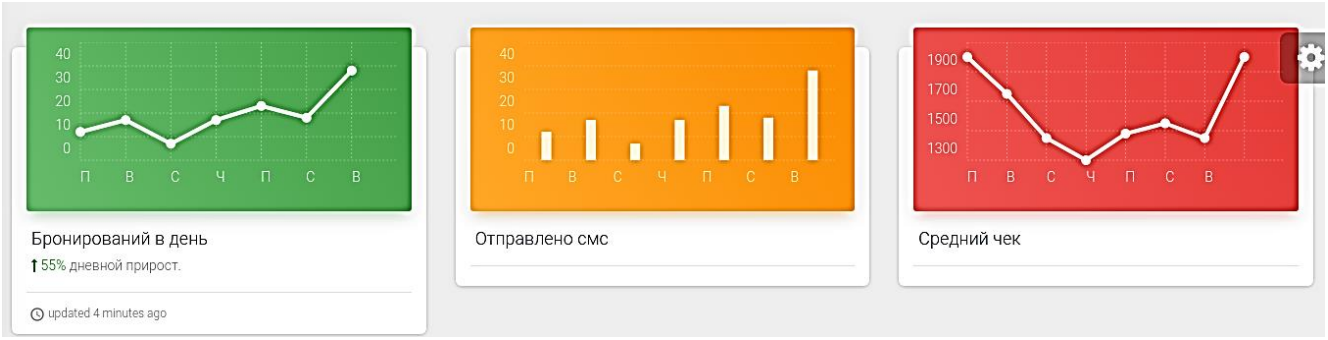


Рисунок 15 – Графики

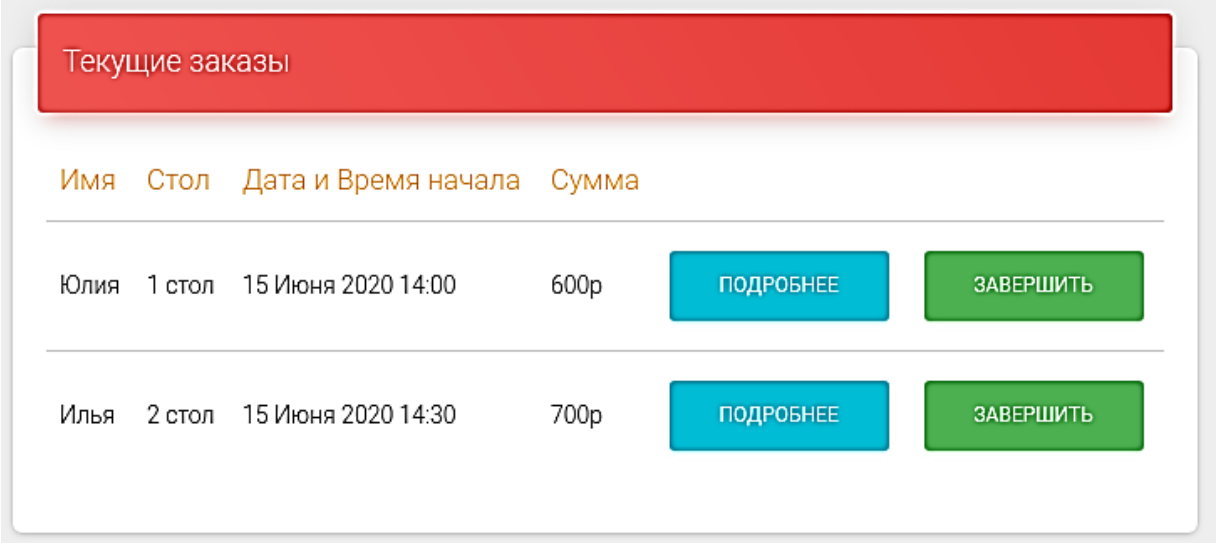


Рисунок 16 – Таблица «Текущие заказы»

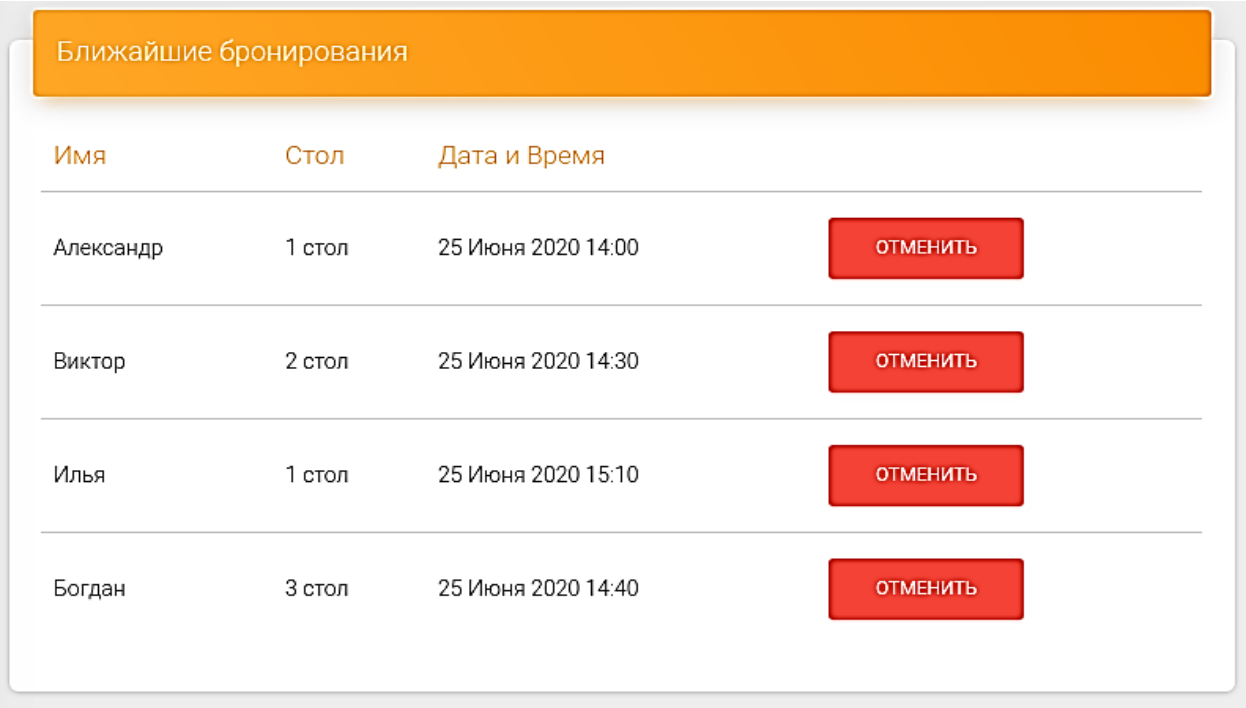


Рисунок 17 – Таблица «Ближайшие бронирования»

