**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| **ДП.09.02.07-3.24.212.11 ПЗ** |  |

**ВЕБ ПРИЛОЖЕНИЕ «ДОСТАВКА ЕДЫ»**

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.С. Александрова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.Э. Кондратенко) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (В.А. Кузнецов) |
|  |  |  |

Иркутск 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc101598117)

# Введение

Сервис доставки еды - организация, занимающаяся доставкой готовых блюд из ресторана до потребителя. В настоящее время в России существует не так много сервисов (Яндекс лавка, Delivery, Яндекс Еда), которые осуществляют деятельность по доставке еды, это влияет на монополизированные рынка и большую загруженность сервисов. Поэтому создание удобного приложения для пользователей, рестораторов и курьеров является целесообразной задачей.

Функционала: регистрация и аутентификация со стороны администратора, клиента, курьера и администратора ресторана. Клиент может выбирать понравившейся ресторан в списке ресторанов, а также искать необходимый ресторан в поисковой строке, в них выбирать блюда и формировать свою корзину товаров. Администратор имеет возможность удалять, изменять пользователей, а также добавлять новые рестораны в базу данных, выдавать права доступа на основании запросов смотреть логи изменений и действий, в конкретном ресторане, и в базе данных пользователей, фильтровать логи. Администратор ресторана может смотреть информацию о своем ресторане и имеет возможность скачать отчет за определенный период

Целью курсового проекта является автоматизация бизнес-процесса по доставке еды и создание Информационной системы по доставке еды, которая реализует и автоматизирует весь вышеописанный функционал.

Основными задачами данного проекта являются:

– создание дизайна для информационной системы доставка еды;

– проектирование и создание базы данных для интернет-магазина;

– верстка дизайна интернет-магазина;

– добавление функционала;

– привязка базы данных к интернет-магазину.

**1 Предпроектное исследование**

**1.1 Описание предметной области**

В качестве предметной области выбрано веб приложение «Доставка еды», которая занимается доставкой еды из ресторана до клиента.

Веб приложение создается для реализации и автоматизации бизнес-процессов по оформлению заказа клиента, доставки заказа по указанному адресу и предоставления статистики для администратора ресторана.

Без веб приложения человек приходит в кафе или ресторан, для начала ему приходиться:

1. отстоять очередь.
2. назвать свой заказ и сделать это быстро так как с сзади него множество других таких же клиентов.
3. подождать приготовление заказа забрать его самому или дождаться официанта.

Так же поход в такие места заранее планируется или бронируется столик, если свободного столика нет, то вы не сможете там поесть.

С использованием веб приложения можно дистанционно заказать блюда из любимого ресторана, оплатить их онлайн или наличкой, можно заказать в любое желаемое время. Так же для рестораторов можно получить отчет о покупках в вашем ресторане.

Авторизованный клиент имеет возможность сделать заказ с помощью веб приложения. При оформлении заказа, клиент выбирает понравившееся позиции меню выбранного им ресторана. В процессе формирования заявки, клиенту предлагается выбрать способ оплаты и адрес доставки, а также написать пояснительный комментарий к заказу. Для администратора все заявки от клиентов отображаться на странице пользователя в таблице пользователей, для самого пользователя его личные заказы отображаться в разделе «мои заказы», для курьера доступные заказы отображаться в разделе доступные заказы. В заявке указаны блюда заказа, пояснительный комментарий, итоговая стоимость заказа, адрес ресторана и адрес доставки, также статус заказа, номер курьера, который занят доставкой.

В данном веб приложении можно выделить следующие сущности:

Пользователь-клиент с атрибутами: id, ФИО, номер телефона, email, пароль, дата создания, дата обновления.

Пользователь-администратор с атрибутами: id, email, пароль, дата создания, дата обновления.

Пользователь-курьер с атрибутами: id, ФИО, номер телефона, email, пароль, id заказа, дата создания, дата обновления.

Пользователь-администратор ресторана с атрибутами: id, ФИО, номер телефона, email, пароль, id ресторана, дата создания, дата обновления.

Основным видом деятельности сервиса доставки еды является доставка еды из заведения к клиенту.

Процесс доставки еды в сервисе доставки еды можно представить так:

После авторизации клиент выбирает блюдо и формирует заказ по его предпочтениям и пожеланиям. Далее для сформированного заказа клиент выбирает место доставки и оплачивает его (Выбирает картой или наличными). Далее заказ попадает в список доступных заказов для курьера, свободный курьер принимает заказ и начинается доставка.

В результате анализа деятельности отдела менеджмента был выявлен комплекс информационных потоков. Входные данные:

* сведения о покупателях;
* сведения о ресторане для администратора ресторана;
* сведения о заведениях;
* сведения о корзине;
* сведения о курьерах;
* сведения об блюдах
* сведения о заказе на доставку.

Информация о готовности заказа, цена заказа, цена каждой позиции, количество блюд, вес блюда.

## **2 Анализ инструментов для разработки ПО**

Инструменты разработки программного продукта определяют будущий результат.

Проектировать структуру информационной системы удобно через MySQL Workbench и Draw.io, а его дизайн – через онлайн-сервис Figma. Информационная система будет состоять из двух частей – клиентская и серверная. Для реализации клиентской части отлично подойдут следующие инструменты: HTML5, CSS3 и JS, Tailwind. Серверная часть будет действовать на PHP с использованием фреймворка Laravel с базой данных MySQL.

MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL. В проекте используется для создания ER-диаграммы БД.

Draw.io – это удобное бесплатное онлайн-приложение для создания диаграмм для рабочих процессов, BPM, организационных, сетевых диаграмм, блок-схем (флоучарты), UML и принципиальных электросхем. В проекте используется для создания прототипа страниц.

Figma – бесплатный удобный онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики. В редакторе можно настроить совместную работу, вносить и обсуждать правки, причём как в браузере, так и через приложение на компьютере. Популярен, в частности, для разработки прототипа и дизайна сайта или приложения. В проекте используется для создания наглядного ожидаемого дизайна проекта, а также уникальных SVG-элементов в едином стиле.

HTML – язык разметки гипертекста. Язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе. Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме.

CSS – каскадные таблицы стилей, которые используются для определения стилей (правил) оформления документов – включая дизайн, вёрстку и вариации макета для различных устройств и размеров экрана.

JavaScript – это полноценный динамический язык программирования, который применяется к HTML-документу и может обеспечить динамическую интерактивность. JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

Tailwind - это CSS-фреймворк, предоставляющий набор готовых классов для стилизации веб-интерфейсов.

Информационная система будет содержать в себе информацию – её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены следующие варианты реализации СУБД:

1. MySQL.

2. SQLite.

3. PostgreSQL.

MySQL – свободная реляционная система хранения и управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощенной Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

На сегодняшний день является самой популярной серверной базы данных (далее – БД), за счет своей простоты, скорости работы функционала. Поддерживаются такие основные движки MyISAM, InnoDB, MEMORY, Berkeley DB. Реализация всех новых возможностей стандарта SQL отсутствует в пользу простоты использования.

PostgreSQL – свободная объектно-реляционная система хранения и управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows. Работает только на одном движке – Storage Engine. Все таблицы представлены в виде объектов, они могут наследоваться, а все действия с таблицами выполняются с помощью объектно-ориентированных функций. Обладает открытым исходным кодом, разрабатывается командой энтузиастов, при этом старается максимально соответствовать стандарту SQL. Реализует все самые новые стандарты, что приводит к ущербу простоты, из-за чего PostgreSQL очень сложный и уступает в популярности MySQL.

