Санкт-Петербургский государственный университет

**Кафедра математической теории микропроцессорных систем управления**

**Коркин Александр Николаевич**

**Дипломная работа**

**Программная поддержка предприятия общественного питания**

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
В.А. Петрова

Санкт-Петербург

2016

Содержание

[Введение 3](#_Toc481531680)

[Обзор инструментов автоматизации ресторанного бизнеса 5](#_Toc481531681)

[Постановка задачи 15](#_Toc481531682)

Глава [1. Анализ предметной области предприятия общественного питания 17](#_Toc481531685)

[1.1. Анализ работы ресторана 17](#_Toc481531686)

[1.2. Моделирование работы ресторана 19](#_Toc481531687)

Глава [2. Проектирование программы поддержки работы ресторана 24](#_Toc481531688)

[2.1. Определение целей и функций работы программного продукта 24](#_Toc481531689)

[2.2. Анализ и выбор инструментов для создания программы поддержки работы ресторана 29](#_Toc481531690)

[2.3. Разработка базы данных программы 33](#_Toc481531691)

[2.4. Анализ требований работы сервисов 39](#_Toc481531692)

Глава [3. Описание реализации программного продукта 41](#_Toc481531693)

[3.1. Описание основных алгоритмов реализации программы поддержки работы ресторана 41](#_Toc481531694)

[3.2. Описание работы пользователей в системе 44](#_Toc481531695)

[3.3. Тестирование разработанной системы 52](#_Toc481531696)

Выводы…………………………………………………………………………...56

[Заключение 5](#_Toc481531698)7

[Список цитируемой литературы](#_Toc481531699) 59

Приложение………………………………………………………………………61

Введение

Чтобы успешно функционировать в современных экономических условиях, предприятию необходимо располагать эффективной информационной системой, которая обеспечивает быстрый поиск необходимой информации, и ее обработку.

С увеличением количества ресторанов усиливается стремительно и конкуренция, что неизбежно повлечет за собой необходимость эффективного и рационального использования имеющихся ресурсов. В таких условиях, чтобы успешно вести свои дела и поддерживать требуемый уровень конкурентоспособности следует инвестировать в инструменты развития и поддержания бизнеса. Важным инструментом, развивающим ресторанный бизнес можно считать современную систему автоматизации ресторанов.

Вопросы построения автоматизированных систем управления для организаций, таких как кафе и рестораны, до настоящего момента не являлись предметом особого общего исследования. В литературе в основном рассматриваются общие проблемы построения систем документооборота и информационных систем в организациях подобного профиля, а также анализируются отдельные технологические разработки. Из широкого круга всевозможных публикаций, затрагивающих проблемы автоматизации процессов документооборота и управления наиболее интересными для этой тематики считаются опубликованные в последний период научно-методические разработки по проблемам создания систем корпоративного электронного документооборота. Эти работы содержат мнения об электронном документообороте и разнообразные компоненты системы по электронному обмену информацией, которые основаны на применении компьютерных информационных технологий. Изложены методические основы их введения, обобщен зарубежный и отечественный опыт постановки задач и организации решений в подобной сфере, которые требуют интеграции разнообразных информационных платформ и аппаратно-программных средств.

Представленная работа нацелена на изучение процессов, которые происходят внутри ресторана, обнаружить, так называемые, «слабые» места в структуре организации, функционирования учреждения и заострить на них внимание. Предметом подробного анализа избрана деятельность, направленная на обслуживание клиентов, поскольку работа данного подразделения считается базой работы ресторана.

Обзор инструментов автоматизации ресторанного бизнеса

В процессе подготовки обзора программных средств автор пытался дать оценку качеству работы служб поддержки предприятий и скорости отклика на требования возможных клиентов.

R-Keeper (рис. 1) считается старейшим игроком на рынке автоматизации бизнеса ресторанной отрасли. Первая версия R-Keeper создана в 1992 году, сейчас система внедрена в 37 тысячах ресторанах по всему миру.



Рис. 1 – ПП R-Keeper

В систему включены различные модули (бэк-офис, предназначенный для контроля издержек и процессов, а также для непосредственной работы в ресторане - фронт-офис) и приложения.

Цена лицензии: от 77000 - 180000 рублей.

Тип установки: устанавливается система локально.

Минусы:

* сложный интерфейс
* высокая цена

Проект iiko (рис. 2) находится на втором месте в России по популярности. Систему можно охарактеризовать наличием большого количества различных модулей, подключаемых к специальному серверу и работающих через него. Располагает функциональностью фронт- и бэк-офиса.

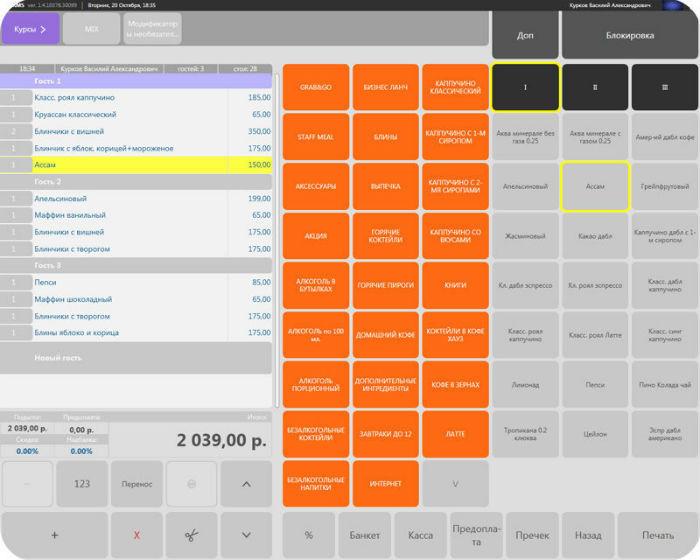


Рис. 2 - Проект iiko

Тип установки: локальный.

Цена: базовый вариант функциональности программы оценен в 23990 рублей за сервер, 19990 за автоматизацию кассовой станции (iikoFront) и 11990 за автоматизацию управления складом (iikoOffice).

Минусы:

* цена
* Значительное количество модулей, подключаемых за дополнительную плату, весьма сложно администрировать, что влечет повышение вероятности появления сбоев.

Jowi (рис. 3) является гибридным сервисом автоматизации, которым совмещаются локальные и облачный подходы — модуль Jowi устанавливается в ресторане, а потом данные синхронизируются на удаленные серверы. Этот факт предоставляет возможность сохранения работоспособности системы даже в случае «падении» интернета.



Рис. 3 – ПП Jowi

Система считается модульной — располагает приложениями для официантов, менеджеров зала которыми синхронизируются между собой данные (официантом принят заказ, повар узнал, какое блюдо необходимо готовить, менеджеру отправляется информация о продолжительности приготовления и др.). По статистике основателя сервиса программу используют 3500 заведений в государствах СНГ. Весьма гибкая система, к примеру, она обеспечивает возможность изменения различных данных даже задним числом — такая необходимость в «реальном мире» может появляться довольно часто.

Тип установки: локальный + SaaS.

Цена: $150 ежемесячно.

Poster (рис. 4) представляет собой систему автоматизации с облачным сохранением данных, функционирующая на планшетах Android и iPad. Рабочее место официанта или кассира работает на планшете, на принтере Epson TM-T20 печатаются чеки. На веб-сайте презентован отличный FAQ по работе с продуктом, существует поддержка в чате.



Рис. 4 – ПП Poster

Вид установки: локальная + SaaS.

Цена: $24-$79 ежемесячно.

Quick Resto (рис. 5) считается облачной системой автоматизации, позволяющая управлять рассадкой гостей, меню, принимать плату и для печати чеков настраивать принтеры. Имеется небольшой недостаток (скорее всего гипотетически) — отсутствует поддержка Windows, исключительно Android и iOS. Чтобы подключить периферийные устройства необходимо приобрести QR Box. Меню блюд в этом приложении изображены с картинками — красиво, однако в процессе работы отвлекает официантов.

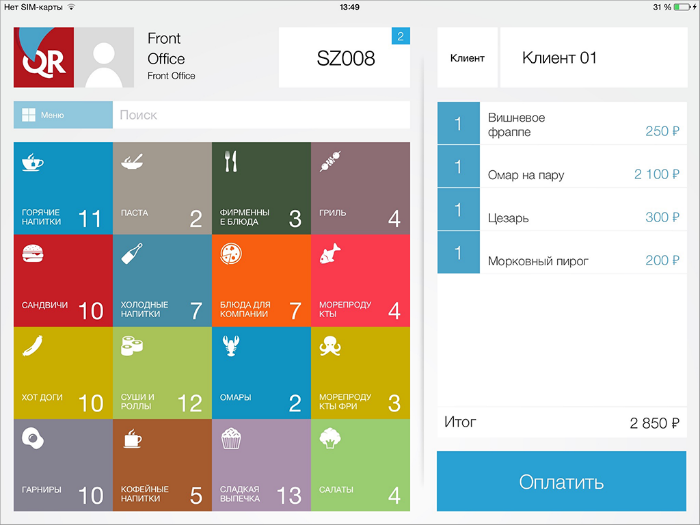


Рис. 5 - Quick Resto

Вид установки: Saas + Mobile.

Цена: от 2990 рублей ежемесячно (1 терминал).

iSOK POS STORE (рис. 6) – система POS, первоначально создавалась для магазинов, однако ее можно использовать и в небольшом кафе или ресторане.



Рис. 6 – Система iSOK POS STORE

Способна функционировать и в одиночном режиме (1 устройство — исключительно iPad), и в сдвоенном, если одно из мобильных устройств считается главным, а второе — клиентским. Осуществляется синхронизация посредством интернета, т.е. модель SaaS отсутствует.

Приложение бесплатное, но необходимо платить за ряд нужных функций, таких как: настройки кассы и принтера. Не самая гибкая система — к примеру, отсутствует возможность вводить данные «задним числом». Длительность хранения данных 2 - 40 дней на торговых устройствах.

Тип установки: Mobile.

Цена: бесплатная с оплатой за дополнительные функции.

Intellect Style (рис. 7) считается элементарной, устанавливаемой локально, программой для автоматизации небольших заведений и гостиниц. Явно перегружен интерфейс модуля официанта — к примеру, имеется большое количество функций, которые активируются при работе со счетом, однако они могут отвлекать внимание официанта.

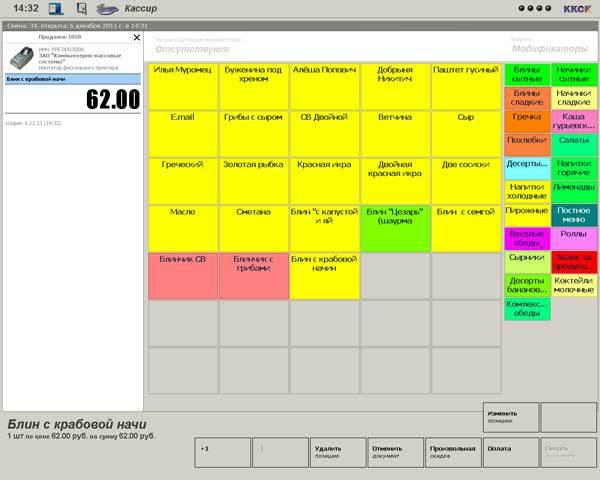


Рис. 7 – Система Intellect Style

Вид установки: локальная.

Цена: от 14900 руб.

Pos Sector (рис. 8). На веб-сайте написано, что пользуются системой свыше 4,5 тысяч. Локальный модуль с интуитивно понятным и простым интерфейсом — в нем формируются склады, отделы, блюда и подобные сущности. Располагает также браузерным приложением для официантов.



Рис. 8 – Система Pos Sector

Вид установки: локальная.

Цена: базовая лицензия оценена в $590 (за 1 компьютер).

Составлена таблица, в которой сравниваются выше описанные системы – таблица 1. Используется пятибальная система для оценки критериев, где 0 считается плохо, а 5 отлично.

Проведенный анализ рынка показывает, что несмотря на большое количество систем цена их довольно высока, поскольку они считаются сетевыми и приобретаются отдельными лицензиями, дополнениями и модулями. Все системы обладают обширным функционалом, но электронное меню присутствует только у системы R-KEEPER.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Критерии | Информационные системы для ресторана | | | | | | | |
| R-Keeper | iiko | Jowi | Poster | Quick Resto | iSOK POS STORE | Intellect Style | Pos Sector |
| Подсистемы  (официант, повар, бармен, кассир, менеджер) | 5 | 4 | 4 | 5 | 5 | 3 | 2 | 4 |
| Функциональные возможности | 5 | 4 | 4 | 4 | 3 | 2 | 4 | 4 |
| Количество ресторанов | сеть | сеть | сеть | сеть | сеть | 1 | сеть | сеть |
| Связь с офисом | 5 | 2 | 2 | 2 | 5 | 4 | 2 | 4 |
| Настройки в работе | 5 | 3 | 3 | 4 | 5 | 4 | 3 | 4 |
| Простота освоения | 2 | 2 | 4 | 3 | 2 | 3 | 3 | 3 |
| Покупка по частям | возможна | возможна | нет | нет | нет | нет | нет | возможна |
| Конечная цена | 2 | 2 | 4 | 3 | 3 | 5 | 2 | 2 |

Таблица 1 - Таблица сравнения существующих систем

Разрабатываемые средства обязаны обеспечить:

1. разделение меню информационного обеспечения на модули: повар, шеф-повар, кладовщик, менеджер, администратор;
2. создание заказа при помощи электронного меню;
3. доступ к рецептам напитков и блюд;
4. расчет остатка порций;
5. учет свободных и занятых мест в ресторане;
6. учет всего ассортимента реализуемых блюд: цена, вес, ингредиенты, название;
7. передачу заказа на кассу для расчета посетителя;
8. передачу заказа в бар и на кухню;
9. получение сведений о наличии продуктов для приготовления блюд со склада.

Основная цель разрабатываемых средств состоит в упорядочивании и структурировании информационных потоков кафе или ресторана, которое направлено на увеличение скорости обслуживания клиента.

Постановка задачи

Целью работы является повышение эффективности работы ресторана на основе разработки информационной системы, позволяющей автоматизировать процесс создания заказа, его передачи на кухню и менеджеру зала.

Формулировка поставленной задачи

Для достижения заданной цели необходимо выполнить ряд задач:

* провести анализ и моделирование работы ресторана
* провести выбор инструментов для создания программы поддержки работы ресторана;
* разработать базу данных программы и провести анализ требований работы сервисов;
* провести описание алгоритмов реализации программы поддержки работы ресторана и работы пользователей в системе;
* провести тестирование и оценку экономической эффективности разработанного программного продукта.