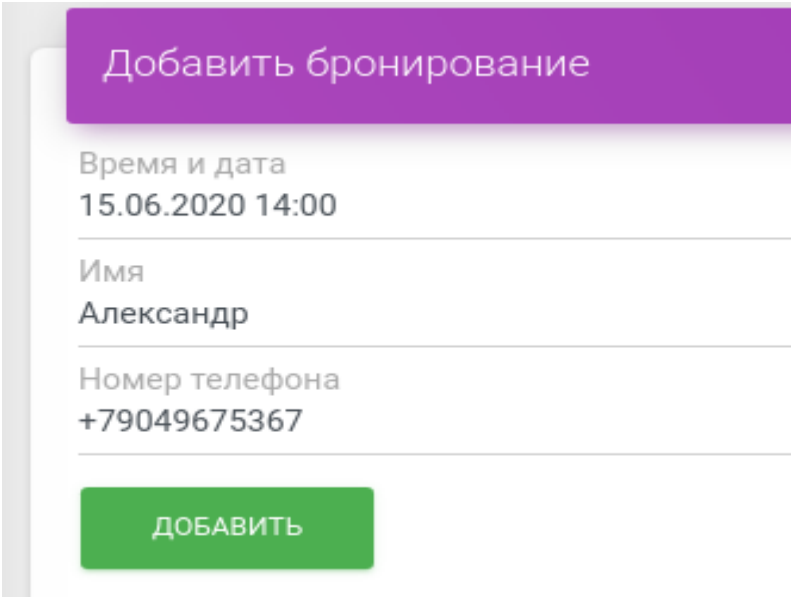


Рисунок 18 – Форма добавления бронирования

Для завершения заказа необходимо заполнить форму завершения (рис. 19), просмотр деталей заказа и его изменение происходит с помощью отдельной формы (рис. 20).