SQLite – компактная встраиваемая СУБД с исходным кодом. В 2005 году проект получил награду Google-O’Reilly Open Source Awards. SQLite поддерживает динамическое типизирование данных. Возможные типы значений: INTEGER, REAL, TEXT и BLOB. Также поддерживается специальное значение NULL. Размеры значений типа TEXT и BLOB не ограничены ничем, кроме константы SQLITE MAX\_LENGTH в исходниках sqlite, равной миллиарду.

SQLite напрямую хранит информацию в одном файле, что облегчает его копирование. Большая популярность в мобильной разработке и небольших автономных приложениях, поскольку она занимает меньше места на дисковом пространстве, имеет высокую скорость работы и не требует в отличии от MySQL не требует наличие сервера для запуска. Минусы: ограничения на запись, всего 5 типов данных, отсутствие встроенного механизма аутентификации.

Для наглядности сравнения вариантов реализации базы данных была составлена таблица 1.

Таблица 1 – Сравнение средств реализации базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название БД | MySQL | SQLite | PostgreSQL |
| Большое кол-во типов данных | + | - | + |
| Популярность | + | + | - |
| Отказоустойчивость | - | - | + |
| Не требует удаленного сервера | - | + | - |
| Простота использования | + |  | - |
| Портативность | - | - | - |

Таким образом, в качестве базы данных для будущего продукта была выбрана MySQL, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении, достаточно быстрая, предоставляет возможность гибкой настройки, легко переносимая.

Для взаимосвязи баз данных и северной части продукта необходимо использовать серверный язык. Для реализации этого были рассмотрены два языка программирования – Python и Php.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным – все является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

Недостатками языка являются использование языка в серверной разработке, только благодаря фреймворку, а также зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как Си или C++. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов. Действует, как и самостоятельно, так и с фреймворками.

Язык и его интерпретатор (Zend Engine) разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

Язык автоматически поддерживает HTTP Cookies в соответствии со стандартами Netscape. Это позволяет проводить установку и чтение небольших сегментов данных на стороне клиента. Работа с Cookies организована посредством сеансов (сессий). У сессий есть срок действия (после его истечения данные удаляются), в сессиях можно хранить и редактировать разные типы данных, в том числе сериализованные PHP-объекты, пропущенные через serialize (процесс происходит автоматически).

Для наглядности сравнения языков программирования была составлена таблица 2.

Таблица 2 – Сравнение языков программирования для разработки программного продукта

|  |  |  |
| --- | --- | --- |
| Название языка программирования | Php | Python |
| Наличие библиотек | + | + |
| Инструменты для работы с БД | + | + |
| Объектно-ориентированные возможности | + | + |
| Лёгкий понятный синтаксис | + | - |
| Более активное сообщество | + | - |
| Более легкая простая модульность | + | - |

Таким образом, Php будет более лучшим вариантом, так как он имеет большое количество библиотек и имеет инструменты для работы с БД. В сравнении с языком Python, у Php работа с программными модулями проще, а также его сообщество более активное.

Для разработки программного продукта рассмотрены следующие инструментальные средства разработки программных продуктов:

1. PyCharm.

2. Visual Studio Code.

3. Spyder.

4. Atom.

PyCharm – одна из лучших полнофункциональных IDE, предназначенных именно для Python. Существуют, как и версия Community(бесплатная), так и Professional (платная). PyCharm доступен на Windows, Mac OS X и Linux. PyCharm уже сразу после установки поддерживает разработку на Python – можете открыть новый файл и сразу же писать код. Вы сможете запускать и отлаживать код прямо в PyCharm. Также в IDE есть поддержка проектов и системы управления версиями. Преимущества: поддержка всего и хорошее сообщество: можно редактировать, запускать и отлаживать Python-код сразу, как скачал и установил PyCharm. Недостатки: из-за большого количества встроенных инструментов может медленно загружаться на слабых компьютерах.

Visual Studio Code – Редактор кода, который построен на основе открытых источников, работает везде и имеет встроенные функции для запуска и отладки кода, а также расширения для установки новых языков, тем, отладчиков и подключения к дополнительным сервисам.

Среди других возможностей VS Code можно выделить:

- IntelliSense: предоставляет умные завершения на основе типов переменных, определений функций и импортированных модулей.

- Debugging: позволяет отлаживать код прямо из редактора.

- Git: встроенные функции для работы с Git и другими SCM-провайдерами.

- Extensibility: возможность установки расширений для добавления новых языков, тем, отладчиков и подключения к дополнительным сервисам.

- Deployment: с помощью Microsoft Azure можно легко и удобно разворачивать и размещать сайты на React, Angular, Vue, Node, Python (и других) платформах, хранить и запрашивать данные, а также масштабировать проекты с помощью сервер less computing.

Spyder – свободная и кроссплатформенная интерактивная IDE для научных расчетов на языке Python, обеспечивающая простоту использования функциональных возможностей и легковесность программной части.

Spyder является частью модуля spyderlib для Python, основанного на PyQt4, pyflakes, rope и Sphinx, предоставляющего мощные виджеты на PyQt4, такие как редактор кода, консоль Python (встраиваемая в приложения) и графический редактор переменных (в том числе списков, словарей и массивов).

Отличительной особенностью Spyder является наличие проводника переменных. Он позволяет просмотреть значения переменных в форме таблицы прямо внутри IDE. Преимущества: он бесплатный, с открытым исходным кодом и доступен на Windows, macOS и Linux. Недостатки: опытные разработчики на Python считают Spyder недостаточно функциональным для работы и предпочитают другую более функциональную IDE.

Atom – это многофункциональный текстовый редактор от разработчиков GitHub. Он поддерживает огромное количество различных расширений, благодаря которым его можно сравнить с настоящей средой разработки. Другая особенность – платформа Electron, на которой держится Atom. Она включает в себя Node.js и движок от Chrome – такой инструментарий позволяет разрабатывать приложения для рабочего стола на основе веб-технологий.

Atom позволяет синхронизировать проект с репозиториев GitHub, находить и устанавливать необходимые компоненты под требуемые задачи, поскольку у Atom открытый исходный код, и много разработчиков уже внесли вклад в развитие программы. Остаётся брать и пользоваться с удовольствием.

Сравнение IDE для разработки программного продукта наглядно представлено в таблице 3.

Таблица 3 – Сравнение IDE для разработки программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название IDE | Spyder | Visual Studio Code | PyCharm | Atom |
| Распространяется бесплатно | + | + | - | + |
| Автоматическое сохранение | + | + | + | + |
| Подсказки по коду | - | + | + | - |
| Интеграция с системой контроля версия (GIT) | - | + | + | - |
| Возможность расширения функционала библиотеками | - | + | + | + |
| Поддержка CSS/HTML/JS | - | + | + | + |
| Комфортное использование на слабых ПК | + | + | - | \_ |

1. Таким образом, после рассмотрения вариантов средств разработок, было принято решение использовать Visual Studio Code.
2. Тем не менее, Visual Studio Code обладает преимуществами:
3. Поддержка всех современных языков для web-разработки, таких как: JavaScript, HTML/CSS, TypeScript;
4. Встроенная система контроля версий Git;
5. Поддержка популярных веб-фреймворков, таких как Laravel;
6. Поддержка большинства ОС: Windows, Linux и MacOS;
7. Встроенные инструменты для работы с базами данных (Доступ к Oracle, SQL Server, PostgreSQL, MySQL и другим базам данных);
8. После рассмотрения всех инструментов разработки для проекта были выбраны HTML5, CSS3, JS, SQLite и PHP-фреймворк Laravel, а инструменты для проектирования и разработки web-приложения – Draw.io, MySQL Workbench, Figma и Visual Studio Code.