Функционал программного продукта

С помощью данной информационной системы, клиент может:

* сделать заказ через электронное меню на столе
* зарезервировать стол на вечер (с выбором блюд или выбрать по приходу в ресторан)
* заказать доставку на дом

Серверная часть для персонала:

* повар сможет просматривать текущие заказы, калькуляционные карты и технологию приготовления блюда
* Кладовщик сможет добавлять поступившие продукты на склад
* Шеф-повар сможет редактировать калькуляционные карты, рецепты и описание блюда, просматривать список заказов
* Менеджер сможет редактировать заказы и их статусы, подтверждать резерв столов и доставку
* Администратор сможет добавлять/удалять/изменять профили пользователей, будет иметь все доступы остальных пользователей.

Глава 1. Анализ предметной области предприятия общественного питания

1.1. Анализ работы ресторана

В качестве объекта исследования был выбран – ресторан. Рассматриваемое предприятие общественного питания предоставляет посетителям на выбор широкий список блюд и напитков, а также возможность культурного отдыха.

В состав ресторана входят такие подразделения: «Финансы и Документооборот», «Главный зал», «Кухня». На рис. 9 представлена структура предприятия.

Финансы и документооборот

Главный зал

Кухня

Рис. 9– Структура предприятия

Отдельно рассмотрим состав сотрудников и функции, которые выполняет каждый из отделов. Данная работа нацелена на автоматизацию процессов создания и исполнения заказов, поэтому отдел «Финансы и документооборот» не будет рассмотрен в исследовании.

* + 1. **Анализ отдела «Кухня»**

Отдел «Кухня» представляют 3-4 повара и один администратор горячего цеха. За приготовление блюд отвечают повара. Администратор кухни принимает заказы от официантов, анализирует их и выстраивает последовательность приготовления блюд, когда блюдо приготовлено администратор извещает об этом официанта. Также администратором горячего цеха ведется отчетность об израсходованных продуктах, контролируется их количество и при необходимости создается заявка на заказ продуктов. Когда продукты поступили, администратор их анализирует и сравнивает их с заявкой. Отчетные документы им передаются в отдел «Финансы и документооборот».

* + 1. **Анализ отдела «Главный зал»**

Отдел «Главный зал» представляют 3-4 официанта и один менеджер зала. Основная задача отдела заключается в обслуживании посетителей. Процесс взаимодействия с посетителями можно представить таким образом: в ресторан приходит посетитель и на основе меню создает свой заказ, официантом заказ отправляется менеджеру зала и на кухню, а после приготовления заказ подается посетителю. Если клиент собрался покинуть заведение, официант предоставляет ему счет, который предварительно подготовил менеджер зала, клиент его оплачивает. Заказ клиента переносится в общий журнал заказов.

* + 1. **Матрица организационных проекций**

Если обобщить описанное выше, то функции, которые выполняют сотрудники разных отделов ресторана можно представить, как матрицу организационных проекций. В таблице 2 представлена матрица организационных проекций.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Оргзвенья | | Функции | | | | |
| Отдел | Сотрудники | Обслуживание клиентов | Расчет посетителя | Управление деятельностью кухни | Формирование меню | Приготовление блюда |
| Кухня | Повара |  |  |  |  | Х |
| Администратор горячего цеха |  |  | Х | X |  |
| Главный зал | Официанты | X |  |  |  |  |
| Менеджер зала |  | Х |  |  |  |

Таблица 2 – Матрица организационных проекций

Проанализировав предоставленные данные о деятельности предприятия, его отделов и сотрудников, следует отметить, что большая часть функций осуществляется вручную, при этом требуется ведение бумажной документации. Поэтому возникает необходимость информационной компьютерной поддержки основных функций ресторана.

1.2. Моделирование работы ресторана

Моделируя работу ресторана, можно выделить как выходную, так и входную информацию, кроме этого необходимо учитывать и другие факторы, которые влияют на работу предприятия – это техническое обеспечение, правила приготовления блюд, законодательство и прочие факторы [10,17].

Анализируя деятельность учреждения общественного питания, необходимо выделить три основные работы, которые входят в состав предприятия (рис. 10).

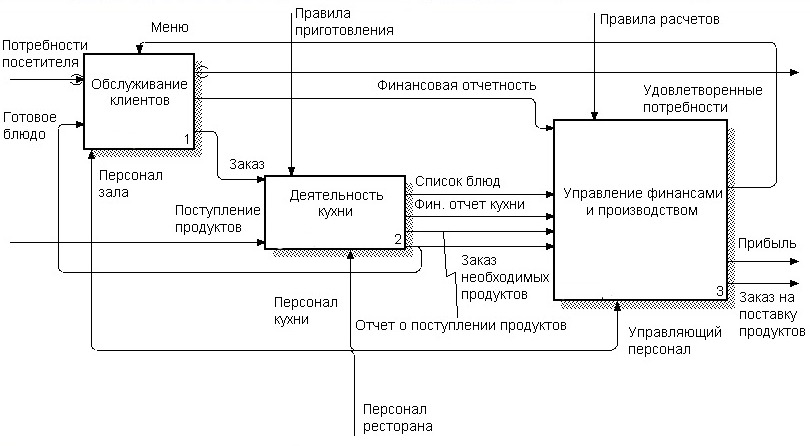


Рис. 10 – Деятельность ресторана

Они представляют собой:

* обслуживание клиентов предприятия;
* работу кухни, которая отвечает за приготовление блюд;
* управление финансами и производством, отвечающими за управление финансами в ресторане, создание ежедневного меню и управление закупкой продуктов.

Предметом изучения деятельности ресторана считается процесс обслуживания посетителей. Чтобы лучше осмыслить логику такого процесса, необходимо декомпозировать деятельность по обслуживанию посетителей на две:

* обслуживание столика;
* расчет клиента.

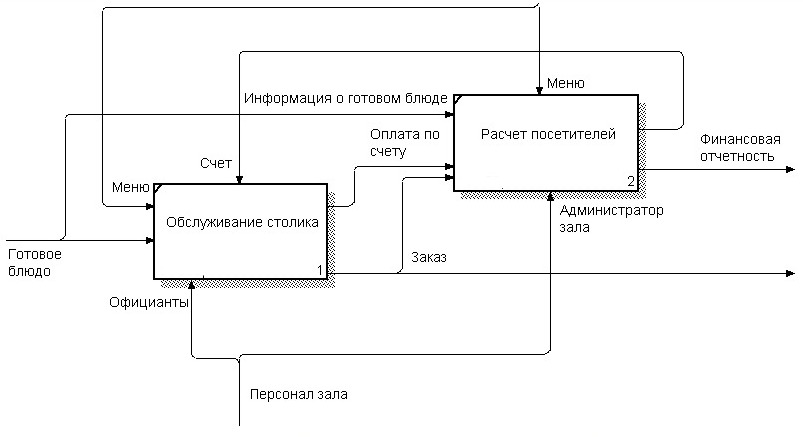


Рис. 11 – Декомпозиция «Обслуживание посетителей»

Диаграмма показывает, что работа официанта состоит в обслуживание столика включающая в себя:

* уборка столика после клиента;
* подача меню новому клиенту;
* принятие заказа;
* передача заказа на кухню для начала исполнения заказа и администратору зала для создания счета;
* подача готового заказа клиенту и, при необходимости, принятие нового заказа;
* подача счета и его оплата.

Работа менеджера зала включает в себя:

* формирование счета на оплату;
* контроль корректности подачи заказа официантом;
* подтверждение резервирования столика;
* подтверждение заказа на доставку.

Для выделения потоков данных следует создать диаграмму DFD. На диаграмме выделены две важнейшие внешние сущности – посетители и поставщики.

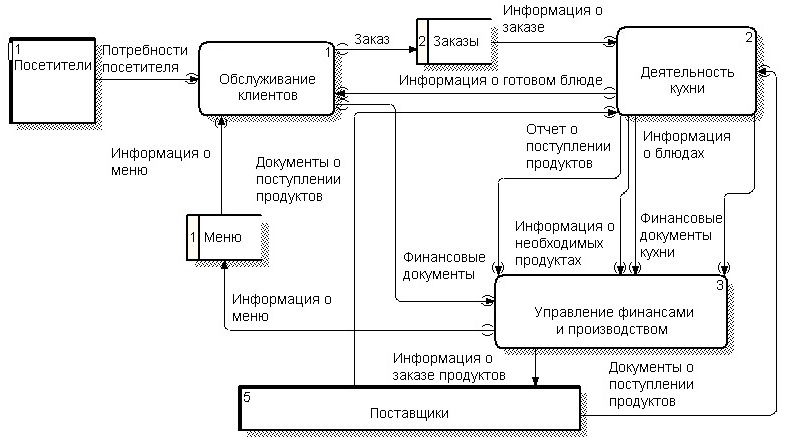


Рис. 12 – Диаграмма DFD. Деятельность ресторана

Предприятием ведется активная деятельность и осуществляется повседневный контакт с посетителями и поставщиками. Ежедневно от клиентов поступают заказы, а с поставщиками осуществляется обмен документаций о заказах и поступлениях товара. С посетителями обмен информацией осуществляется в процессе обслуживания посетителей, когда принимаются от них заказы на блюда. Поставщики же непосредственно контактируют с отделом производством и управления финансами, а также с кухней в процессе поставки продуктов.

Особенное внимание следует уделить исходящей и поступающей информации в процессе обслуживании посетителей.

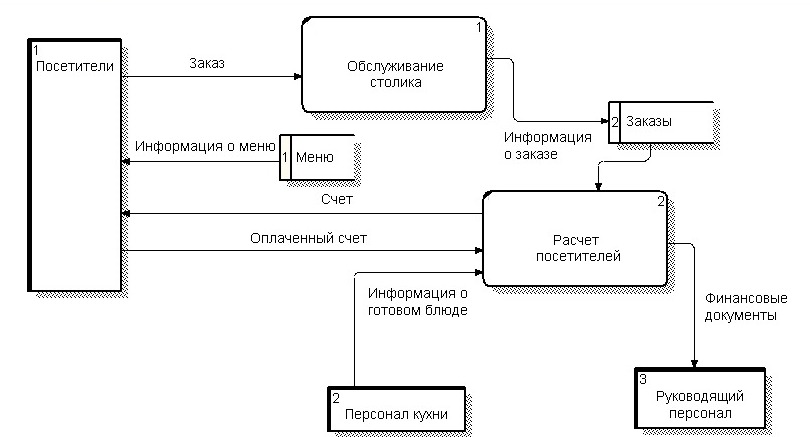


Рис. 13 – Диаграмма DFD. Обслуживание посетителей

Клиент знакомится с информацией из меню и на ее основе создает свой заказ. От сотрудника кухни для расчета клиента поступает информация о готовности блюда. Затем клиенту направляется документ “Счет”. Клиентом счет оплачивается и передается официанту. Администратором зала на основе проплаченного счета формируется журнал заказов и в конце смены передается руководящему составу.

**Глава 2. Проектирование программы поддержки работы ресторана**

**2.1. Определение целей и функций работы программного продукта**

Успешная реализация производственного процесса находится в зависимости от правильной организации и оперативного планирования деятельности учреждений общественного питания. Суть оперативного планирования состоит в организации производственной программы ресторана. За планирование производственной программы отвечают работники бухгалтерии, бригадиры, начальники производственных цехов, заведующие производством (заместители).

В ресторане оперативное планирование состоит из следующих компонентов:

1. Распределение сырья между бригадами и цехами.
2. Расчет необходимости в продуктах для приготовления блюд, которые предусмотрены планом - меню.
3. Формирование планового меню на неделю и разработка на его основе меню - плана, которое отражает дневную программу ресторана.
4. Производственную программу составляют на основании:
5. Создание технологических карт.
6. Расчета сырья, которое необходимо для приготовления данных блюд.
7. Составления плана - меню.
8. Определения числа блюд продаваемых за день.
9. Графика расчета посетителей и загрузки торгового зала.

Чтобы решить перечисленные задачи на предприятии в наши дни довольно широко применяется передовая компьютерная техника. Фактически степень автоматизации ряда служб ресторана (планово-финансовый отдел, отдел кадров, бухгалтерия) находится на значительном уровне.

Главной стадией оперативного планирования считается формирование плана–меню. В нем указываются номера рецептур, количество и наименования блюд. Основные факторы, которые следует учесть при формировании плана–меню: приблизительный ассортимент изготовляемой продукции, которые рекомендован для общественного питания в зависимости от его вида и типа предоставленного рациона, наличие и сезонность сырья. Закуски и блюда, которые включены в план–меню, обязаны быть разнообразными по видам сырьевой продукции, а также и по методам тепловой обработки. Кроме этого необходимо учитывать квалифицированный состав сотрудников, мощность и оснащенность производства торгово-технологическим оборудованием и трудоемкость блюд. Заведующий производством и директор, утверждая план-меню, отвечают за то, чтобы блюда, включенные в меню, были в реализации на протяжении всего дня предприятия.

План-меню составляют специалисты отдела планирования. Основу плана-меню составляют уже созданные заказы (в среднем, степень обеспеченности предприятия предварительными заказами находится сейчас в пределах 71-76%, что зависит от сезона), а также анализ статистики заказов, который позволяет раскрыть сезонные закономерности предпочтений и посещаемости посетителей. Для отдела планирования в качестве источников оперативной информации может служить информация, которая поступает от менеджеров ресторана, контактирующих с постоянными и корпоративными клиентами.

Поэтому сейчас ресторану необходима достаточно простая и эффективная информационная система, которая позволит решать такие задачи:

- гарантировать составление плана-меню по блюдам;

- гарантировать учет ингредиентов блюд для заказа у поставщиков;

- гарантировать учет клиентов;

- гарантировать заказа столиков и блюд онлайн как в ресторане, так и на доставку.

Применяемый программный комплекс, при этом обязан быть весьма простым в эксплуатации, чтобы им могли пользоваться и неподготовленные работники, располагающие только основными навыками работы с вычислительной техникой.

Чтобы решить поставленную ранее задачу, в первую очередь необходимо располагать четкой концепцией информационных потоков в ресторане, которая связана с регистрацией и обслуживанием заказов, учетом ресурсов, требуемых для реализации плана-меню. Проанализируем общий алгоритм работы менеджеров ресторана по обслуживанию плана-меню, практически никак не зависящий от того, в какой виде собираются, и осуществляется обработка информационных данных. На рис. 17 представлена блок-схема алгоритма.