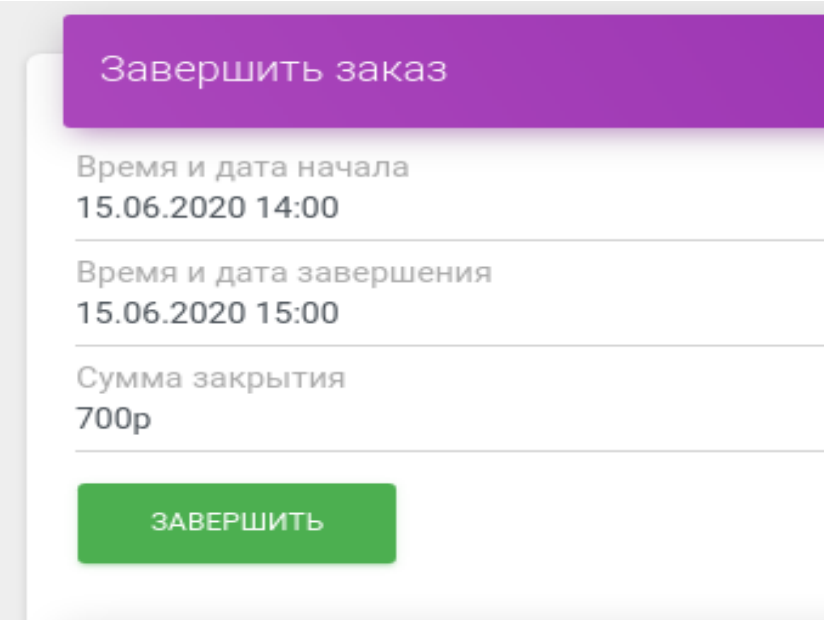


Рисунок 19 – Форма завершения заказа

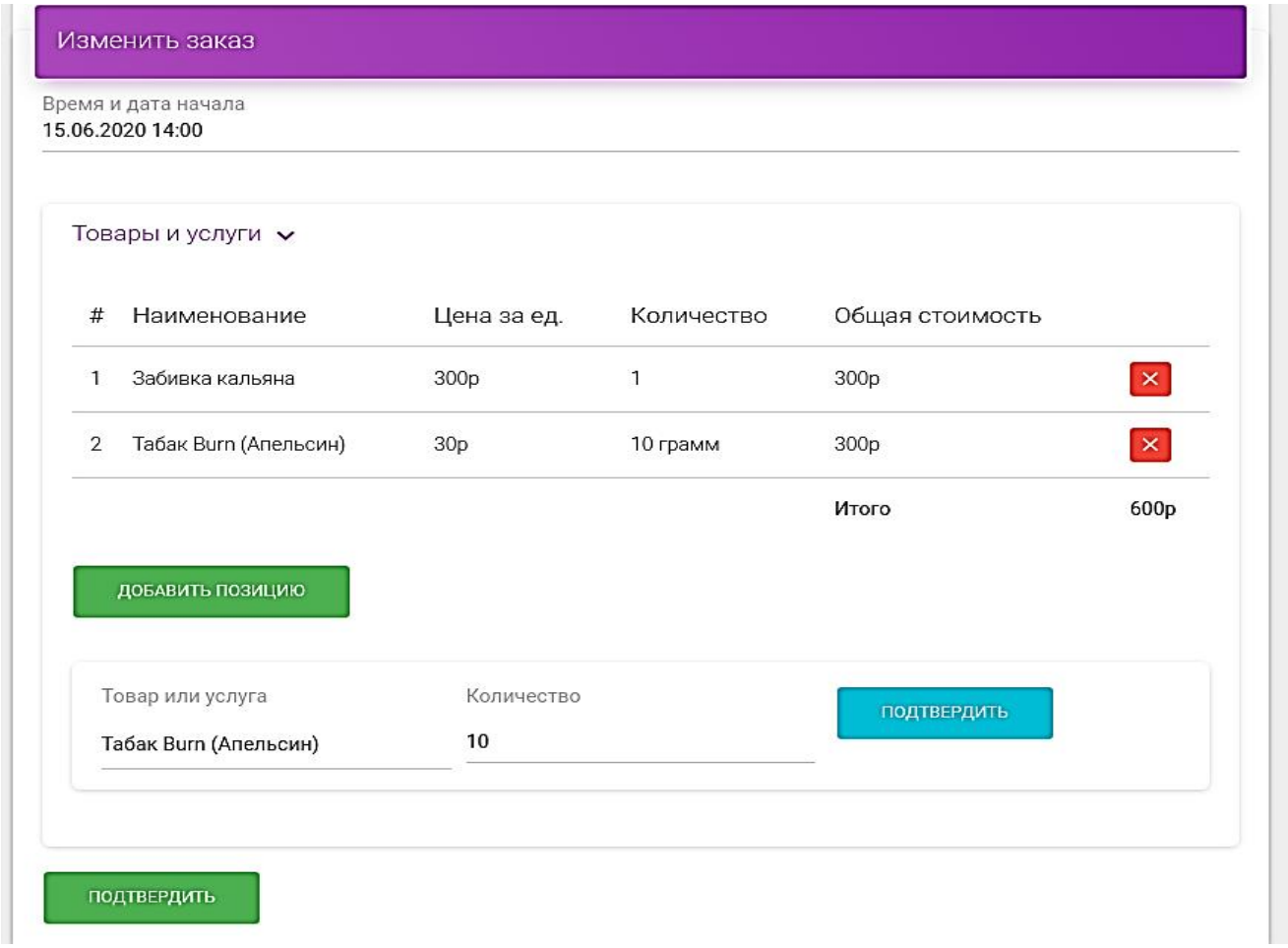


Рисунок 20 – Форма изменения заказа

С помощью главного меню, расположенного на левой части страницы (рис. 19), доступна навигация.

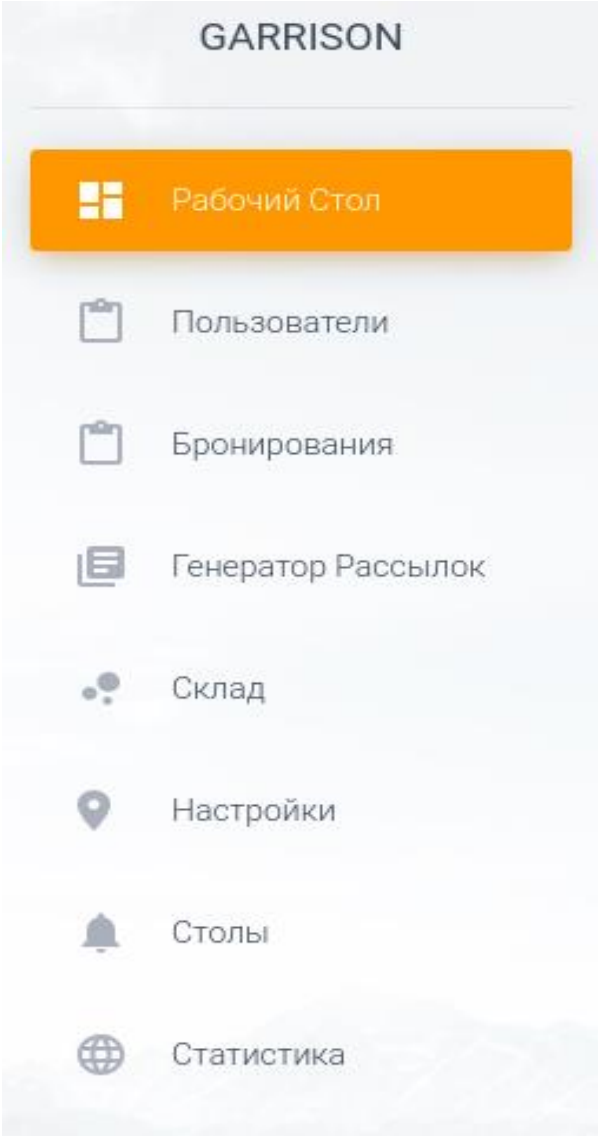


Рисунок 21 – «Главное меню»

На странице «Список пользователей» (рис. 22) представлены все пользователи, зарегистрированные в системе. Также доступен функционал для блокировки пользователей и просмотра истории бронирований пользователей (рис. 23).