# 3 Техническое задание

# В начале разработки создавалось техническое задание, в котором указывались основные требования.

# Для создания технического задания использовался стандарт ГОСТ 34.602-2020.

# Согласно ГОСТ 34.602-2020 техническое задание должно включать следующие разделы:

# 1. общие сведения;

# 2. назначение и цели создания информационной системы;

# 3. требования к информационной системе в целом;

# 3.1. требования к структуре и функционированию информационной системы;

# 3.2. требования к надежности;

# 3.3. требования к безопасности;

# 3.4. требования к эксплуатации, техническому обслуживанию компонентов системы;

# 4. требования к документированию;

# 5. состав и содержание по разработке информационной системы.

# Техническое задание на разработку информационной системы представлено в приложении А.

# 4 Проектирования информационной системы

## **4.1 Структурная схема информационной системы**

На рисунке 1 изображена диаграмма деятельности, которая показывает бизнес-процессы, которые происходят в информационной системе. По ней можно определить автоматизация каких процессов происходит в информационной системе

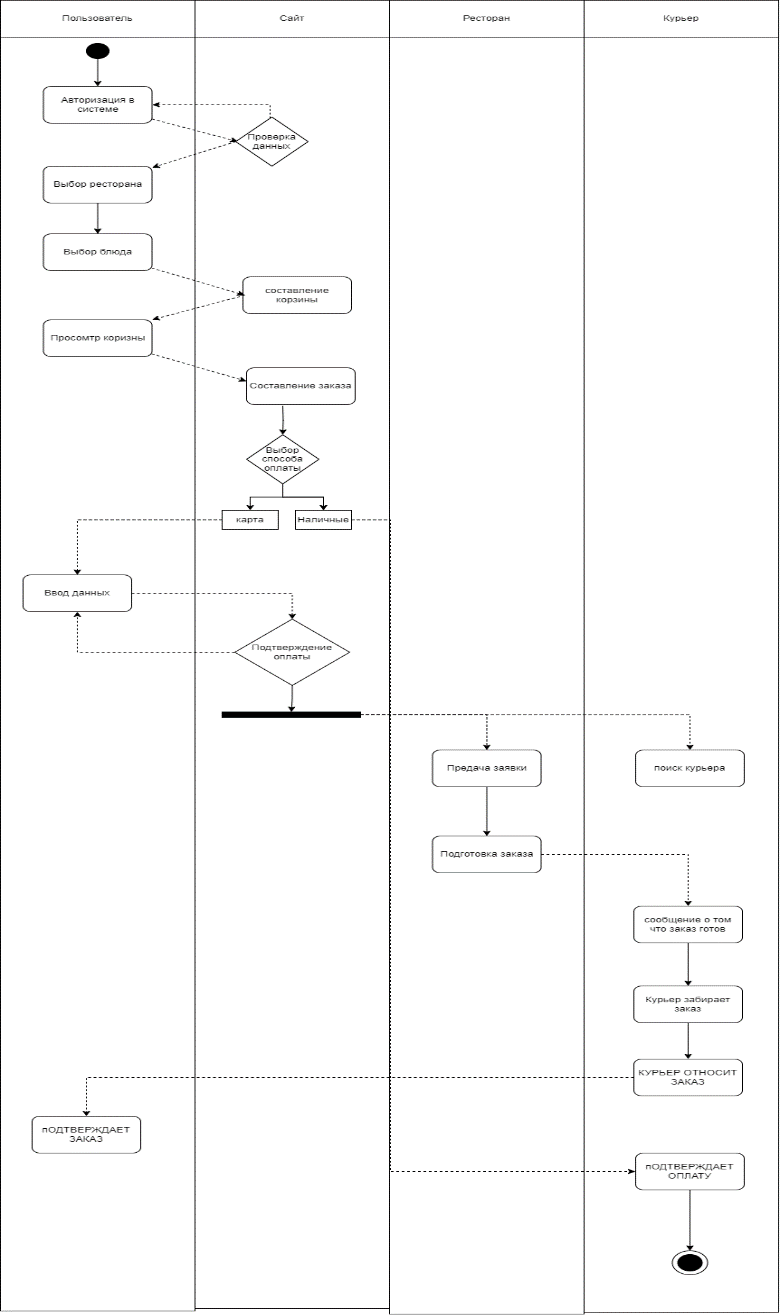


Рисунок 1 – Диаграмма деятельности

На рисунке 2 изображена диаграмма A0 которая демонстрирует общую логику системы. Можно сделать вывод о том, как построен процесс заказа и построена логика системы.

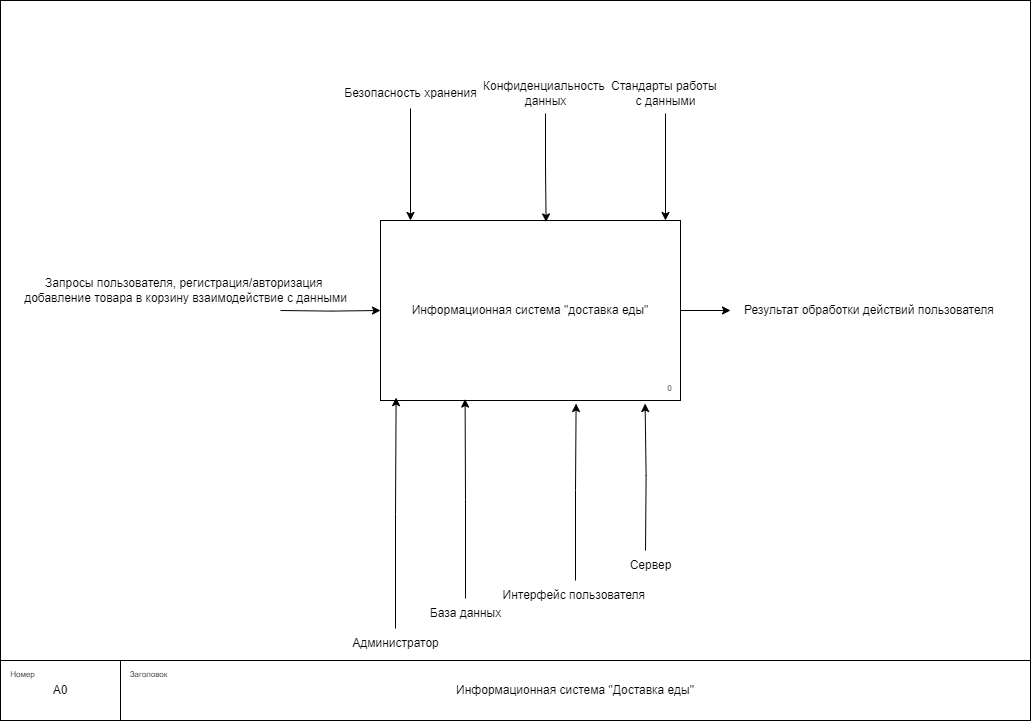


Рисунок 2 – Диаграмма A0

На рисунке 3 изображена диаграмма A1 которая демонстрирует общую логику системы и декомпозицию определенного процесса. Можно сделать вывод о том, как построен процесс заказа и построена логика системы.