Самыми уязвимыми местами в этой схеме работы считаются:

1. Самым слабым местом считается процесс заказа недостающих позиций у поставщиков. Главная проблема заключается в том, что формирование исчерпывающего списка необходимых для реализации плана-меню товарных позиций выполняется менеджерами вручную с применением технологических карт ресторана, на что затрачивается много времени и при этом могут появиться различные неточности и ошибки. Осуществлять такие действия наиболее оперативно при имеющихся условиях невозможно, так как не имеется эффективного механизма составления заказов поставщикам товаров.
2. Выставление бухгалтерией счета заказчику осуществляется по заявке оперативного менеджера ведущего заказ. Подобная заявка может поступить в бухгалтерию лишь после того, как менеджер удостовериться в возможности выполнения заказа (товар имеется на складе или заказан и в ближайшее время он поступит на склад). Оперативный менеджер о поступлении на склад товара узнает от менеджера склада и формирует заявку в бухгалтерию. Так как подобная связь между отправкой счета и движением товара по складу считается опосредованной, то могут быть значительные задержки в отправке счетов посетителям.



Рис. 17 - Алгоритм работы по составлению и реализации плана-меню в ресторане

1. Проверка наличия товара на складе оперативным менеджером. На сегодняшний день, после составления заявки, оперативный менеджер, который отвечает за поступивший заказ, должен самостоятельно обратиться на склад и менеджер склада, применяя имеющиеся исключительно в его распоряжении данные по складу, делает заключение о возможности незамедлительного выполнения заказа. В данной ситуации интервал между созданием заказа и решением о продолжительности его выполнения, которое получил менеджер, отвечающий за заказ, от менеджера склада, может по времени быть весьма длительным, что в наше время уже считается неприемлемым.

На рис. видно, что имеющуюся систему работы можно считать недостаточно эффективной. Она может значительно задержать или даже вообще сорвать реализацию плана-меню ресторана. Обеспечение доступности информации в реальном времени и автоматизация работы для всех заинтересованных исполнителей обязано значительно увеличить производительность работы предприятия. Необходимо создать систему, которая позволит повысить эффективность обслуживания клиента посредством электронного меню чем сократится время на передачу заказа на кухню и менеджеру зала для формирования счета. Таким образом, в процессе разработки системы важнейшими задачами автоматизации считаются:

1. Создание программного модуля, который обеспечит автоматизированное составление плана-заказа на поставку товаров на склад ресторана, учитывая задание плана-меню, а также сведения о товарных остатках на складе.
2. Формирование информационного хранилища данных, которое содержит данные по движению товарных позиций на складе ресторана.
3. Формирование информационного хранилища данных, которое содержит все планы-меню ресторана с данными по их реализации, как с позиции осуществления производственной программы, так и с позиции реализации блюд посетителям ресторана.
4. Формирование информационного хранилища данных, которое содержит все данные по клиентам ресторана.
5. Создание функций бронирования столика и заказа блюд в режиме онлайн как в самом ресторане, так и на доставку с последующим изменением статуса заказа.

**2.2. Анализ и выбор инструментов для создания программы поддержки работы ресторана**

Для корректной работы информационной системы необходимо провести анализ программных средств, имеющихся на рынке, оценить их производительность, надежность и доступность [14].

При выборе языка программирования следует провести анализ существующих решений. Для этого выберем наиболее часто используемые языки и проведем сравнение.

Perl (Practical Extraction and Reporting Language) - один из наиболее популярных и мощных языков программирования [21].

Perl –это сложный полнофункциональный язык. Он интересен тем, что заполняет разрыв между С-приложениями и способами программирования командного процессора Unix, обладая функциональностью первых и простотой последних.

Perl - интерпретируемый язык, который оптимизирован для сканирования произвольных текстовых файлов, извлечения информации из таких файлов и печати отчетов, применяя эту информацию.

PYTHON представляет собой скриптовый язык программирования, в основном используемый в области многообразных Интернет-приложений.

Синтаксис главных конструкций PYTHON аналогичен языку программирования C++. Это довольно молодой язык, который заменил Perl. PYTHON прост в изучении, ориентирован в основном на web-программирование, располагает большим количеством многообразных подключаемых модулей, которые расширяют его использование. Основной целью использования PYTHON считается создание динамического HTML, который дает возможность отображения различного контекста в приложении, находясь в зависимости от действий пользователей [22,23].

Преимуществами PYTHON в сравнении с языком Perl является простой синтаксис, что делает его легче в изучении, менее раздутый и более четкий исходный код, что делает его легче при разработке модулей.

Недостатком PYTHON 3 является его низкая в сравнении с Perl производительность на сложных приложениях при обработке больших скриптов, т. е. когда сайт состоит из нескольких страниц, но с длинным кодом. Рациональнее использовать в этих случаях CGI. По этой причине при разработке новой версии PYTHON 3.6 уделялось основное внимание повышению быстроты работы. Затрагивались также вопросы безопасности, и внедрялась поддержка сессий. Новейшая версия PYTHON располагает ядром Zend Engine, которое позволяет увеличить стабильность и производительность за счет наиболее качественной поддержки модулей.

ASP.NET – часть технологии NET, которая используется для написания мощных клиент-серверных интернет приложений. Технология дает возможность создания динамических страниц HTML. Эта технология появилась за счет объединения NET Framework и старой технологии ASP. Технология располагает множеством готовых элементов управления, применение которых дает возможность быстрого создания интерактивных web-сайтов. Также есть возможность применять сервисы, которые предоставляются остальными сайтами, прозрачно для пользователей. Собственной разработки [28].

Сравнение основных языков веб-программирования приведено в таблице 3 и на рис. 18.

В качестве языка разработки клиента подключения к базе данных был выбран язык PYTHON.

Python - это язык программирования для динамической генерации HTML кода со стороны сервера. В нём имеется встроенная поддержка базы данных MySQL, что дает возможность считать выбранную связку MySQL-PYTHON наиболее оптимальной. PYTHON-скрипты интерпретируется и выполняются на сервере.

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Баллы | | |
| Python | ASP | Perl |
| Возможность компиляции | 100 | 100 | 50 |
| Многопоточная компиляция | 100 | 0 | 100 |
| Интерпретатор командной строки | 75 | 50 | 0 |
| Многомерные массивы | 100 | 0 | 100 |
| Динамические массивы | 100 | 100 | 0 |
| Ассоциативные массивы | 50 | 0 | 0 |
| Интерфейсы | 100 | 0 | 50 |
| Мультиметоды | 100 | 0 | 0 |

Таблица 3 - Сравнение языков программирования

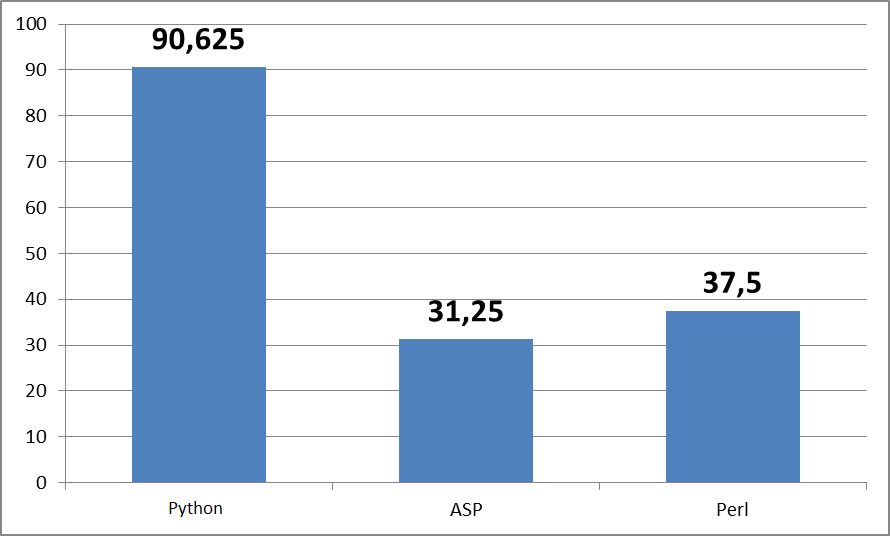


Рис. 18 - Диаграмма результатов сравнения языков программирования

Предпочтение PYTHON было отдано по следующим характеристикам:

* В процессе выбора языка разработки альтернативой PYTHON был язык ASP(Active Server Pages), схожий по структуре но построенный на технологии СОМ.
* В модулях PYTHON все запускается в области памяти, выделенной программе операционной системой. ASP загружает для различных действий соответствующие СОМ-модули, чем сильно загружает оперативную память и процессор.
* Интеграция PYTHON с выбранной СУБД MySQL значительно более полная, чем у ASP. Существует большое количество утилит на PYTHON для работы с базами данных MySQL, где реализуется набор свойств наиболее полный по сравнению с остальными базами данных. Есть очень полезные встроенные функции, которые недоступны для остальных баз данных. Одним из значительных преимуществ PYTHON является поддержка широкого круга баз данных: Oracle, Microsoft SQL server, MySQL и другие [9].
* Несомненное достоинство PYTHON - это отсутствие временных проблем с исправлением внутренних ошибок, что позволяет оперативно реагировать и исправлять недоработки.

Для выбора СУБД так же проведем анализ существующих решений. Для этого выберем наиболее часто используемые языки и проведем сравнение.

В общем случае, набор требований к СУБД, в зависимости от установленных целей может изменяться. Критерии можно поделить на несколько групп: требования к рабочей среде; производительность; особенности разработки приложений; функциональные возможности; структура данных [8,20].

Рассмотрим в качестве альтернатив такие СУБД: 1) PostgreSQL, 2) MySQL; 3) Microsoft SQL Server; 4) Oracle; 5) DB2.

Соответственно, был проведен анализ выше представленных критерий и распределили таким образом: MySQL (0.32), DB2 (0.28), MS SQL Server (0.16), Oracle (0.13), PostgreSQL (0.11) (рис. 19).

Рис. 19 - Результаты анализа СУБД

На основании данного сравнения выбираем для использования СУБД MySQL.

Принимая во внимание функциональность и задачи проекта, принято решение осуществить реализацию веб-сервиса с применением следующих технологий: HTML; CSS; JavaScript; PYTHON; MySQL [1,25,29].

**2.3. Разработка базы данных программы**

Построим инфологическую модель «сущность-связь» предметной области [2,7,18,19].

С учетом особенностей построения были разработаны 11 объектов данных (сущностей):

* Ингредиенты;
* Состав блюд;
* Пользователь;
* Блюда;
* Заказы блюд;
* Статус заказа;
* Заказы;
* Бронирование;
* Столы;
* Статус доставки;
* Доставка.

Инфологическая модель для проектируемой системы представлена на рис. 20.

Между сущностями установлены следующие связи – таблица 4. [1,3,6]

| Сущность 1 | Сущность 2 | Вид связи |
| --- | --- | --- |
| Ингредиенты | Состав блюд | Один-ко-Многим |
| Блюда | Состав блюд | Один-ко-Многим |
| Заказы блюд | Блюда | Многие-к-Одному |
| Заказы | Заказы блюд | Один-ко-Многим |
| Заказы | Статус заказа | Многие-к-Одному |
| Столы | Заказы | Один-ко-Многим |
| Бронирование | Заказы | Многие-к-Одному |
| Бронирование | Столы | Многие-к-Одному |
| Статус доставки | Доставка | Один-ко-Многим |
| Доставка | Заказы | Многие-к-Одному |

Таблица 4 – Связи между сущностями

На основании построенной модели была проведена нормализация и разработана схема базы данных системы - рис. 21 [5,12,16]. Так же на рис. отображены типы и длина полей соответствующих полей.



Рис. 20 - Инфологическая модель для проектируемой системы



Рис. 21 – Структура базы данных разрабатываемой системы

Значений свойств таблиц представлено в таблицах 5-14.

|  |  |
| --- | --- |
| Название | Описание |
| Orders\_Id | уникальный идентификатор |
| Type | тип заказа делается для того чтобы было понятно как отдавать блюдо (на доставку или в зал) |
| Created\_at | дата создания заказа |
| Date | дата исполнения заказа |
| Status | статус заказа показывающий состояние заказа. Внешний ключ на таблицу order\_status |
| Table\_id | внешний ключ на таблицу table, может быть null если это доставка |

Таблица 5 – Таблица Orders

|  |  |
| --- | --- |
| Название | Описание |
| Order\_status\_Id | уникальный идентификатор |
| Name | ждет подтверждения, подтвержден, готовится, готов |

Таблица 6 – Таблица Order\_status

|  |  |
| --- | --- |
| Название | Описание |
| Order\_dishes\_Id | уникальный идентификатор |
| Order\_id | ссылка на номер заказа |
| Dish\_id | блюда в заказе, ссылка на таблицу dish |
| Count | количество блюд в заказе |

Таблица 7 – Таблица Order\_dishes: хранит блюда в заказе

|  |  |
| --- | --- |
| Название | Описание |
| Dishes\_Id | уникальный идентификатор |
| Name | наименование блюда |
| Price | цена блюда |
| Description | описание |
| Weight | вес блюда |
| Photo\_url | фото блюда |

Таблица 8 – Таблица Dishes

|  |  |
| --- | --- |
| Название | Описание |
| Tables\_Id | уникальный идентификатор стола |
| Person\_count | кол-во посадочных мест за столом |
| Photo\_url | фотографии стола |
| Updated\_at | время обновления статуса стола |
| Table\_status | статус стола (зарезервирован с определенного времени, занят, свободен) |

Таблица 9 – Таблица Tables

|  |  |
| --- | --- |
| Название | Описание |
| Dishes\_ingredients\_Id | уникальный идентификатор |
| Dish\_id | внешний ключ на наименование блюда |
| Ingredient\_id | внешний ключ на наименование ингридиента |
| quantity | количество блюд которые возможно приготовить исходя из ингридиентов |

Таблица 10 – Таблица Dishes\_ingredients: состав блюда

|  |  |
| --- | --- |
| Название | Описание |
| Id | уникальный идентификатор |
| Name | наименование ингридиента |
| Quantity | кол-во ингридиента на складе |

Таблица 11 – Таблица Ingredients

|  |  |
| --- | --- |
| Название | Описание |
| Reservation\_Id | уникальный идентификатор |
| Table\_id | номер стола, внешний ключ на таблицу tables |
| Name | имя гостя |
| Second\_name | фамилия гостя |
| Phone | телефон |
| Order\_id | номер заказа, внешний ключ на таблицу Orders |
| Created\_at | дата создания заказа |
| Reservation\_date | время и дата прихода гостя |

Таблица 12 – Таблица Reservation: резерв столов

|  |  |
| --- | --- |
| Название | Описание |
| Delivery\_Id | уникальный идентификатор |
| Address | адрес доставки |
| Name | Имя |
| Second\_name | фамилия |
| Created\_at | дата создания заказа |
| Order\_id | номер заказа, внешний ключ на таблицу Orders |
| Date | дата и время доставки заказа |
| Status | статус заказа, внешний ключ на таблицу delivery\_status |

Таблица 13 – Таблица Delivery

|  |  |
| --- | --- |
| Название | Описание |
| Delivery\_status\_Id | уникальный идентификатор |
| Name | ожидает подтверждения, подтвержден, готовится к доставке, в пути, доставлено |

Таблица 14 – Таблица Delivery\_status

**2.4. Анализ требований работы сервисов**

В разрабатываемой системе использованы следующие виды пользователей:

- Менеджер – должен быть предоставлен сервис доступа к таблицам reservation, delivery, order, order\_dishes;

- Администратор - взаимодействует со всеми таблицами на чтение и запись;

- Повар - предоставлен доступ к таблицам orders, order\_dishes, dishes, dishes\_ingridients, ingridients;

- Шеф-повар – предоставлен доступ к таблицам orders, dishes, dishes\_ingredients, ingredients.