Вход в панель управления осуществляется с помощью формы авторизации

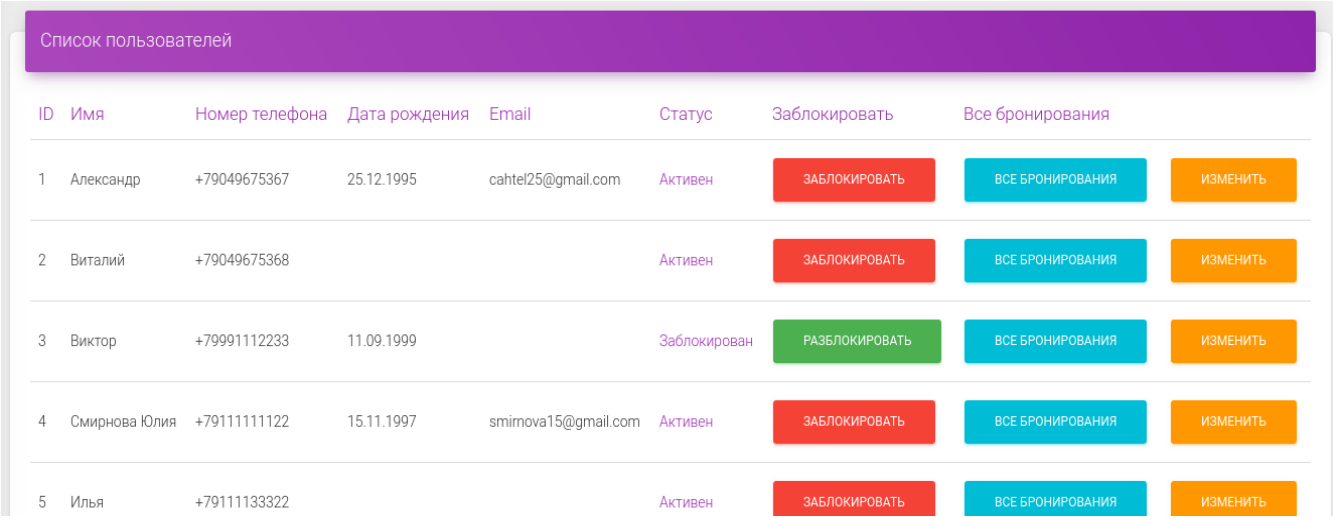


Рисунок 22 – Список пользователей

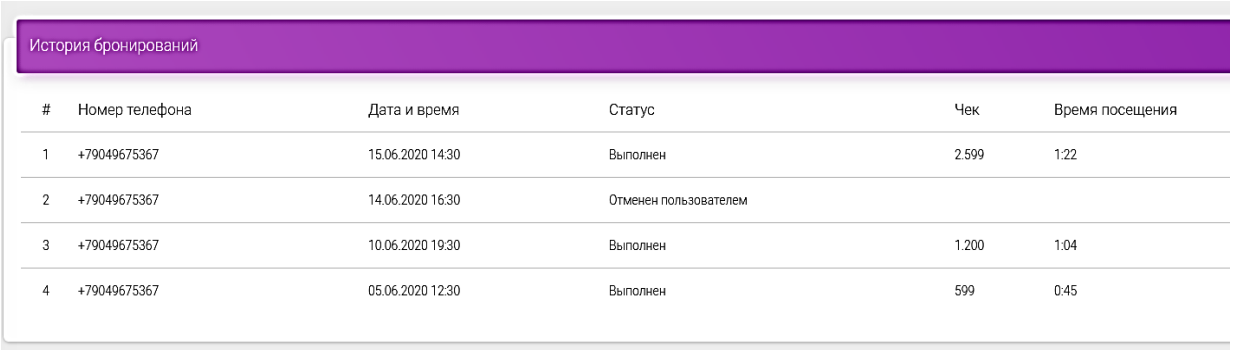


Рисунок 23 – История бронирований пользователя

Далее по списку меню – «Бронирования» (рис. 24), на этой странице доступны к просмотру все бронирования, есть возможность фильтровать их по статусу и дате с помощью формы над таблицей.

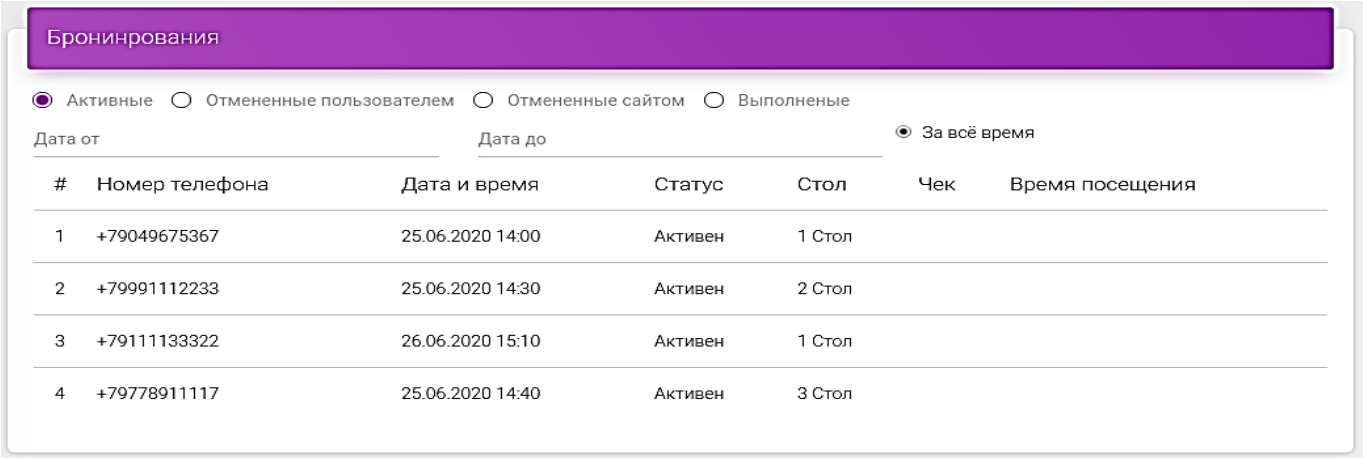


Рисунок 24 – Бронирования

Следующий пункт, это создание рассылок (рис. 25). В данный момент функционал принимает на вход уникальные идентификаторы пользователей, в следующем обновлении

планируется внедрить конструктор аудитории, который будет автоматически выбирать нужных людей из заданных параметров.