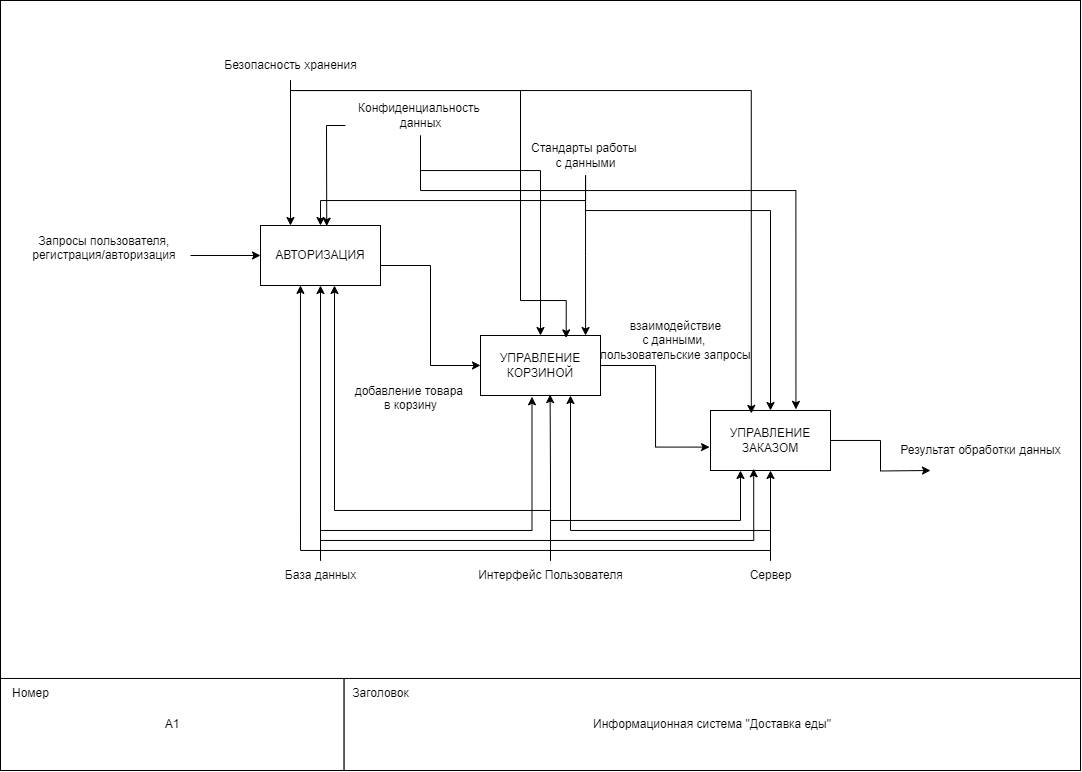


Рисунок 3 – Диаграмма A1

## **4.2 Функциональная схема информационной системы**

На рисунке 3 изображена диаграмма прецедентов, которая показывает какие актеры участвуют в системе и какие действия им доступны. Можно сделать вывод о какие функции исполнят каждая сущность.

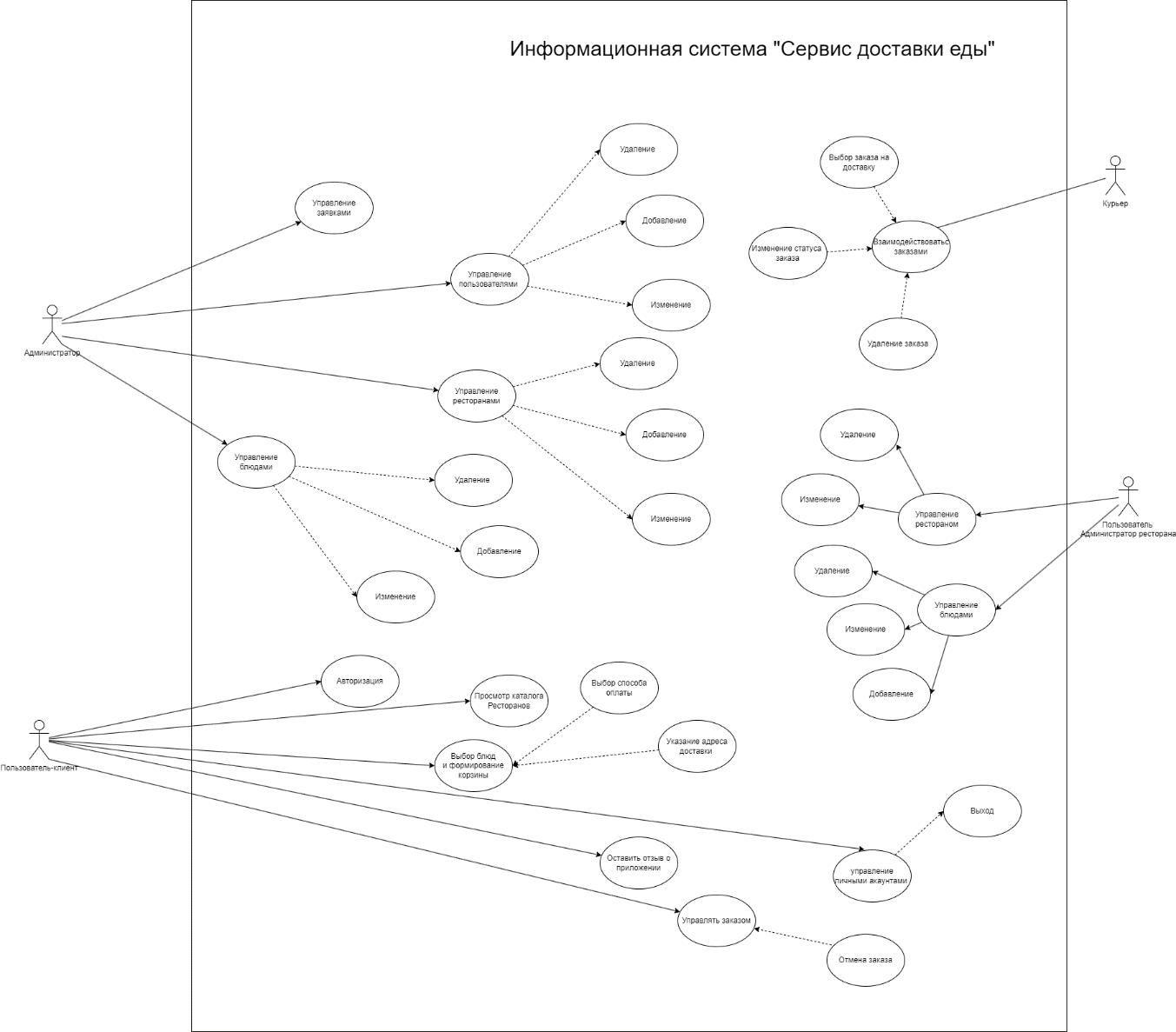


Рисунок 4 – Диаграмма прецедентов

На рисунке 5 изображена диаграмма DFD предназначенная для моделирования информационной системы с точки зрения хранения, обработки и передачи данных. Можно сделать вывод о том, как построено взаимодействие с Базами Данных.