- Кладовщик доступ к таблице ingredients.

Поварам предоставляются текущие заказы на исполнение: id заказа, блюда в заказе и тип заказа(доставка или в зал)

* возможность начать исполнение заказа
* изменить статус заказа на «готовится», «готов»
* возможность посмотреть состав блюда.

Для менеджера предоставлены возможности:

* просмотр и изменение текущих заявок на доставку (имя, фамилия, когда создана заявка, текущий статус);
* просмотр и изменение заказа (возможность изменения блюд в заказе)
* подтвердить заказ;
* просмотр и изменение резерва стола

Шеф-повару предоставлены возможности:

* просмотр текущих заказов.
* просмотр/изменение блюда в меню
  + просмотр/изменение калькуляционной карты
  + просмотр/изменение рецепта блюда

Администратор имеет все доступы шеф-повара и менеджера, а также предоставляется возможность работы со списком пользователей (изменение данных, заведение новых), может просматривать/изменять/удалять статусы заказов доставок и бронирования.

Клиенту предоставляется функционал:

- Раздел электронного меню (просмотр списка блюд);

- Информация о блюде

- Фото изображение

- Название

- Цена

- состав

- Сортировка блюд по параметрам

- по названию,

- по катеогрии,

- по цене

- Фильтрация

- по категориям

- по наличию

- по цене

- Полнотекстовый поиск блюд

- Возможность оформления электронного заказа:

- формирование списка блюд

- указание количества по каждому из блюд

- уточнение ингредиентов

- Оформление доставки заказа

- указание личных данных пользователя (телефона/имени)

- указание адреса доставки

- указание времени доставки

- Оформление резерва стола

- указание телефона/имени

- выбор стола;

- указание времени доставки

Глава 3. Описание реализации программного продукта

3.1. Описание основных алгоритмов реализации программы поддержки работы ресторана

Для вывода информации в счете и отчетах необходимо провести ряд расчетов.

Для расчета позиций счета на оплату необходимо рассчитать суммы заказанных блюд по каждой строке заказа, а затем рассчитать общую сумму заказа. Алгоритм расчета для формирования счета приведен на Рис. 22.



Рис. 22 - Алгоритм расчета для формирования счета

Для формирования отчета о выработке за смену вводится дата, на которую формируется отчет. Затем проводится последовательный перебор всех записей полученных заказов. Если дата записи совпадает с заданной датой, то счетчик количества заказов увеличивается на единицу, а счетчик суммы – на сумму строки заказа. По завершении расчетов числа, содержащиеся в счетчиках, выводятся в отчет в качестве итогового результата.

Алгоритм расчета для формирования отчета о выработке за смену приведен на Рис. 23.



Рис. 23 - Алгоритм расчета для формирования отчета о выработке за смену

Для формирования отчета о выработке за смену одного официанта вводится дата, на которую формируется отчет и ФИО официанта. Затем проводится последовательный перебор всех записей полученных заказов. Если дата записи совпадает с заданной датой, а ФИО официанта, указанная в заказе с заданными ФИО официанта то счетчик количества заказов увеличивается на единицу, а счетчик суммы – на сумму строки заказа. По завершении расчетов числа, содержащиеся в счетчиках, выводятся в отчет в качестве итогового результата.

Алгоритм расчета для формирования отчета о выработке за смену одного официанта приведен на Рис. 24.



Рис. 24 - Алгоритм расчета для формирования отчета о выработке за смену одного официанта

3.2. Описание работы пользователей в системе

Рассмотрим описание пользовательской части системы.

В для входа в систему клиент набирает в сроке адреса браузера адрес системы и попадает на главную страницу системы (рис. 25).

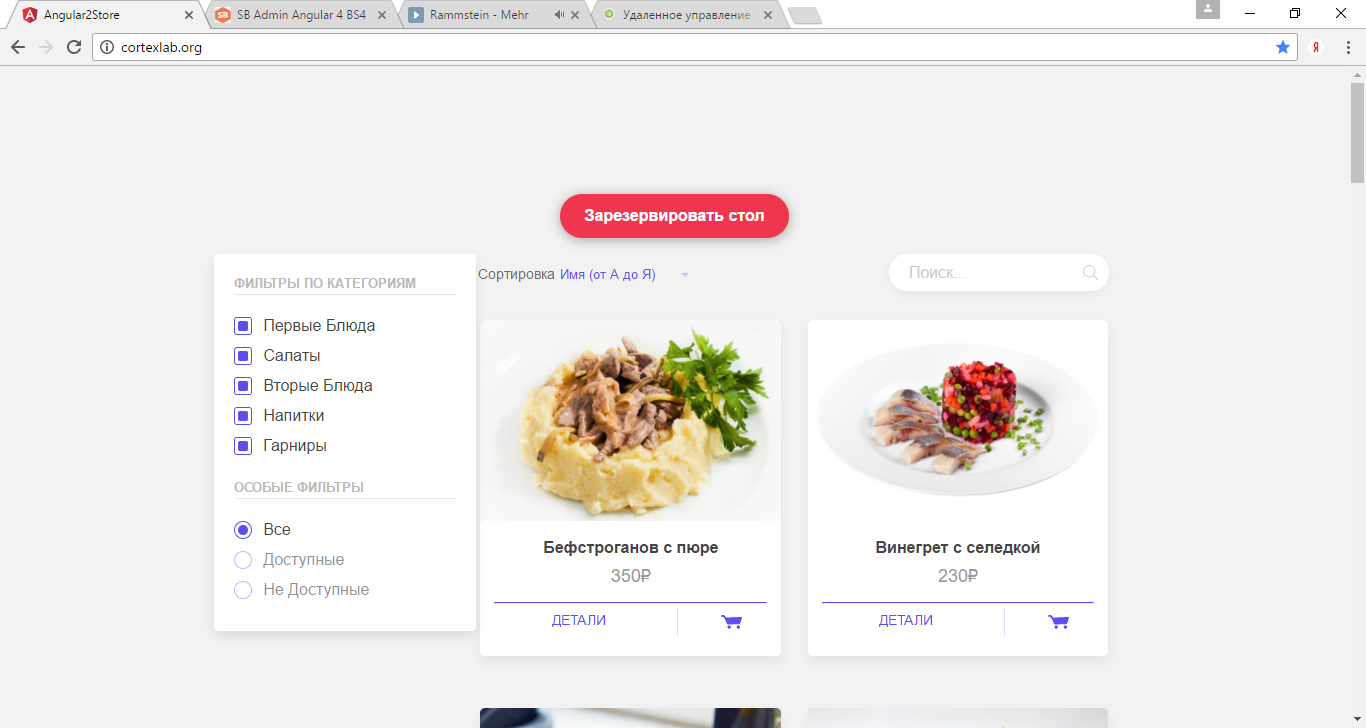


Рис. 25 – Главная страница системы

Страница содержит каталог товаров ресторана. На странице реализованы фильтры по категориям блюд и фильтры по доступности заказа блюд, реализована сортировка и поиск товаров. Блюда можно добавить в корзину и оформить заказ на доставку как показано на рис. 26, также можно просмотреть детально информацию о блюде (рис. 27).

Как видно из рис. 27, вверху меню появляется значок корзины с обозначением количества добавленных товаров, при нажатии на который можно открыть меню корзины и оформить заказ – рис. 28.

При нажатии на кнопку «Оформить заказ», открывается меню оформления с выбором резервирования столика или доставки на дом выбранных блюд, как показано на рис. 29. После чего необходимо ввести все необходимые данные и подтвердить заказ, после чего он поступит к администратору системы.

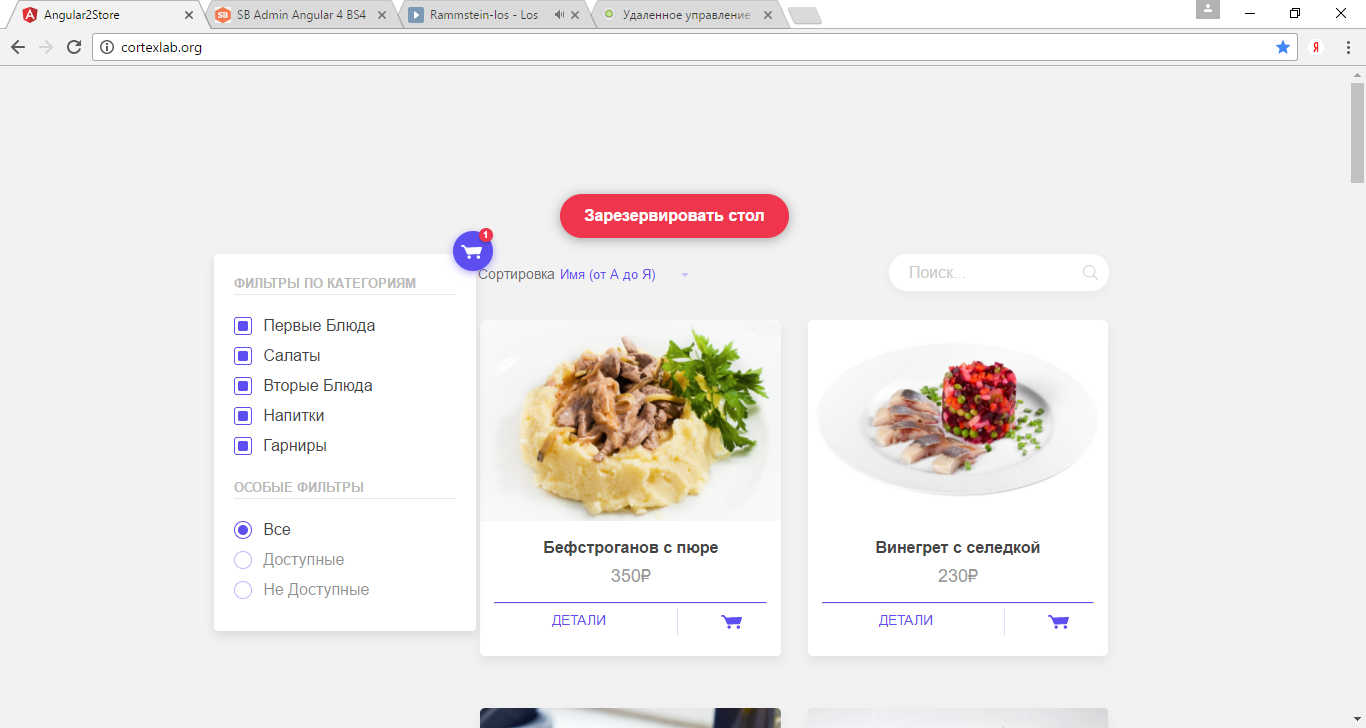


Рис. 26 - Добавление блюда в корзину

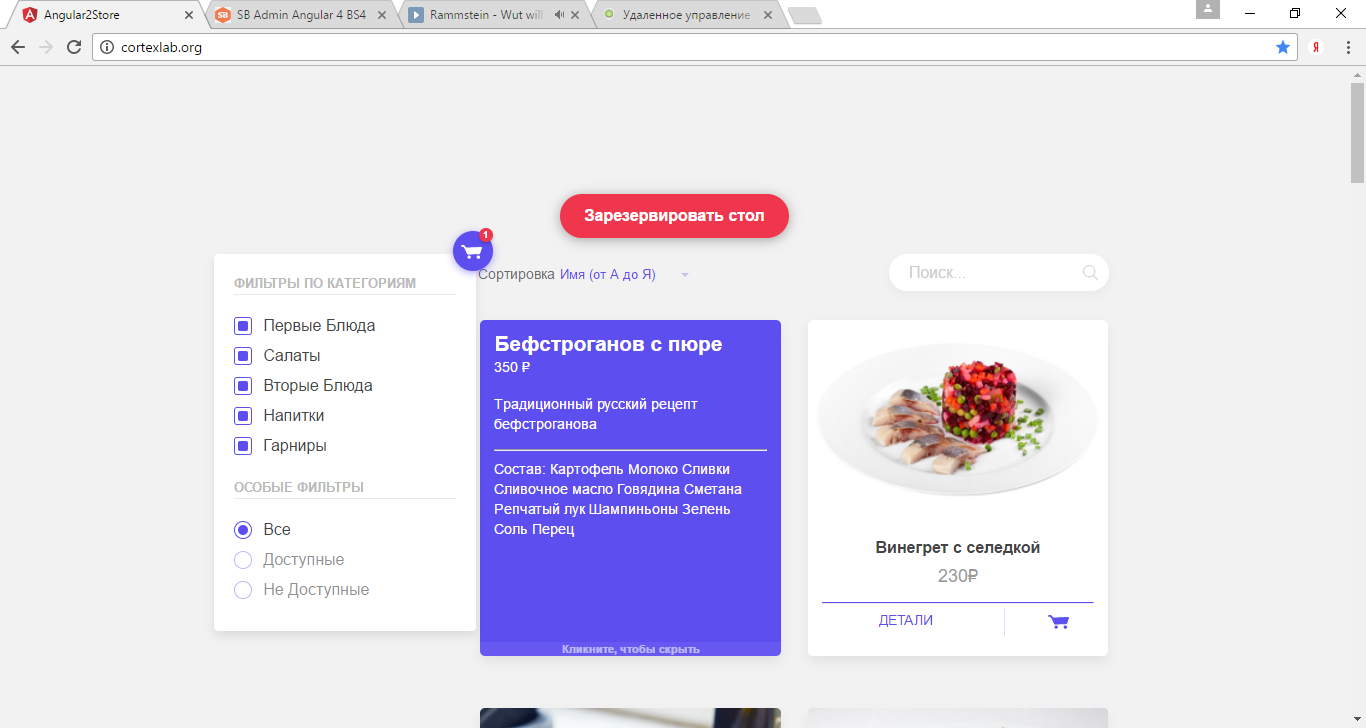


Рис. 27 – Просмотр описания товара

Как видно из главной страницы системы, на ней еще реализована ссылка резервирования столика. С помощью нее, клиент может выбрать и зарезервировать себе необходимый столик в ресторане как показано на рис. 30. После ввода контактных данных и подтверждения заказ поступит к администратору системы.