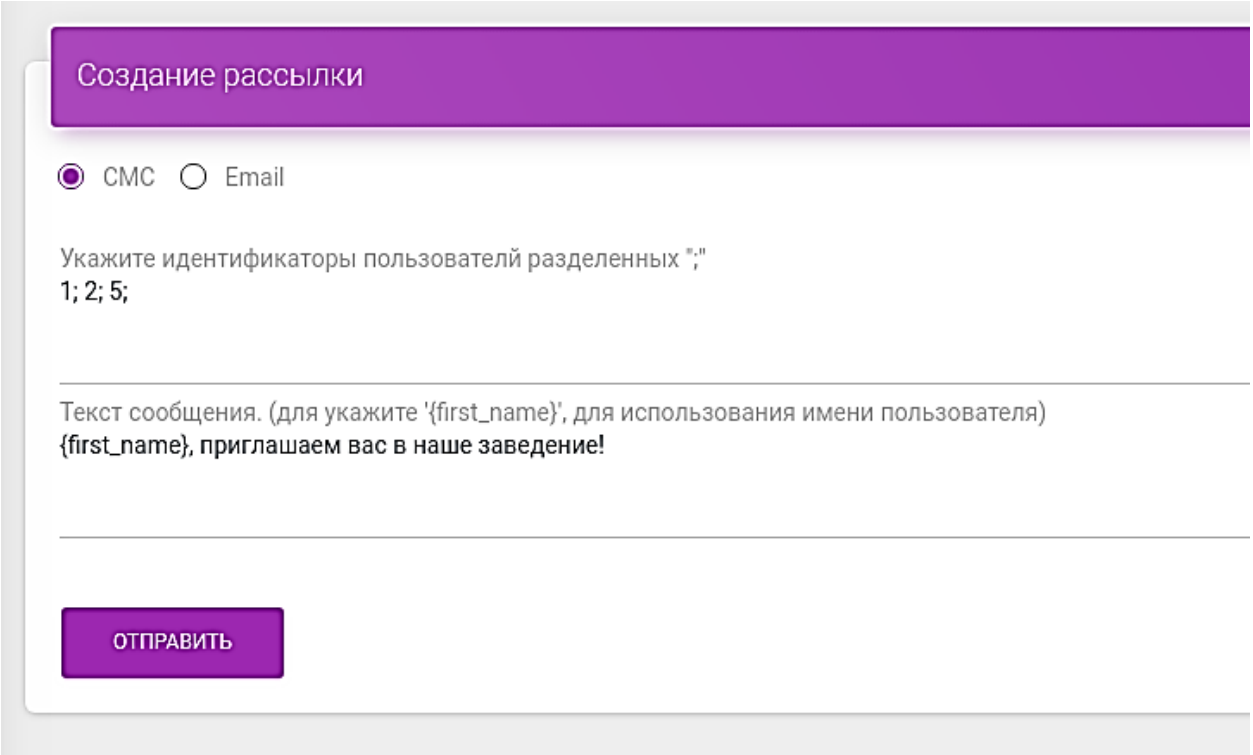


Рисунок 25 – Создание рассылок

В пункте меню «Склад»(рис.26), находится информация об остатках товара, с возможностью добавить (рис.27) или списать (рис. 28) определенное количество товара.