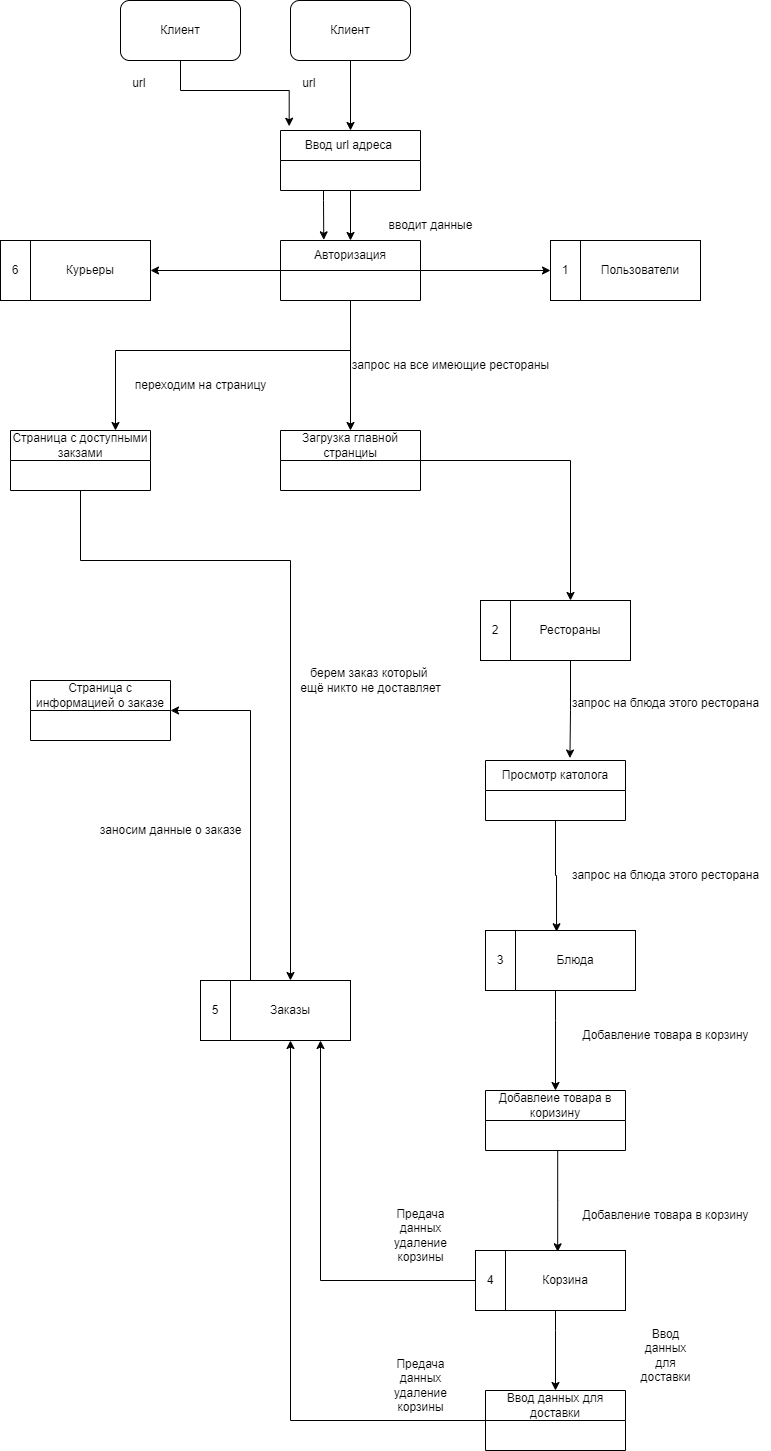


Рисунок 5 – Диаграмма потоков данных

## **4.3 Проектирование базы данных**

На рисунке 6 изображена er модель которая демонстрирует работу базы данных, показывает типы данных и внешние ключи по которым работает приложение.

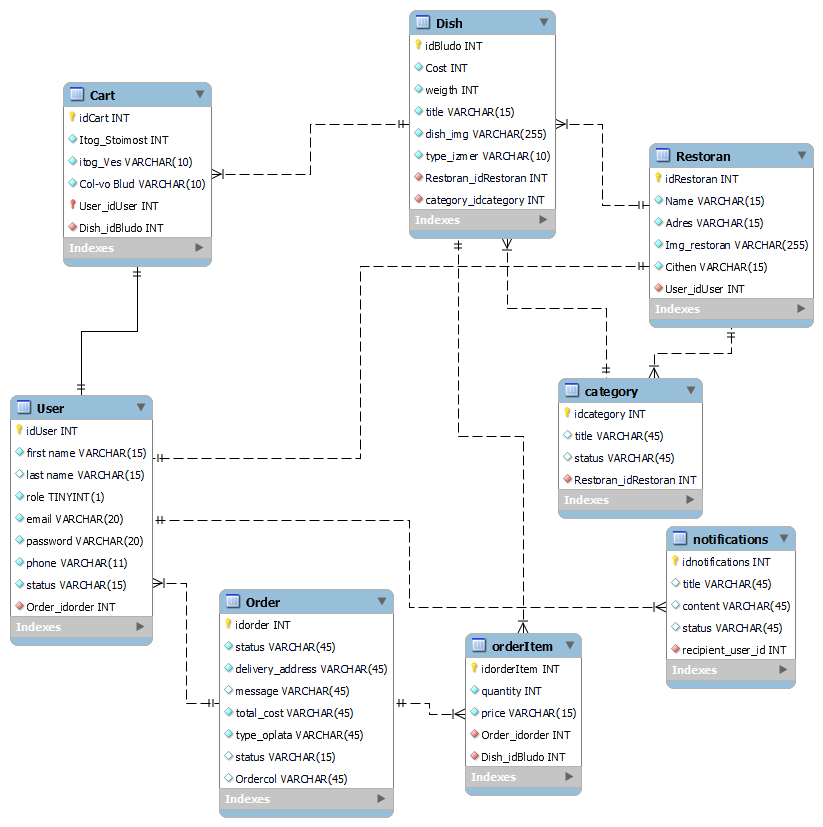


Рисунок 6 – ER-модель базы данных

Таблица 4 – Restoran

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idRestoran | INT | Показывает номер ресторана в базе для поиска |
| Name | VARCHAR(45) | Показывает название ресторана в базе |
| Img\_restoran | VARCHAR(45) | Показывает название картинки для ресторана в базе |
| Adres | VARCHAR(45) | Показывает адрес заведения ресторана в базе |
| Cithen | VARCHAR(45) | Показывает кухню ресторана |
| User\_id | Int | Внешний ключ, который используется для связи между рестораном и администратором ресторана |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 4, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 5 – Cart

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idCart | INT | Показывает номер корзины в базе для поиска |
| Itog\_  Stoimost | INT | Показывает итоговую стоимость в базе |
| Col-vo\_blud | INT | Показывает количество блюд в базе |
| Dish\_id | INT | Внешний ключ используется для связи между корзиной и рестораном |
| User\_clientid | INT | Связь между клиентом и корзиной |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 5, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 6 – Order