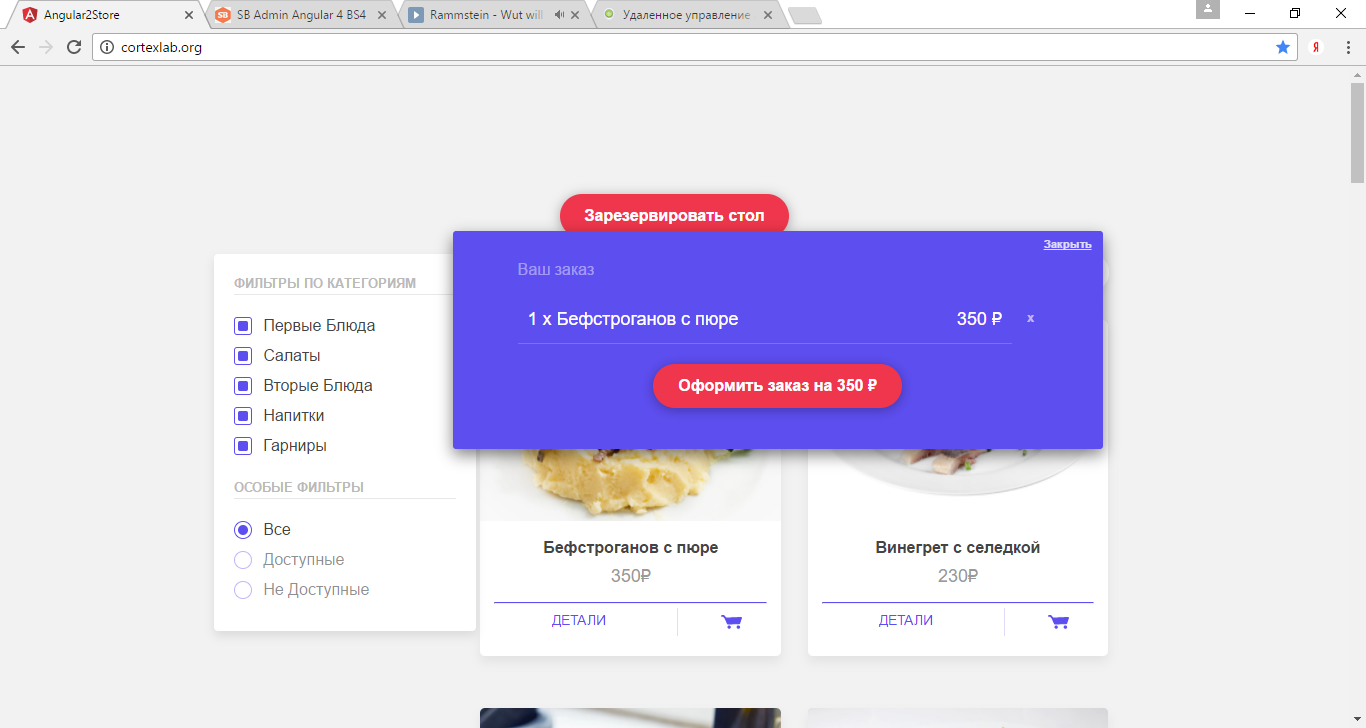


Рис. 28 – Всплывающее меню корзины

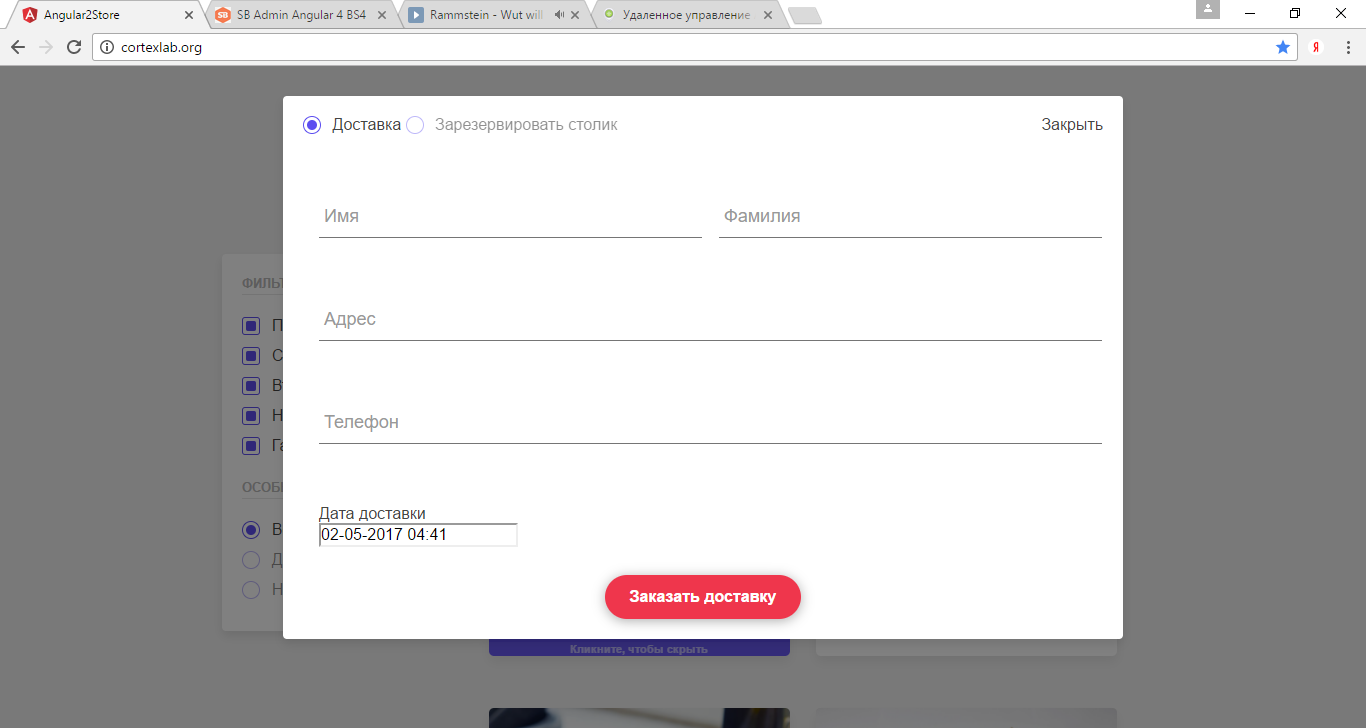


Рис. 29 – Оформление заказа на доставку

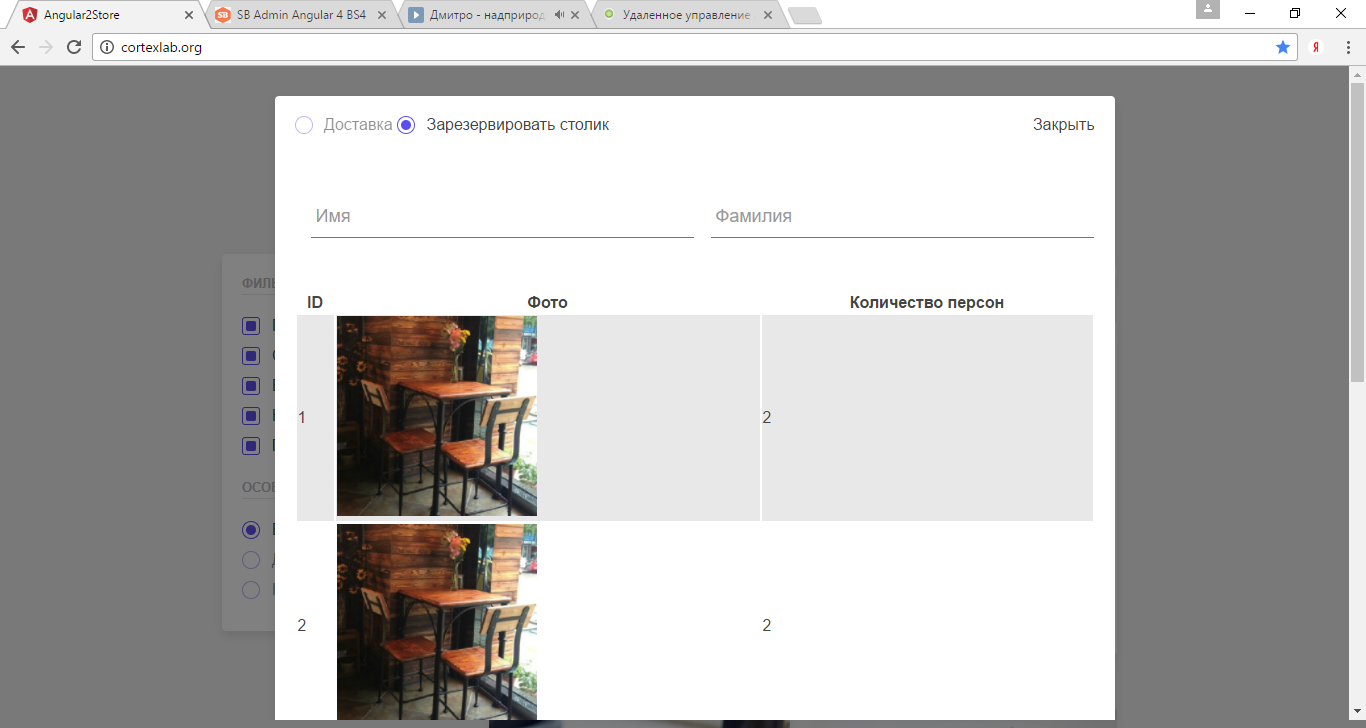


Рис. 30 – Резервирование столика

Рассмотрим административную часть системы.

Для входа в административную часть системы в строке адреса браузера необходимо ввести http://admin.cortexlab.org/ после чего пользователь попадает на страницу входа в систему как показано на рис. 31.

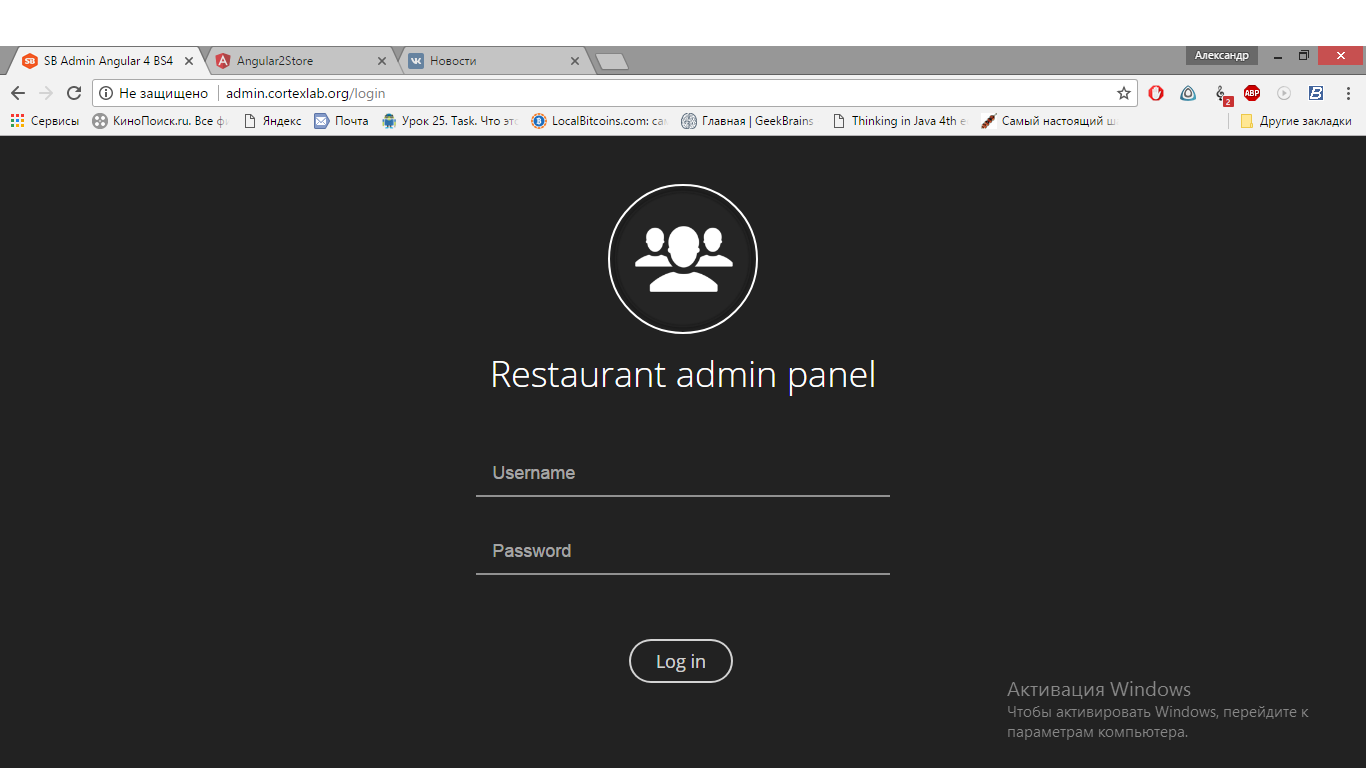


Рис. 31 – Страница входа в систему

Рассмотрим работу пользователей в системе на примере администратора. Как видно из рисунка главная страница содержит общую информацию о статистике (рис. 32).