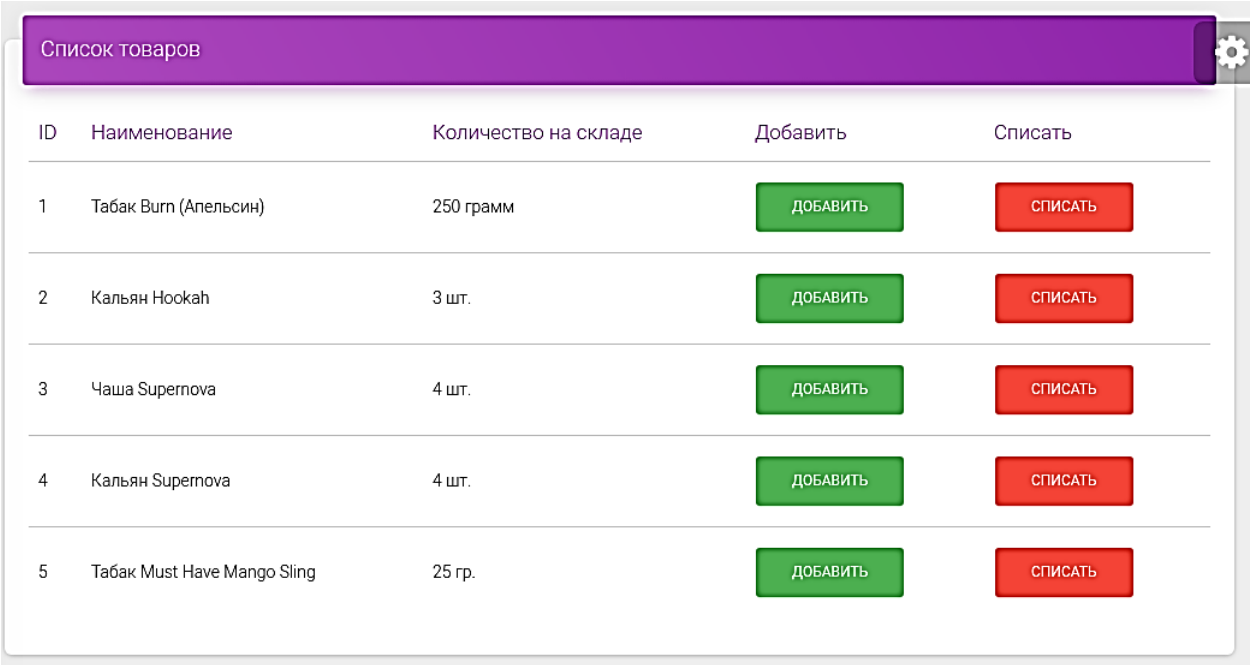


Рисунок 26 – Список товаров

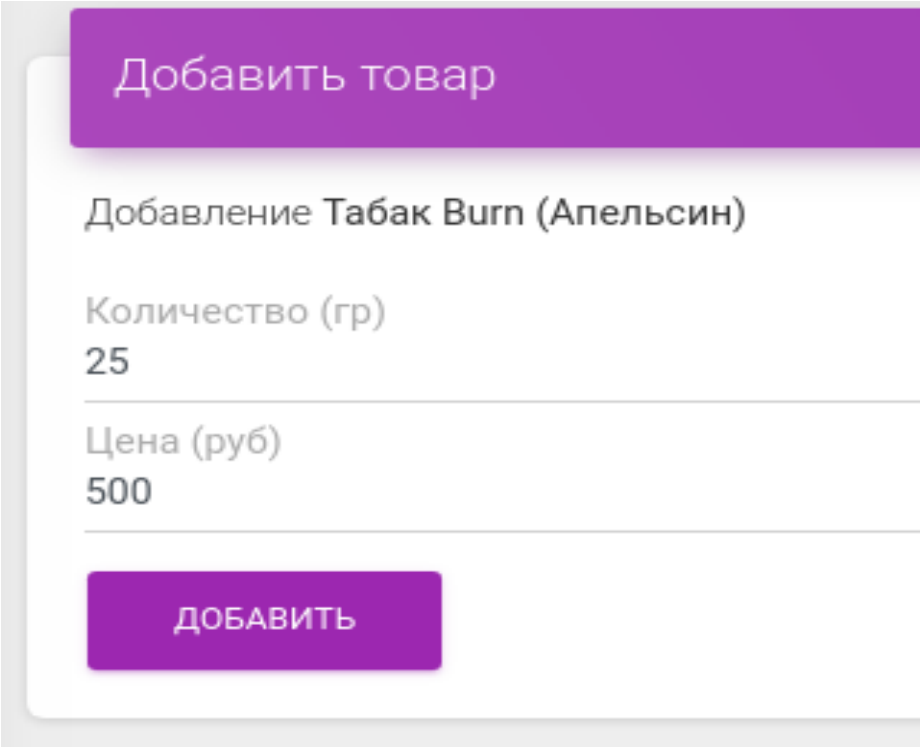


Рисунок 27 – Форма «Добавление товара»

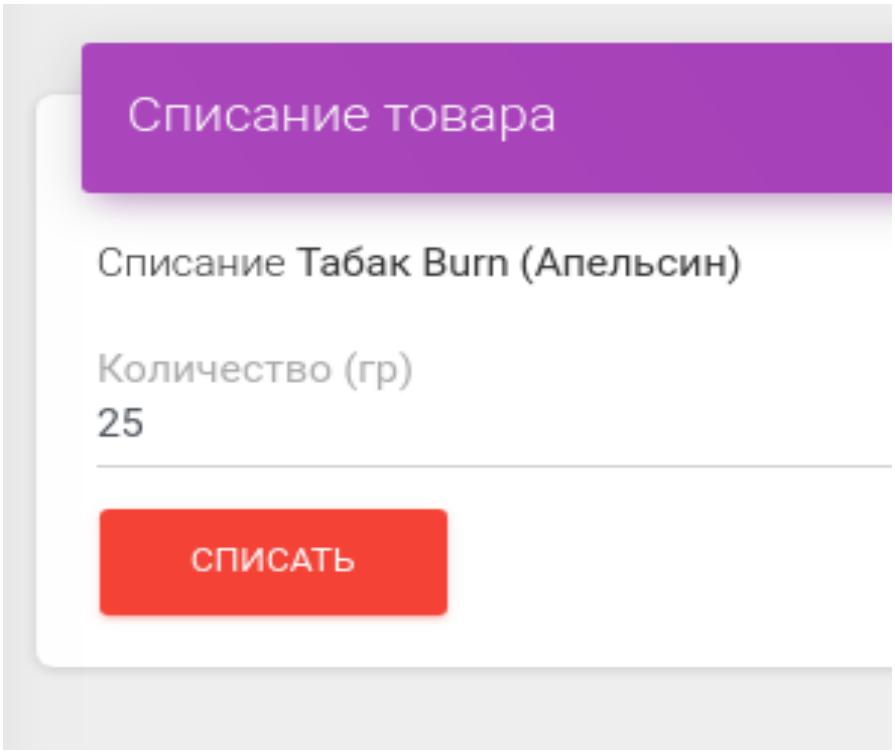


Рисунок 28 – Форма «Списание товара»

Управление настройками (рис. 29) выполнено в режиме «ключ-значение», для настроек в которых предусматривается только значение «включено» или «отключено», используется «1» или «0» соответственно.

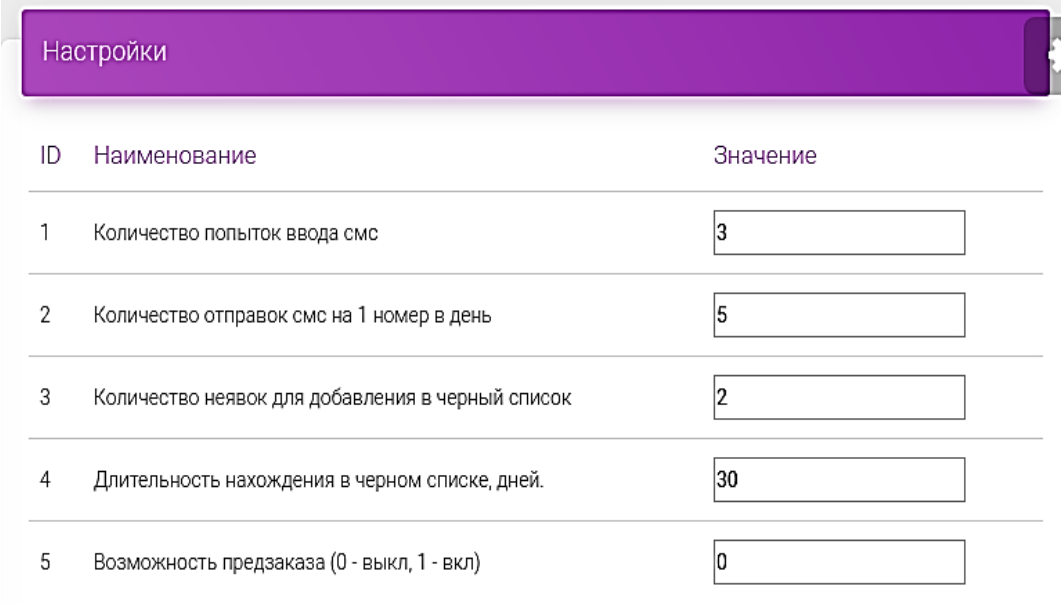


Рисунок 29 – Настройки

Управление столами и их расписанием происходит на вкладке «Столы» (рис. 30), помимо управления расписанием на странице можно добавить новый стол и временно отключить имеющиеся. Форма добавления нового стола представлена на рисунке 31.