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idorder | INT | Показывает номер Заказа в базе для поиска |
| Total\_cost | INT | Показывает стоимость Заказа в базе |
| Status | VARCHAR(45) | Показывает статус Заказа в базе |
| delivery\_  address | VARCHAR(45) | Показывает адрес доставки Заказа в базе |
| message | VARCHAR(45) | Показывает комментарий к заказу доставки Заказа в базе |
| Type\_oplata | VARCHAR(45) | Показывает способ оплаты доставки Заказа в базе |
| Curier\_id | int | Связь между заказом и курьером |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 6, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 7 – User

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idClient | INT | Показывает номер пользователя в базе для поиска |
| name | VARCHAR(45) | Показывает имя пользователя в базе |
| surname | VARCHAR(45) | Показывает фамилию пользователя в базе |
| Tel | VARCHAR(45) | Показывает телефон для связи пользователя в базе |
| Emial | VARCHAR(45) | Показывает почту для связи ресторана в базе |
| Password | VARCHAR(45) | Показывает пароль пользователя в базе |
| Role | TINIINT(1) | Показывает роль пользователя в базе |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 7, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 8 – dish

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idDish | INT | Показывает номер блюда в базе для поиска |
| title | VARCHAR(45) | Показывает название блюда в базе |
| Cost | INT | Показывает стоимость блюда в базе |
| weigth | INT | Показывает вес блюда в базе |
| dish\_img | VARCHAR(45) | Показывает название картинки блюда в базе |
| Restoran\_  idRestoran | VARCHAR(45) | Связь между блюдом и рестораном |
| type\_izmer | VARCHAR(10) | Показывает какой тип измерения используется |
| Category  \_idcategory | INT | Показывает какой категории соответствует блюдо |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 8, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 9 – Order\_Item

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idorderItem | INT | Показывает номер блюда в базе для поиска |
| quantity | INT | Показывает количество блюд в таблице |
| Order\_idorder | INT | Внешний ключ, используется для определения к какому заказу относиться запись |
| Dish\_idBludo | INT | Внешний ключ, используется для связи с таблицей блюд |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 8, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 9 – Category

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idcategory | INT | Показывает номер категории в базе для поиска |
| title | VARCHAR(45) | Показывает название категории в базе |
| status | VARCHAR(45) | Показывает статус категории в базе |
| Restoran  idRestoran | INT | Внешний ключ, для связи с таблицей ресторан |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 8, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

Таблица 8 – notifications

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Назначение |
| idnotifications | INT | Показывает уведомления в базе для поиска |
| title | VARCHAR(45) | Показывает заголовок уведомления в базе |
| content | VARCHAR(45) | Показывает контент уведомления в базе |
| status | VARCHAR(45) | Показывает статус уведомления в базе |
| Recipient  user\_id | INT | Внешний ключ |

Вывод: при детальном рассмотрении таблицы с подробным описанием таблицы 8, можно сделать вывод о том какие атрибуты будут записаны в таблице и использованы во время работы информационной системы.

## **4.4 Проектирование пользовательского интерфейса**

Дизайн страниц информационной системы «Доставки еды» был выполнен с помощью онлайн инструмента Figma.

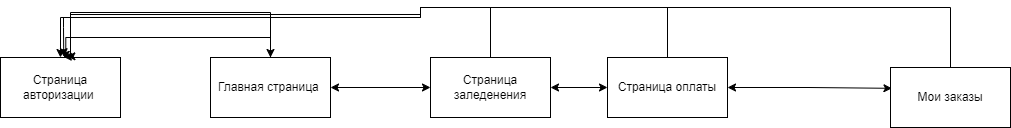


Рисунок 6 – Диаграмма навигации по информационной системы

### 4.4.1 Разработка прототипов интерфейса

На рисунке 7 изображен прототип страницы истории заказов. Можно сделать вывод о будущем расположении элементов и структуры страницы.

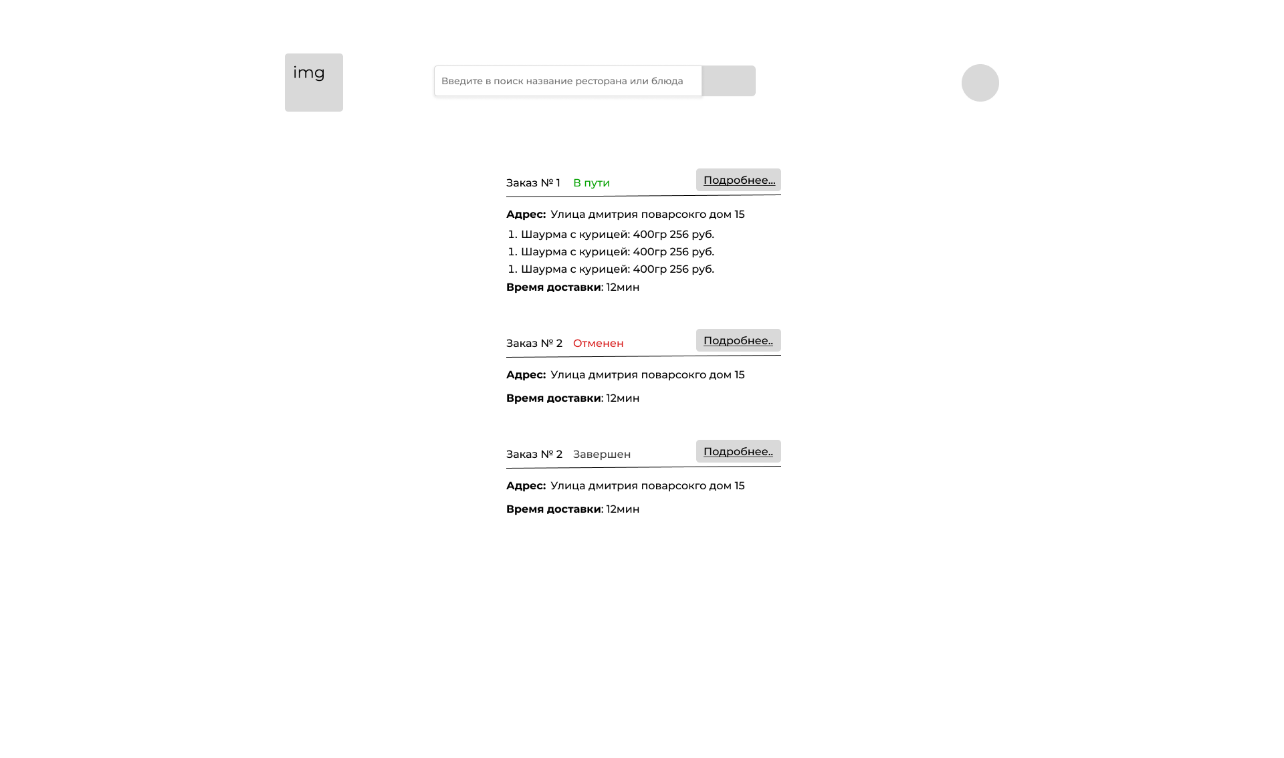


Рисунок 8 – Прототип страницы истории заказов

На рисунке 8 изображен прототип страницы регистрации. Можно сделать вывод о будущем расположении элементов и структуры страницы.