Рис. 32 - Главная страница административной панели системы Административная панель, кроме предыдущей страницы содержит страницы:

- «Столы» - для добавления и изменения столиков ресторана (рис. 33);

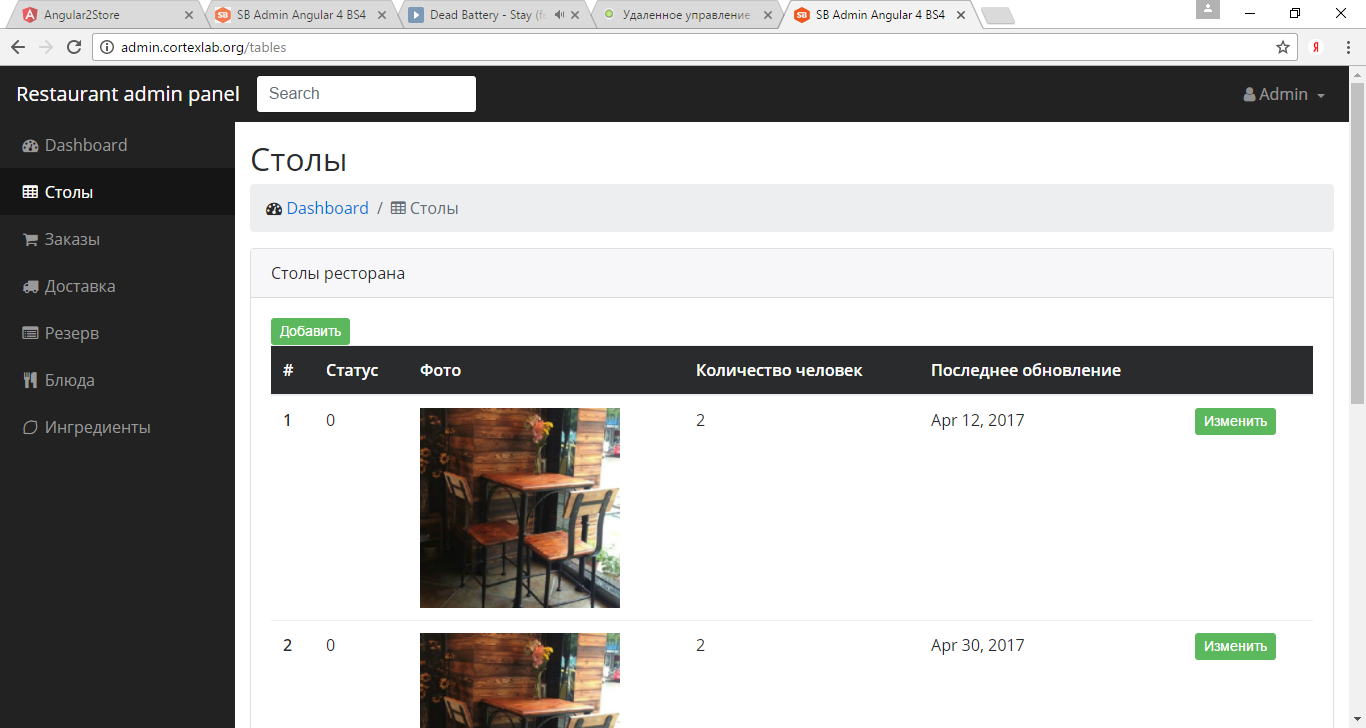


Рис. 33 – Страница «Столы»

- «Заказы», «Доставка», «Резерв» - для добавления, просмотра, обработки и изменения статусов заказов, информации о заказе в зале, доставку и резервирования столиков соответственно (рис.34-36);

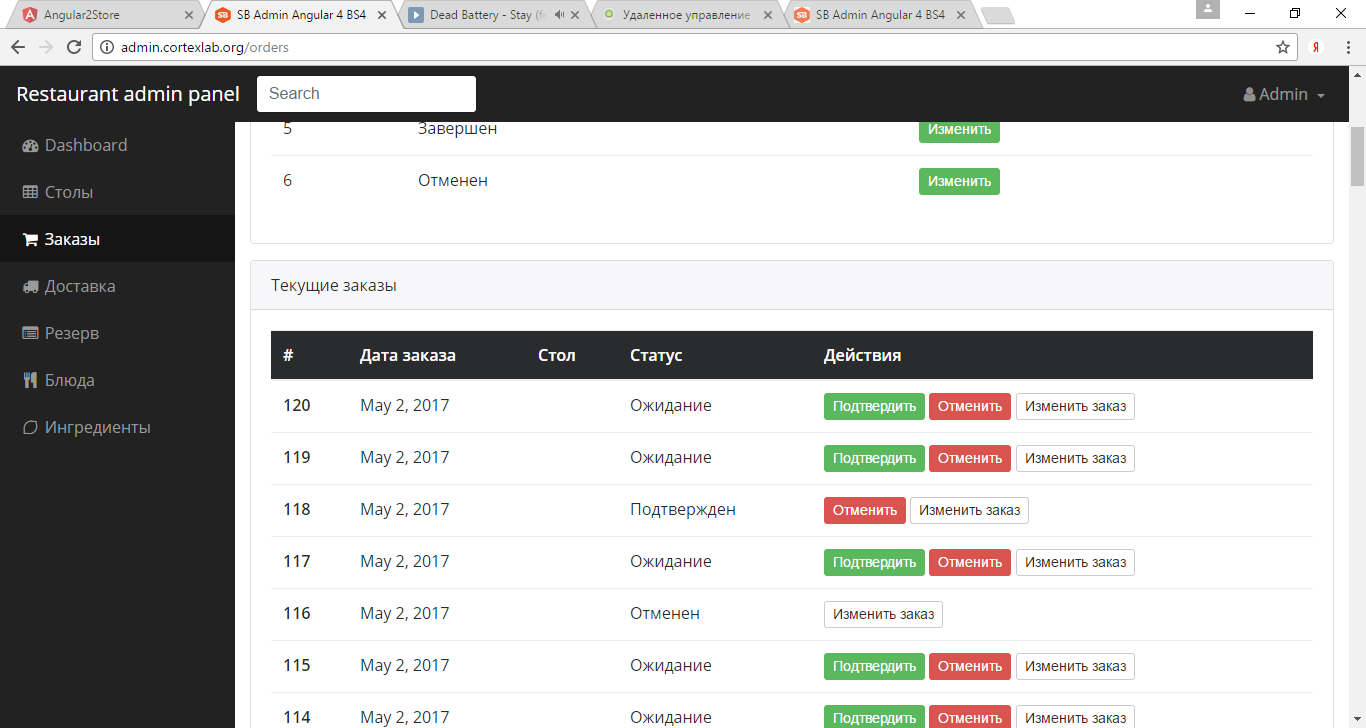


Рис. 34 – Страница «Заказы»

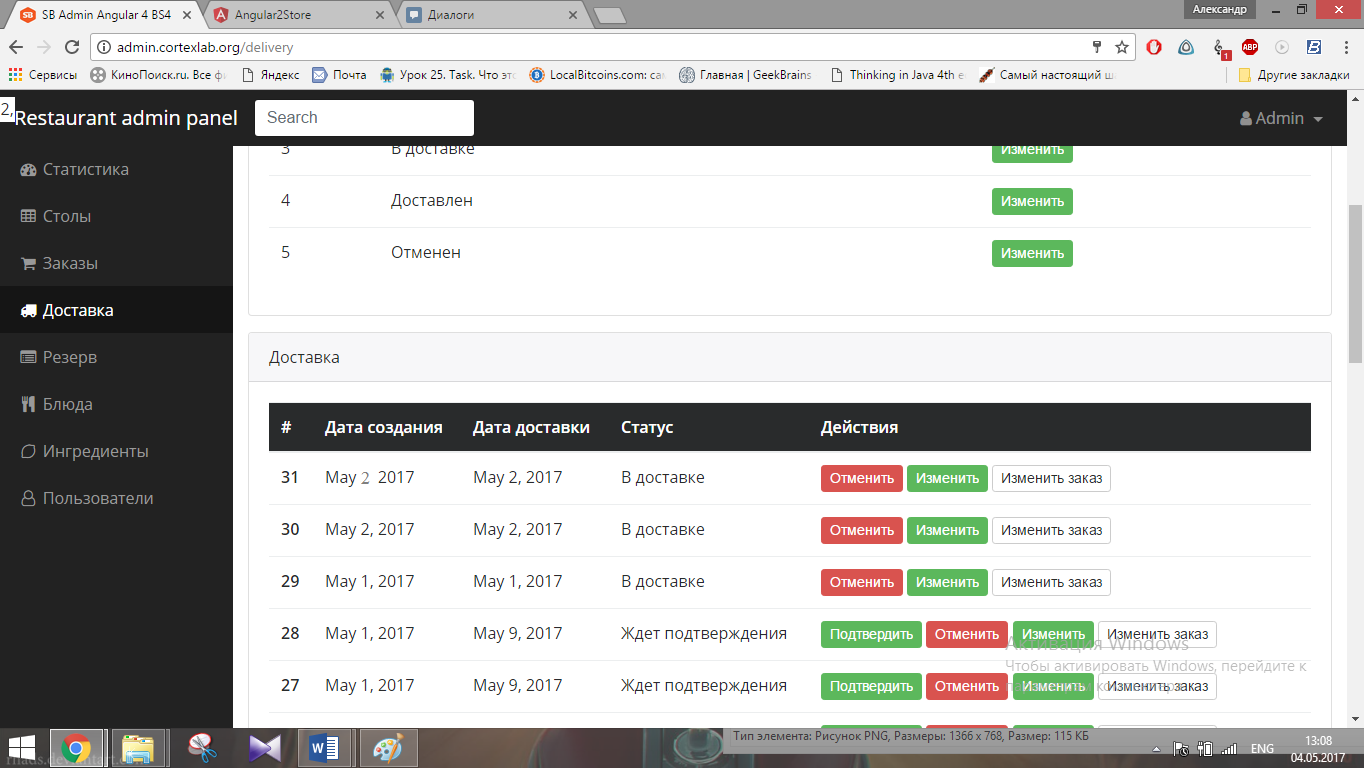


Рис. 35 – Страница «Доставка»

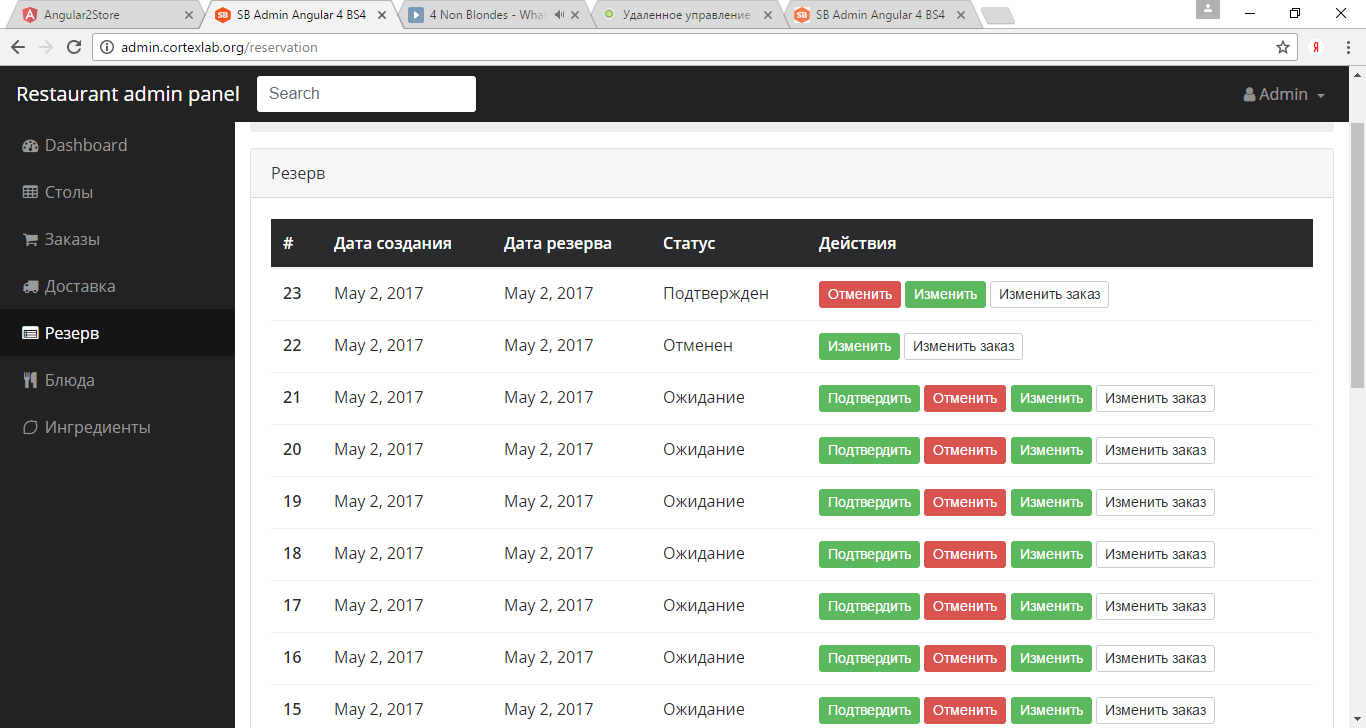


Рис. 36 – Страница «Резерв»

- «Блюда» - для добавления и изменения блюд ресторана, (рис. 37);

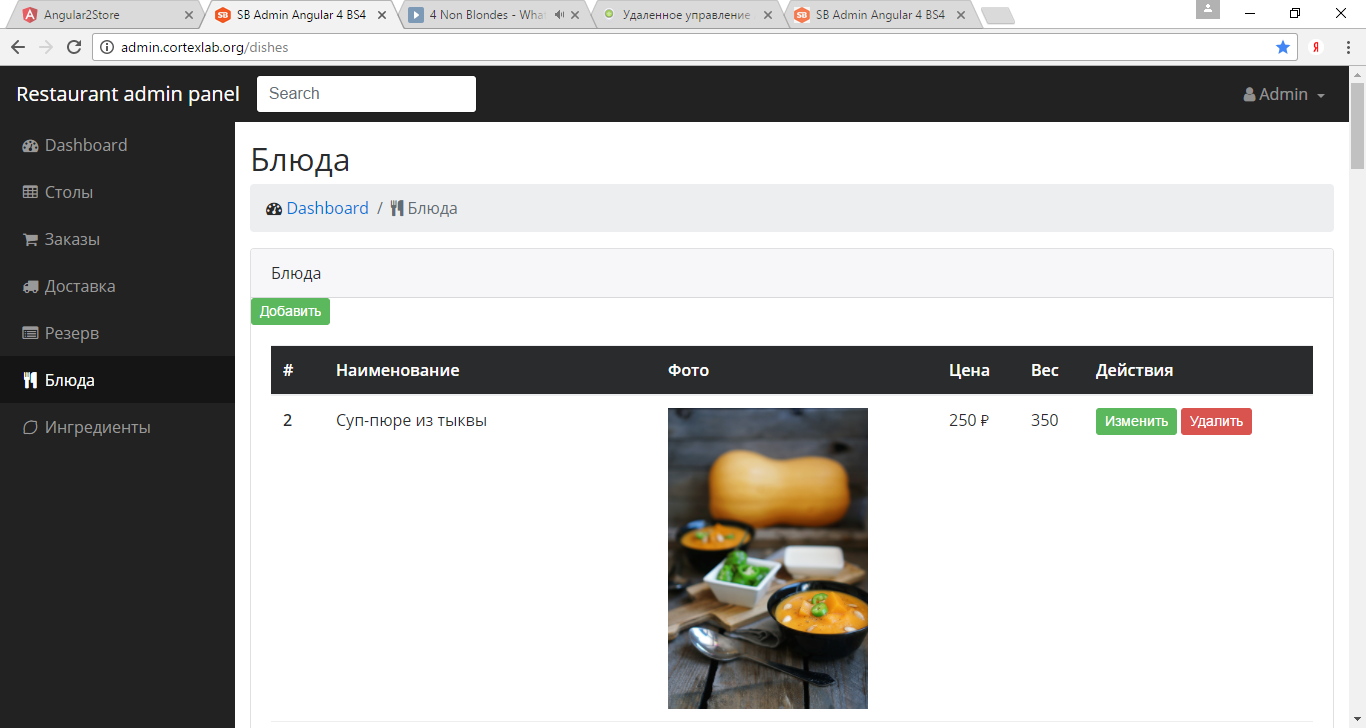


Рис. 37 – Страница «Блюда»

- «Ингредиенты» - необходима для учета ингредиентов блюд на складе ресторана (рис. 38).