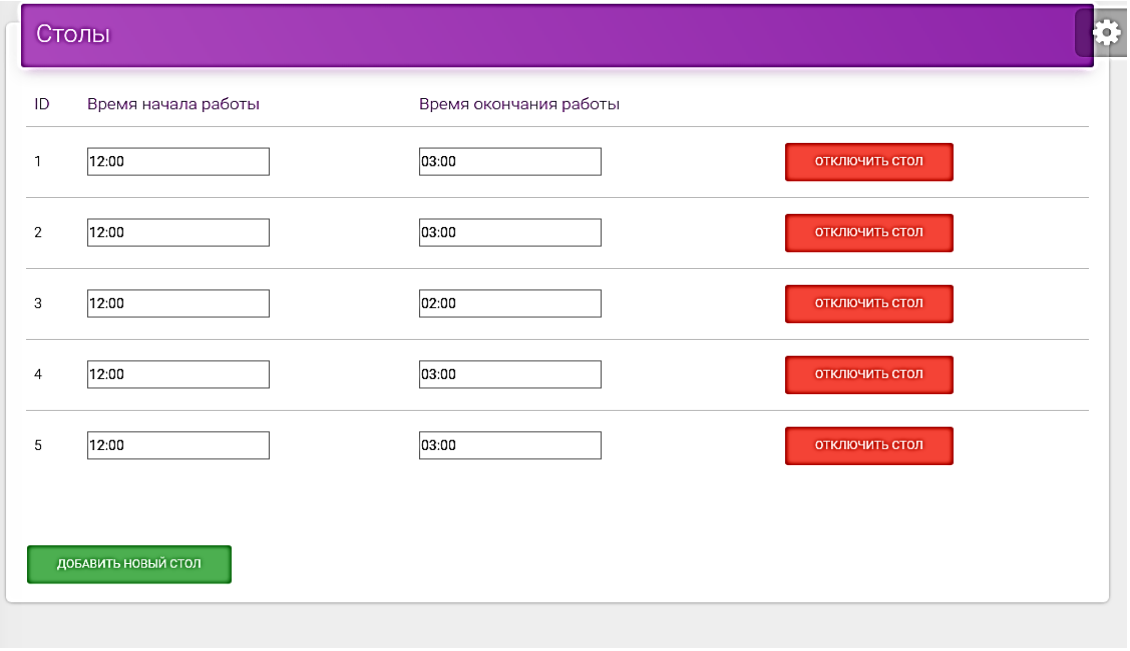


Рисунок 30 – Страница «Столы»

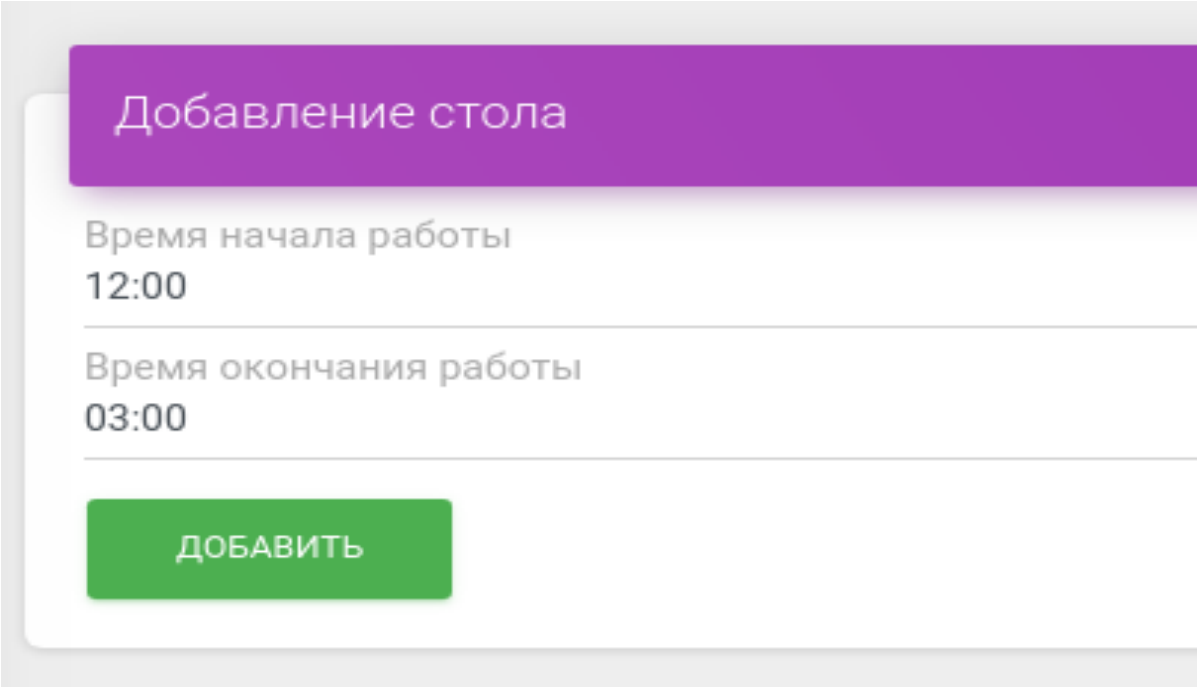


Рисунок 31 – Форма «Добавление стола»

В данной версии на странице «Статистика» (рис. 32), есть возможность определять фиксированные временные рамки. В следующем обновлении будет внедрена возможность указывать гибкие временные рамки, а также будут добавлены графики.