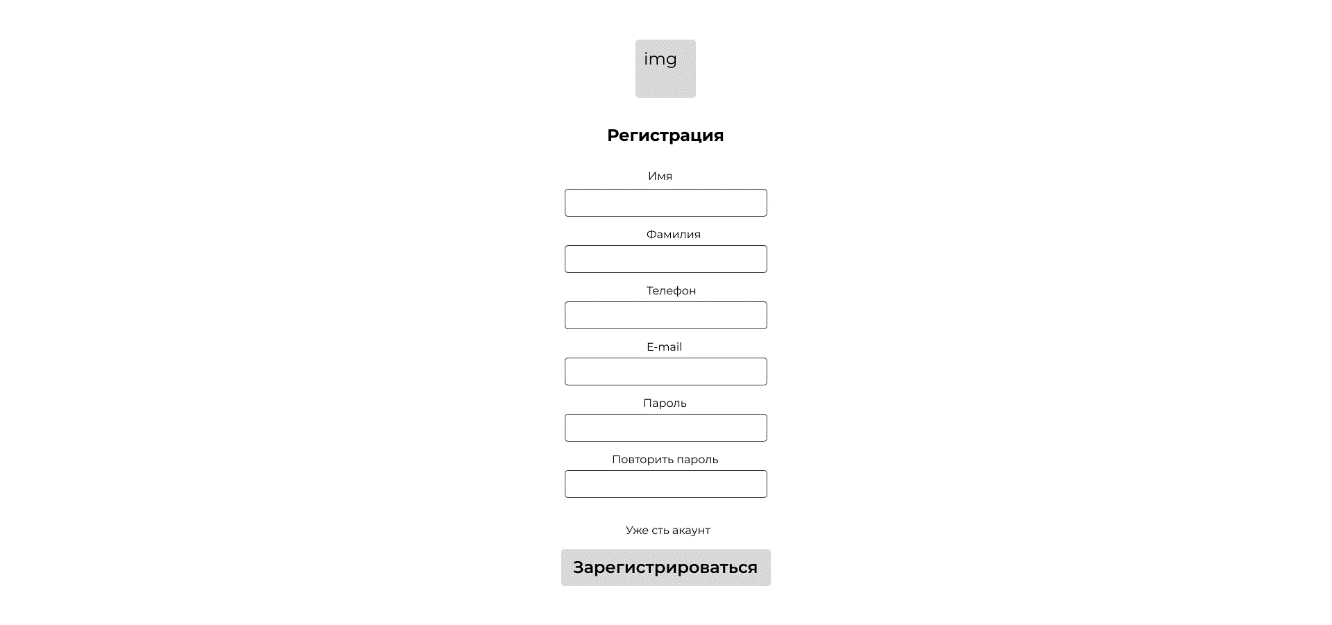


Рисунок 8 – Прототип страницы регистрации

На рисунке 9 изображен прототип страницы входа на сайт. Можно сделать вывод о будущем расположении элементов и структуры страницы.

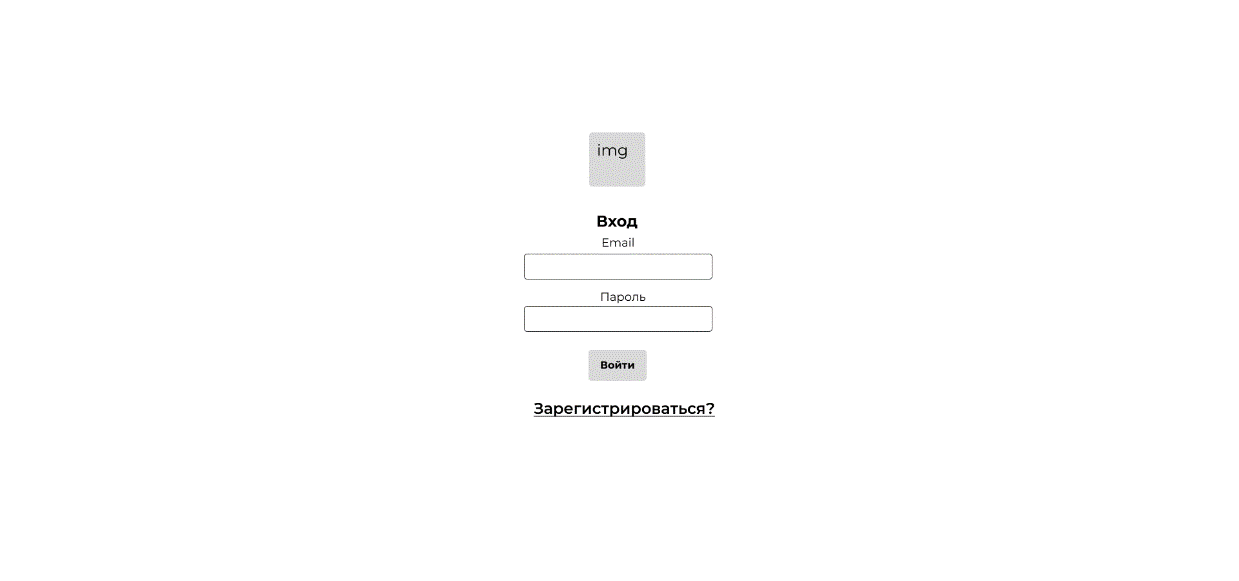


Рисунок 9 – Прототип страницы входа

На рисунке 10 изображен прототип страницы базы данных пользователей. Можно сделать вывод о будущем расположении элементов и структуры страницы.

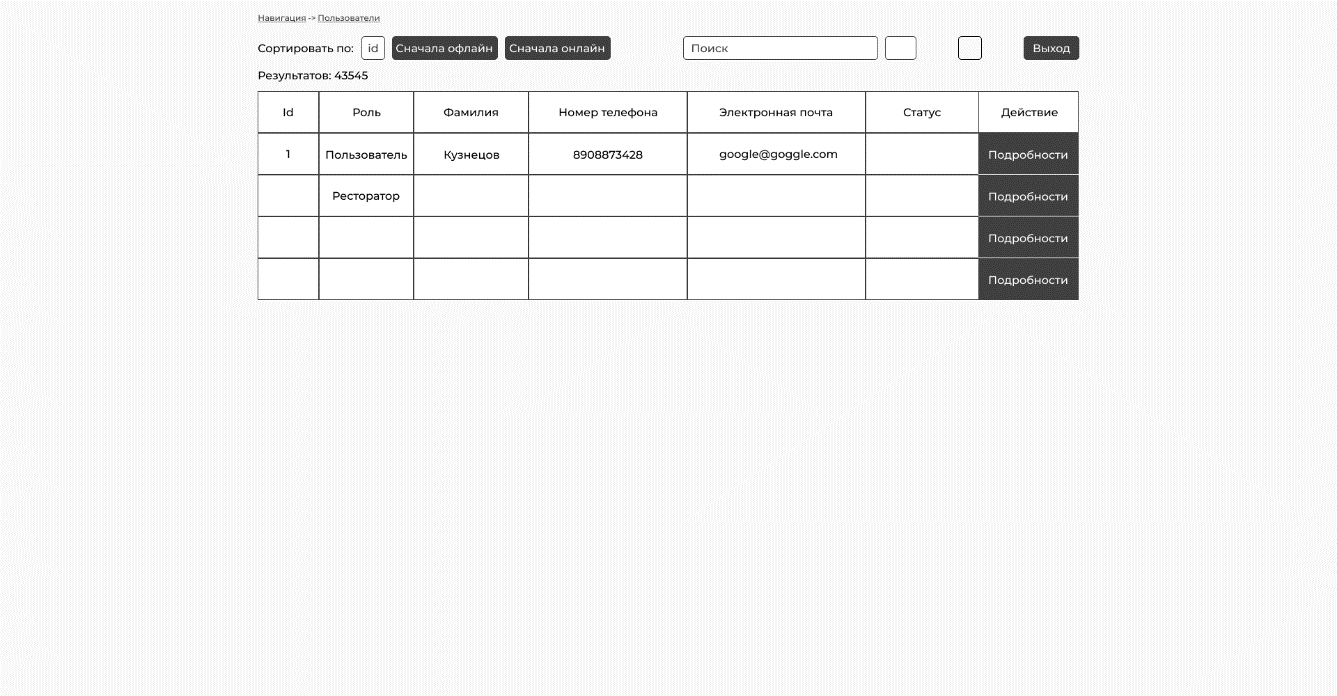


Рисунок 10 – Прототип страницы базы данных

На рисунке 11 изображен прототип страницы оплаты заказа на которой можно выбрать способ доставки и оплаты. Можно сделать вывод о будущем расположении элементов и структуры страницы.