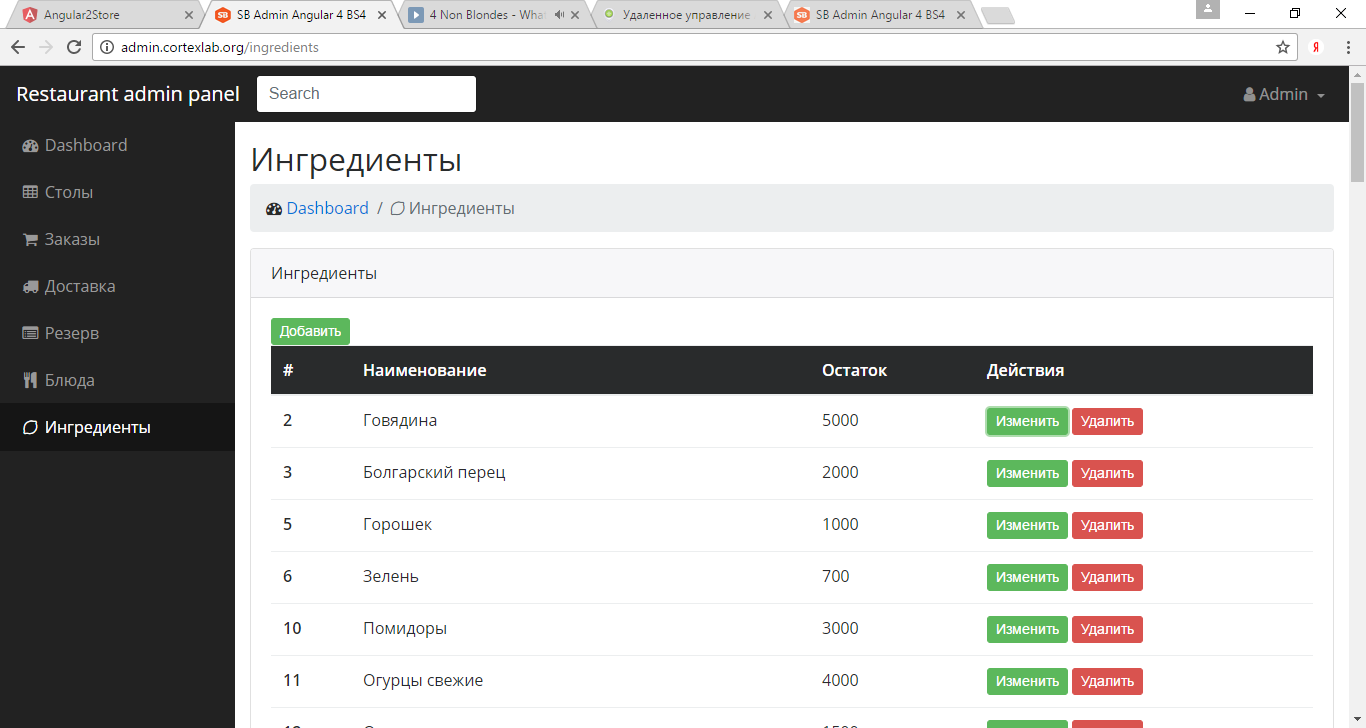


Рис. 38 – Страница «Ингредиенты»

3.3. Тестирование разработанной системы

В наше время существует большое количество браузеров и устройств, с которых можно просмотреть Web-страниц в интернете. Невзирая, что разрабатываемая система в виде сайта и отвечает w3c спецификациям, нельзя гарантировать, что система будет отображаться на всех устройствах одинаково. Рынок браузеров и их часть по информации можно представить таким образом (таблица 15) [3,14].

На рис. 39 представлено наглядное изображение в виде круговой диаграммы рейтингов браузеров.

Для наглядности сравним рейтинг браузеров в РФ, поскольку система рассчитана на аудиторию, проживающих в нашей стране (рис. 40). Взяты статистические данные из 3 серверов LiveInternet, OpenStat, HotLog.

Как видно из рисунков, что данные не сильно отличаются друг от друга. В тройку лидеров, которые считаются по статистическим данным наиболее посещаемыми и популярными браузерами, входят Mozilla Firefox, Google Chrome, Opera. Ненамного от этих браузеров отстал Internet Explorer.

|  |  |  |
| --- | --- | --- |
| Рейтинг браузеров | Наименование браузера | Популярность, % |
| 1 место | GoogleChrome (самый популярный) | 61.00 |
| 2 место | MozillaFirefox | 23.60 |
| 3 место | Internet Explorer | 8.00 |
| 4 место | [Apple Safari](http://sd-company.su/article/browsers/safari) | 3.70 |
| 5 место | Opera | 1.60 |
|  | Прочие | 0.8 |

Таблица 15 – Популярность интернет браузеров по миру на 2016 год [24]

Рис. 39 - Рейтинг браузеров [24]

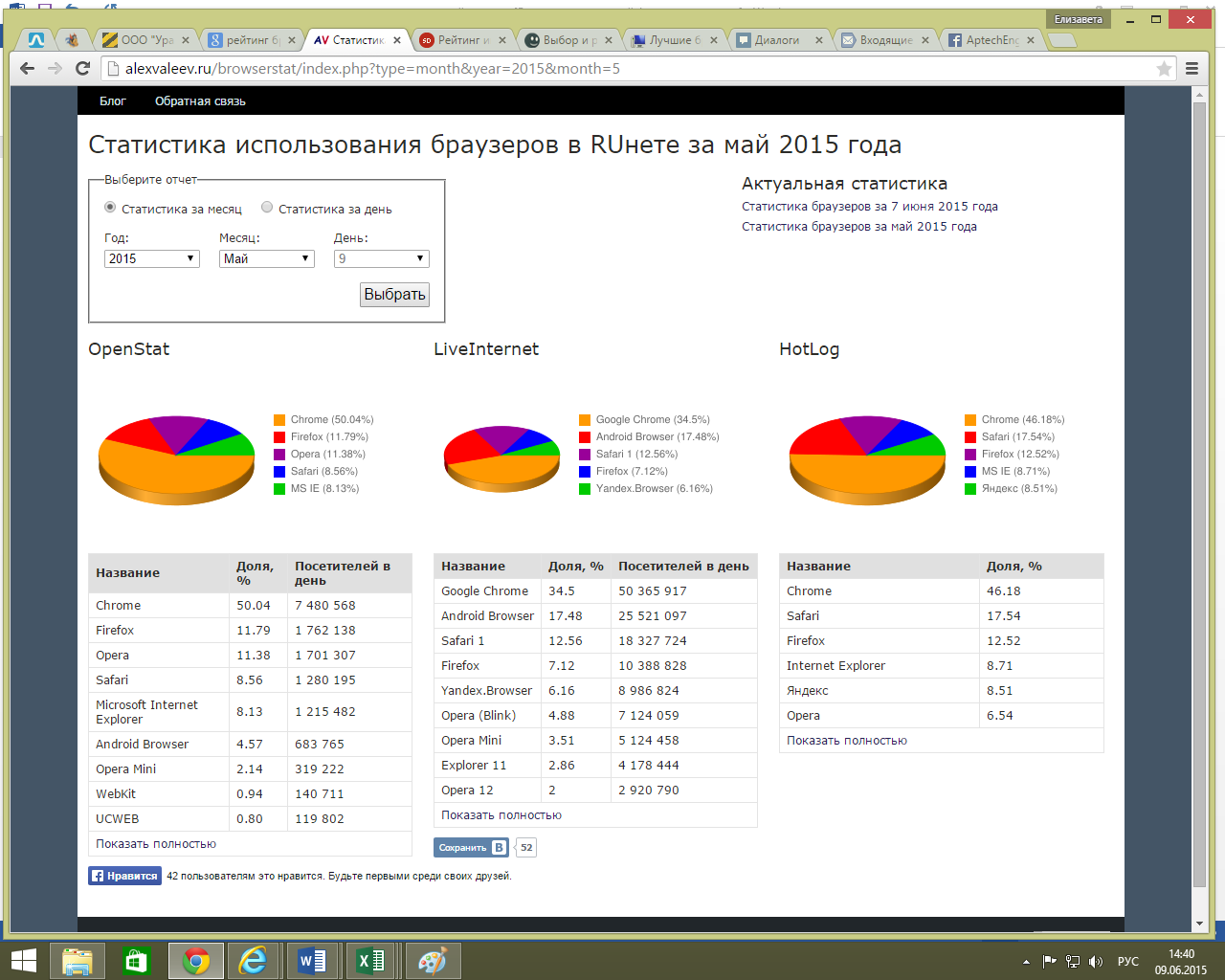


Рис. 40 - Рейтинг браузеров на территории России май 2016 года [27]

Из сказанного выше следует сделать вывод: чтобы систему посещало как можно больше пользователей, необходимо в первую очередь выполнить проверку отображения системы в трех лидирующих браузерах, так как вероятность того, что систему откроют собственно с этих браузеров, считается наиболее высокой.

Поэтому и тестирование на корректность отображения проводилось в данных браузерах. Процесс тестирования показал, что сайт корректно отображается во всех трех браузерах. Применение стандартизированных технологий без привязки к определенной ОС и специфичным технологиям, позволяет предположить, что данная система будет корректно функционировать в большинстве современных браузеров. Корректным отображением системы можно считать работу без ошибок, реализацию всех функций, идентичность отображения во всех браузерах [13].

Разработанная система соответствует всем требованиям. Тест можно считать пройденным.

Отображение системы во всех самых распространенных браузерах говорит о его грамотной и правильной разработке, пользователь увидит привлекательную и полноценную систему при его открытии, а не его фрагмент, что, в конечном результате, может привести к его незамедлительному закрытию.

Выводы

В результате выполнения работы разработана система программной поддержки предприятия общественного питания – ресторана.

По результатам выполнения решены следующие задачи в работе:

- проведена постановка задачи исследования информационных технологий в ресторанном бизнесе и определены основные критерии разработки программного обеспечения поддержки работы ресторана;

- проведено исследование аналогичных программных продуктов в сфере ресторанного бизнеса;

- проведен анализ и моделирование работы ресторана;

- определены цели и функции работы программного продукта, проведен выбор инструментов для создания программы поддержки работы ресторана;

- разработана база данных программы и проведен анализ структуры работы сервисов;

- проведено описание алгоритмов реализации программы поддержки работы ресторана и работы пользователей в системе;

- проведено тестирование программного продукта.

**Заключение**

Система позволит повысить эффективность обслуживания клиента посредством электронного меню, чем сократится время на передачу заказа на кухню и менеджеру зала для формирования счета. Таким образом, в результате внедрения системы, она повысит эффективность работы ресторана путем автоматизации процесса создания заказа, его передачи на кухню и менеджеру зала.

Разработанная система обеспечивает все функции, поставленные в постановке задачи выпускной работы, а именно:

1. разделение меню информационного обеспечения на модули: повар, шеф-повар, кладовщик, менеджер, администратор;
2. создание заказа при помощи электронного меню;
3. доступ к рецептам напитков и блюд;
4. расчет остатка порций;
5. учет свободных и занятых мест в ресторане;
6. учет всего ассортимента реализуемых блюд: цена, вес, ингредиенты, название;
7. передачу заказа на кассу для расчета посетителя;
8. передачу заказа в бар и на кухню;
9. получение сведений о наличии продуктов для приготовления блюд со склада.