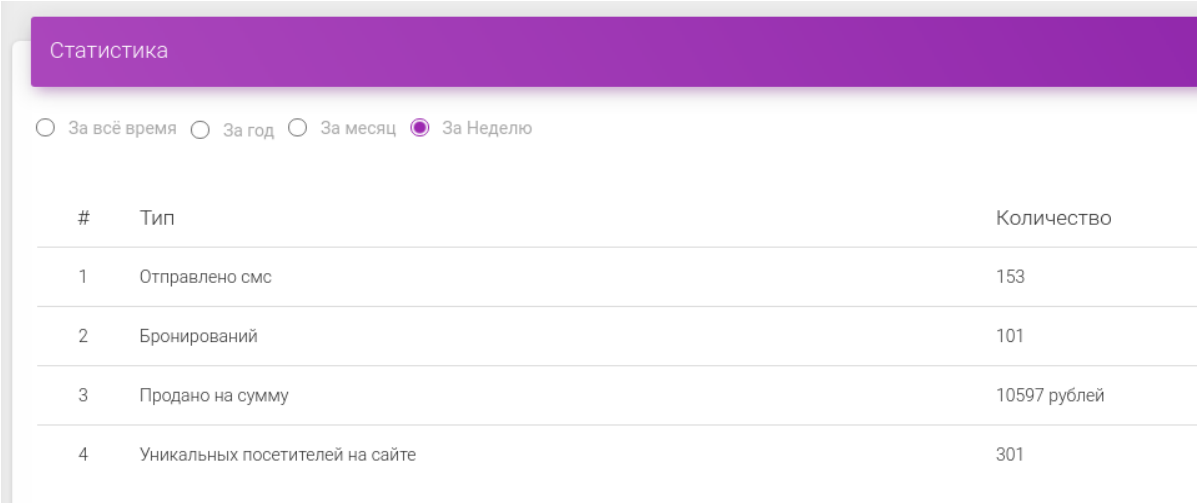


Рисунок 32 – Страница «Статистика»

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР был проведен анализ существующих CRM-систем для управления малым бизнесом. Исходя из результатов анализа, было принято решение о проектировании и создании собственного приложения.

На этапе проектирования была определена архитектура приложения и спроектирована база данных.

После определения требований и архитектуры приложения были проанализированы инструментальные среды, используемые для разработки веб-ориентированных систем:

Заключительным этапом была разработка и тестирование web-приложения с микросервисной архитектурой, взаимодействующей по REST API, что позволил решить поставленные в работе задачи.

У данной системы имеется перспектива развития: после второго этапа тестирования система будет значительно доработана, как в плане функционала так и в плане дизайна, будут привлечены сторонние компании, разрабатывающие дизайн, для обеспечения возможности выбирать и менять дизайн панели администратора. На следующем этапе будет внедрена интеграция с двумя популярными социальными сетями, такими как «Вконтакте» и «Instagram».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 10 лучших CRM для малого бизнеса в России [Электронный ресурс]. – URL: https://startpack.ru/ratings/10-best-crm-smb-in-russia (дата обращения 14.06.2020)
2. Макконнелл, Стив. Совершенный код, «Русская Редакция», 2019, 896 с.
3. REST. Описание концепции. Особенности реализации API [Электронный ресурс]. – URL: https://ziginsider.github.io/rest-api/ (дата обращения 14.06.2020)
4. Принципы SOLID [Электронный ресурс]. – URL: https://medium.com/webbdev/solid-4ffc018077da (дата обращения 14.06.2020)
5. Принцип программирования DRY [Электронный ресурс]. – URL: https://web-creator.ru/articles/dry (дата обращения 14.06.2020)
6. Фабричный метод [Электронный ресурс]. – URL: https://refactoring.guru/ru/design-patterns/factory-method (дата обращения 14.06.2020)
7. Фасад [Электронный ресурс]. – URL: https://refactoring.guru/ru/design-patterns/facade (дата обращения 14.06.2020)
8. Адаптер [Электронный ресурс]. – URL: https://refactoring.guru/ru/design-patterns/adapter (дата обращения 14.06.2020)
9. Docker Docs – Docker [Электронный ресурс]. – URL: https://docs.docker.com/ (дата обращения 14.06.2020)
10. Docker Docs – Docker Compose [Электронный ресурс]. – URL: https://docs.docker.com/compose/ (дата обращения 14.06.2020)

АНТИПЛАГИАТ
ТВОРИТЕ СОБСТВЕННЫМ УМОМ

Томский Государственный Университет

СПРАВКА

о результатах проверки текстового документа на наличие заимствований

Проверка выполнена в системе Антиплагиат.ВУЗ

Автор работы	**Пупышев Александр Викторович**
Подразделение	кафедра информационного обеспечения инновационной деятельности
Тип работы	Выпускная квалификационная работа
Название работы	Создание CRM-системы для малого бизнеса
Название файла	Пупышев ВКР_исправления.pdf
Процент заимствования	**12.25 %**
Процент самоцитирования	**0.00 %**
Процент цитирования	**6.69 %**
Процент оригинальности	**81.06 %**
Дата проверки	**06:30:05 17 июня 2020г.**
Модули поиска	Модуль поиска ИПС "Адилет"; Модуль выделения библиографических записей; Сводная коллекция ЭБС; Модуль поиска "Интернет Плюс"; Коллекция РГБ; Цитирование; Модуль поиска переводных заимствований; Модуль поиска переводных заимствований по elibrary (EnRu); Модуль поиска переводных заимствований по интернет (EnRu); Коллекция eLIBRARY.RU; Коллекция ГАРАНТ; Коллекция Медицина; Диссертации и авторефераты НББ; Модуль поиска перефразирований eLIBRARY.RU; Модуль поиска перефразирований Интернет; Коллекция Патенты; Модуль поиска общеупотребительных выражений; Модуль поиска "ТГУ"; Кольцо вузов
Работу проверил	Гольцова Полина Андреевна
	ФИО проверяющего
Дата подписи	17.06.2020

Подпись проверяющего

Чтобы убедиться в подлинности справки, используйте QR-код, который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.