Список цитируемой литературы

1. Аверченков В. И. Информационные системы в производстве и экономике: учебное пособие / В. И. Аверченков. – Изд-во: Флинта, 2011. – 274 с.
2. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем: учебник. – М.: Финансы и статистика, 2012.
3. Вигерс Карл. Разработка требований к программному обеспечению / Карл Вигерс. – М.: Издат.-торговый дом «Русская редакция», 2014. – 576с.
4. Виленский П. Л. Оценка эффективности инвестиционных проектов. Теория и практика. / П. Л. Виленский, В. Н. Лившиц, С. А. Смоляк – М.: Дело, 2014. – 888 с.
5. Винокуров М.А. и др. Автоматизация управленческого учета. / М.А. Винокуров и др. – М. ИНФРА-М, 2011. – 578 с.
6. Гвоздева, Т. В. Проектирование информационных систем: учебное пособие / Т. В. Гвоздева, Б. А. Баллод. – Ростов-н/Д: Феникс, 2011. – 508 c.
7. Исаев Г.Н. Информационные системы в экономике: учебник для студентов вузов. – М.: Издательство «Омега-Л», 2011. – 462 с.
8. Лобуренко Е. О. Школьный сайт: создание, наполнение и привлечение посетителей / Е. О. Лобуренко, Е. В. Якушина // Народное образование. – 2012. – № 4. – С. 171-181.
9. Плещев, В.В. Базы данных. VisualFoxPro, Access, SQLServer, Oracle с примерами и упражнениями [Текст]: Учебное пособие. - Екатеринбург: Изд-во Урал.гос. экон.ун-та, 2002.
10. Рузакова О.В. Теория проектирования информационных систем [Текст]: Учеб. пособие. – Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2003. – 70 с.
11. Скрипкин К. Г. Экономическая эффективность информационных систем. / К. Г. Скрипкин – ДМК Пресс, 2012. – 256 с.
12. Смирнова Г.Н. Проектирование экономических информационных систем: учебник. / Г.Н. Смирнова – М.: Финансы и статистика, 2011. – 578 с.
13. Уткин В.Б. Информационные системы в экономике: Учебник. / В.Б. Уткин, К.В. Балдин – М.: Дашков и К, 2012. – 394 с.
14. Экслер А. Б. Укрощение Интернета, или Самый полный и понятный самоучитель работы в Сети. / А. Б. Экслер – М.: НТ Пресс, 2013. – 944 с.
15. Ясенев В. Н. Информационные системы и технологии в экономике: учеб пособие для студентов вузов, обучающихся по специальностям экономики и управления./ В. Н. Ясенев – М.: ЮНИТИ-ДАНА, 20ёё. – 560 с.
16. Alcorn Wade, Frichot Christian, Orru Michele. The Browser Hacker's Handbook. - Wiley, 2014. - 648p.
17. Alexander Philip. Information Security: A Manager's Guide to Thwarting Data Thieves and Hackers. - Praeger Security International. Westport, CT, USA. 2008. 188 pages.
18. Barnett R.C., Grossman J. Web Application Defender's Cookbook: Battling Hackers and Protecting Users - John Wiley & Sons, Inc., 2013. — 552 p
19. Böhme R. (Ed.) The Economics of Information Security and Privacy - Springer, 2013. — 321 p
20. Большая техническая энциклопедия. Показатели нормативного коэффициента и эффективности: [Электронный ресурс]. – Режим доступа: <http://www.ai08.org/index.php/term/,9da4ab975b545ba09f5c525f56aea9589c56535c59649e61a86b5b63929da260666b53976b545ca25ba8975f549fa4549f58705f66a8a7a0a4525f575da89d62a25c586992acab57ae566e5c6da3a2629f94.xhtml> (Дата обращения 01.04.2017г.)
21. Документация и книги по программированию: [Электронный ресурс]. - Режим доступа: http://helloworld.ru/(Дата обращения 11.04.2017г.)
22. Костромин В.А. Конспект вебмастера [Электронный ресурс]. − Режим доступа: <http://website-ru.net/txt/basic/dinamic-vs-static.shtml> (Дата обращения 25.04.2017г.)
23. О чем говорят в сети. Блог создание сайтов: [Электронный ресурс]. – Режим доступа: <http://slovomeg.ru/sozdanie-sajtov/denver-nabor-dlya-veb-dzhentlmenov.html>(Дата обращения 12.04.2017г.)
24. Рейтинг браузеров по занимаемой ими доли на рынке: [Электронный ресурс]. – Режим доступа: <http://sd-company.su/article/browsers/rating-browsers-russia>(Дата обращения 3.04.2017г.)
25. Сео-блог о продвижении и создании сайтов: [Электронный ресурс]. − Режим доступа: <http://www.adeptseo.kz/metody_sozdanija_sajta>(Дата обращения 14.04.2017г.)
26. Справочник HTML: [Электронный ресурс]. − Режим доступа: <http://htmlbook.ru/>(Дата обращения 10.04.2017г.)
27. Статистика использования браузеров на территории РФ за 2015 год: [Электронный ресурс]. – Режим доступа: <http://alexvaleev.ru/browserstat/index.php?type=month&year=2015&month=1>(Дата обращения 12.04.2017г.)
28. Статистические данные расчета стоимости сайта [Электронный ресурс]. – Режим доступа: <http://www.cmsmagazine.ru/creators/price/?sk=_works&so=desc&st=1&m_f=&m_t=&c=1&cc=18&cms=121&cpp=50> (Дата обращения 02.04.2017г.)
29. Электронный блог: Этапы разработки качественного сайта: [Электронный ресурс]. − Режим доступа: <http://siteexpo.ru/blog/razrabotka-kachestvennogo-sajta>(Дата обращения 13.04.2017г.)
30. Электронный справочник. Основы HTML: [Электронный ресурс]. −Режим доступа: <http://on-line-teaching.com/html/lsn016.html>(Дата обращения 28.04.2017г.)

**Приложение**

**Schema.sql:**

DROP TABLE IF EXISTS `delivery`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `delivery` (

`delivery\_id` int(11) NOT NULL AUTO\_INCREMENT,

`address` varchar(255) DEFAULT NULL,

`name` varchar(255) DEFAULT NULL,

`second\_name` varchar(255) DEFAULT NULL,

`created\_at` datetime DEFAULT NULL,

`date` datetime DEFAULT NULL,

`status\_id` int(11) DEFAULT NULL,

`order\_id` int(11) NOT NULL,

`phone` varchar(255) DEFAULT NULL,

PRIMARY KEY (`delivery\_id`),

UNIQUE KEY `delivery\_id\_uindex` (`delivery\_id`),

KEY `delivery\_orders\_id\_fk` (`order\_id`),

KEY `delivery\_delivery\_status\_id\_fk` (`status\_id`),

CONSTRAINT `delivery\_delivery\_status\_id\_fk` FOREIGN KEY (`status\_id`) REFERENCES `delivery\_status` (`delivery\_status\_id`),

CONSTRAINT `delivery\_orders\_id\_fk` FOREIGN KEY (`order\_id`) REFERENCES `orders` (`orders\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=32 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `delivery\_status`

--

DROP TABLE IF EXISTS `delivery\_status`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `delivery\_status` (

`delivery\_status\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`delivery\_status\_id`),

UNIQUE KEY `delivery\_status\_id\_uindex` (`delivery\_status\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `dishes`

--

DROP TABLE IF EXISTS `dishes`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `dishes` (

`dishes\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) NOT NULL,

`price` int(11) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`weight` int(11) NOT NULL,

`photo\_url` varchar(255) DEFAULT NULL,

`category\_id` int(11) DEFAULT NULL,

`receipt` varchar(255) DEFAULT NULL,

PRIMARY KEY (`dishes\_id`),

UNIQUE KEY `dishes\_id\_uindex` (`dishes\_id`),

KEY `dishes\_dishes\_categories\_dishes\_categories\_id\_fk` (`category\_id`),

CONSTRAINT `dishes\_dishes\_categories\_dishes\_categories\_id\_fk` FOREIGN KEY (`category\_id`) REFERENCES `dishes\_categories` (`dishes\_categories\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=33 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `dishes\_categories`

--

DROP TABLE IF EXISTS `dishes\_categories`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `dishes\_categories` (

`dishes\_categories\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`dishes\_categories\_id`),

UNIQUE KEY `dishes\_categories\_dishes\_categories\_id\_uindex` (`dishes\_categories\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `dishes\_ingredients`

--

DROP TABLE IF EXISTS `dishes\_ingredients`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `dishes\_ingredients` (

`dishes\_ingredients\_id` int(11) NOT NULL AUTO\_INCREMENT,

`dish\_id` int(11) NOT NULL,

`ingredient\_id` int(11) NOT NULL,

`quantity` int(11) NOT NULL,

PRIMARY KEY (`dishes\_ingredients\_id`),

UNIQUE KEY `dishes\_ingredients\_id\_uindex` (`dishes\_ingredients\_id`),

KEY `dishes\_ingredients\_dishes\_id\_fk` (`dish\_id`),

KEY `dishes\_ingredients\_ingredients\_id\_fk` (`ingredient\_id`),

CONSTRAINT `dishes\_ingredients\_dishes\_id\_fk` FOREIGN KEY (`dish\_id`) REFERENCES `dishes` (`dishes\_id`),

CONSTRAINT `dishes\_ingredients\_ingredients\_id\_fk` FOREIGN KEY (`ingredient\_id`) REFERENCES `ingredients` (`ingredients\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=165 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `ingredients`

--

DROP TABLE IF EXISTS `ingredients`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `ingredients` (

`ingredients\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) NOT NULL,

`quantity` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`ingredients\_id`),

UNIQUE KEY `ingredients\_id\_uindex` (`ingredients\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=81 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `order\_dishes`

--

DROP TABLE IF EXISTS `order\_dishes`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `order\_dishes` (

`order\_dishes\_id` int(11) NOT NULL AUTO\_INCREMENT,

`order\_id` int(11) NOT NULL,

`dish\_id` int(11) NOT NULL,

`count` int(11) DEFAULT NULL,

PRIMARY KEY (`order\_dishes\_id`),

UNIQUE KEY `order\_dishes\_id\_uindex` (`order\_dishes\_id`),

KEY `order\_dishes\_orders\_id\_fk` (`order\_id`),

KEY `order\_dishes\_dishes\_id\_fk` (`dish\_id`),

CONSTRAINT `order\_dishes\_dishes\_id\_fk` FOREIGN KEY (`dish\_id`) REFERENCES `dishes` (`dishes\_id`),

CONSTRAINT `order\_dishes\_orders\_id\_fk` FOREIGN KEY (`order\_id`) REFERENCES `orders` (`orders\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=112 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `order\_status`

--

DROP TABLE IF EXISTS `order\_status`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `order\_status` (

`order\_status\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`order\_status\_id`),

UNIQUE KEY `order\_status\_id\_uindex` (`order\_status\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `orders`

--

DROP TABLE IF EXISTS `orders`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `orders` (

`orders\_id` int(11) NOT NULL AUTO\_INCREMENT,

`type` int(11) NOT NULL,

`created\_at` datetime DEFAULT NULL,

`date` datetime NOT NULL,

`status\_id` int(11) NOT NULL,

`table\_id` int(11) DEFAULT NULL,

PRIMARY KEY (`orders\_id`),

UNIQUE KEY `orders\_id\_uindex` (`orders\_id`),

KEY `orders\_order\_status\_id\_fk` (`status\_id`),

KEY `orders\_tables\_id\_fk` (`table\_id`),

CONSTRAINT `orders\_order\_status\_id\_fk` FOREIGN KEY (`status\_id`) REFERENCES `order\_status` (`order\_status\_id`),

CONSTRAINT `orders\_tables\_id\_fk` FOREIGN KEY (`table\_id`) REFERENCES `tables` (`tables\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=127 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `reservation`

--

DROP TABLE IF EXISTS `reservation`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `reservation` (

`reservation\_id` int(11) NOT NULL AUTO\_INCREMENT,

`table\_id` int(11) NOT NULL,

`name` varchar(255) DEFAULT NULL,

`second\_name` varchar(255) DEFAULT NULL,

`phone` varchar(255) DEFAULT NULL,

`order\_id` int(11) DEFAULT NULL,

`created\_at` datetime DEFAULT NULL,

`reservation\_date` datetime NOT NULL,

PRIMARY KEY (`reservation\_id`),

UNIQUE KEY `reservation\_id\_uindex` (`reservation\_id`),

KEY `reservation\_orders\_id\_fk` (`order\_id`),

KEY `reservation\_tables\_id\_fk` (`table\_id`),

CONSTRAINT `reservation\_orders\_id\_fk` FOREIGN KEY (`order\_id`) REFERENCES `orders` (`orders\_id`),

CONSTRAINT `reservation\_tables\_id\_fk` FOREIGN KEY (`table\_id`) REFERENCES `tables` (`tables\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=28 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `tables`

--

DROP TABLE IF EXISTS `tables`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tables` (

`tables\_id` int(11) NOT NULL AUTO\_INCREMENT,

`person\_count` int(11) NOT NULL,

`photo\_url` varchar(255) DEFAULT NULL,

`updated\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`table\_status` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`tables\_id`),

UNIQUE KEY `tables\_id\_uindex` (`tables\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `users`

--

DROP TABLE IF EXISTS `users`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `users` (

`type` int(11) NOT NULL,

`login` varchar(255) NOT NULL,

`password\_hash` varchar(255) NOT NULL,

`users\_id` int(11) NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`users\_id`),

UNIQUE KEY `users\_id\_uindex` (`users\_id`),

UNIQUE KEY `users\_login\_uindex` (`login`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

Admin\_packages.json

{

"@angular/common": "^4.0.0",

"@angular/compiler": "^4.0.0",

"@angular/core": "^4.0.0",

"@angular/forms": "^4.0.0",

"@angular/http": "^4.0.0",

"@angular/platform-browser": "^4.0.0",

"@angular/platform-browser-dynamic": "^4.0.0",

"@angular/router": "^4.0.0",

"@ng-bootstrap/ng-bootstrap": "^1.0.0-alpha.22",

"angular2-jwt": "^0.2.3",

"bootstrap-material-design": "^0.5.10",

"core-js": "^2.4.1",

"font-awesome": "^4.7.0",

"ionicons": "^3.0.0",

"ng2-charts": "^1.5.0",

"ng2-select": "^1.2.0",

"rxjs": "^5.1.0",

"zone.js": "^0.8.4"

}

Main.ru

from flask import Flask

from flask import jsonify

from datetime import datetime, timedelta

from MySQLdb.cursors import DictCursor

from flaskext.mysql import MySQL

import os

app = Flask('restaurant')

DB\_USER = os.getenv('MYSQL\_USER', 'user')

DB\_PASSWORD = os.getenv('MYSQL\_PASSWORD', '55F3WzF7yudAd8L#')

DB\_NAME = os.getenv('MYSQL\_DATABASE', 'restaurant')

DB\_HOST = '172.19.0.2'

app.config.update(

MYSQL\_DATABASE\_HOST=DB\_HOST,

MYSQL\_DATABASE\_USER=DB\_USER,

MYSQL\_DATABASE\_PASSWORD=DB\_PASSWORD,

MYSQL\_DATABASE\_DB=DB\_NAME

)

mysql = MySQL(cursorclass=DictCursor)

mysql.init\_app(app)

TABLE\_STATUS\_FREE = 0

TABLE\_STATUS\_RESERVED = 1

TABLE\_STATUS\_BUSY = 2

@app.route("/tables")

def all\_tables():

tables = []

cursor = mysql.get\_db().cursor()

sql = "SELECT tables\_id, table\_status, updated\_at FROM tables"

cursor.execute(sql)

rows = cursor.fetchall()

for row in rows:

# tables[index] = {'tables\_id': <int>, 'table\_status': <int>, 'updated\_at': <datetime>}

if not can\_reserve(datetime.now(), row['tables\_id']):

row['table\_status'] = TABLE\_STATUS\_RESERVED

tables.append(row)

jsonify(tables)

def can\_reserve(date, table\_id):

cursor = mysql.get\_db().cursor()

table\_sql = "SELECT tables\_id, table\_status, updated\_at FROM tables WHERE tables\_id = {}"\

.format(table\_id)

reservation\_sql = "SELECT reservation\_date, table\_id " \

"FROM reservations " \

"WHERE table\_id = {} ORDER BY reservation\_date DESC".format(table\_id)

cursor.execute(table\_sql)

table = cursor.fetchone()

cursor.execute(reservation\_sql)

reservation = cursor.fetchone()

if table['table\_status'] == TABLE\_STATUS\_FREE \

and table['updated\_at'] > reservation['reservation\_date']:

return True

if table['table\_status'] == TABLE\_STATUS\_BUSY and table['updated\_at'] > reservation['reservation\_date']:

return date > (table['updated\_at'] + timedelta(hours=5))

return False

def can\_create(dish\_id, count=1):

pass

app.run(host='0.0.0.0', port=8080)