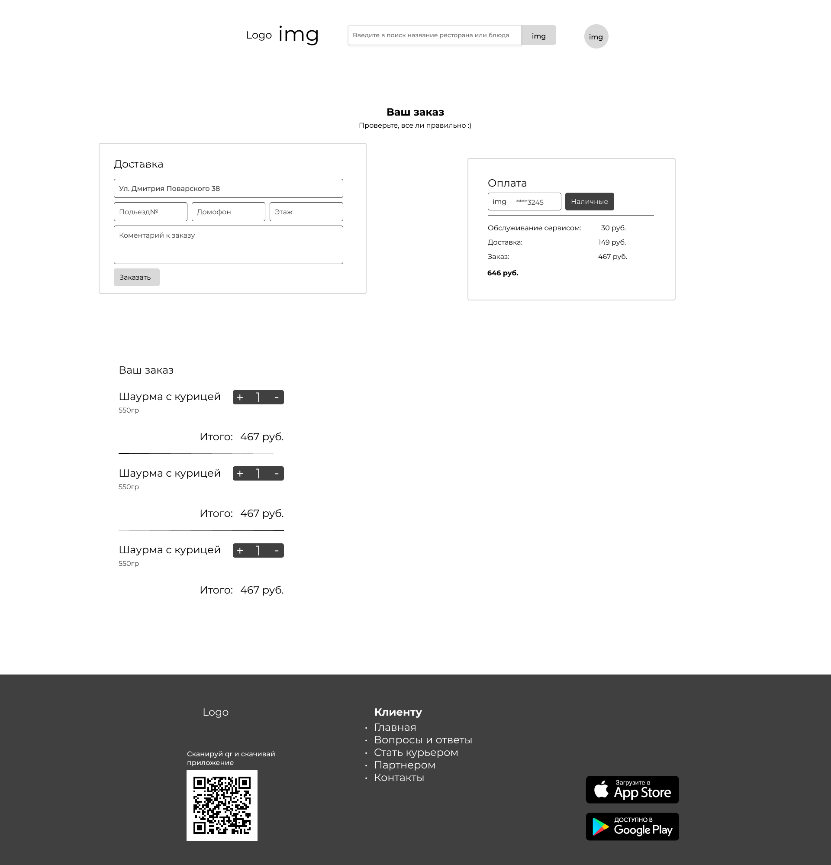


Рисунок 11 – Прототип страницы оплаты заказа

На рисунке 12 изображен прототип страницы заведения. Можно сделать вывод о будущем расположении элементов и структуры страницы.

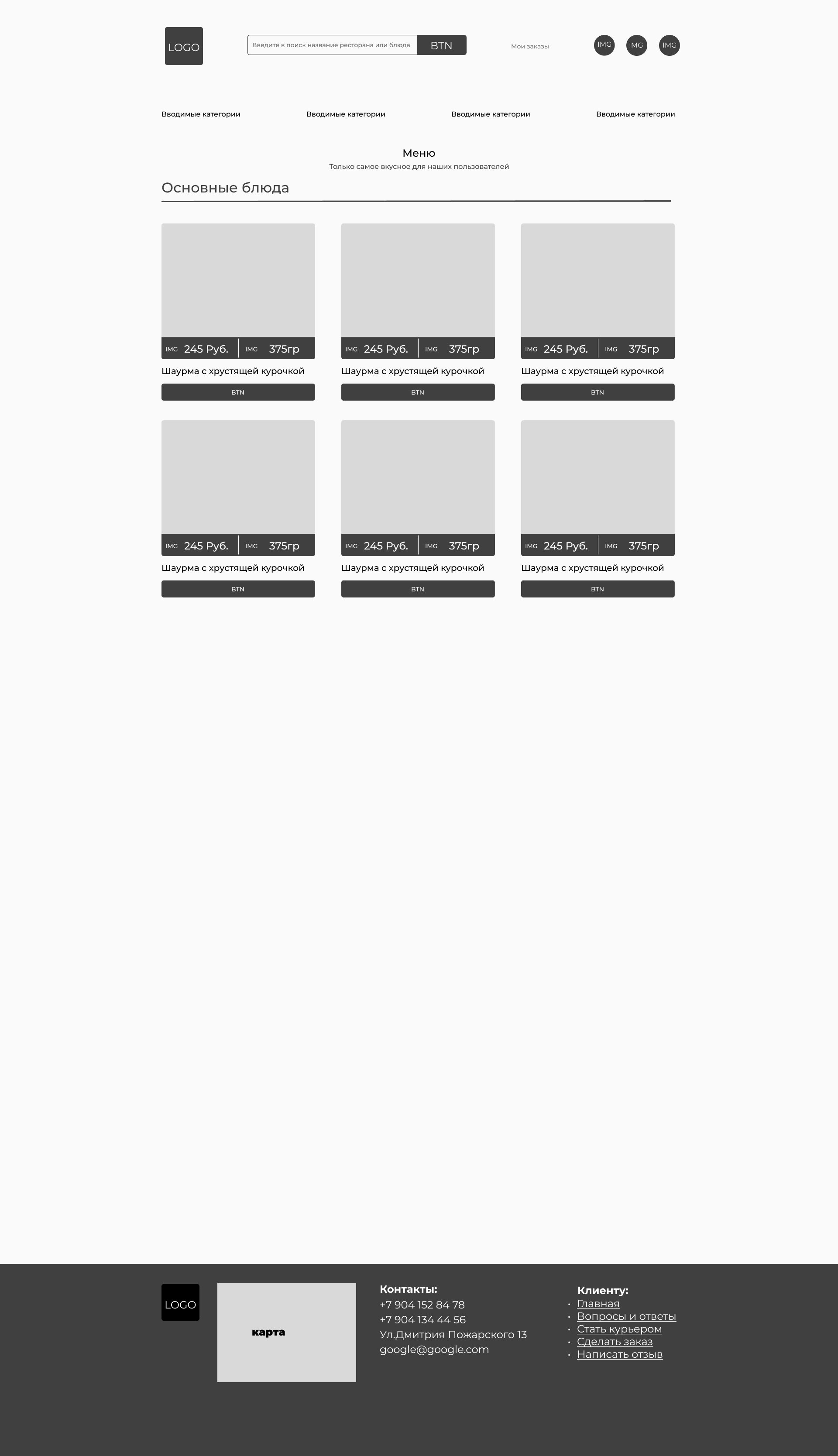


Рисунок 12 – Прототип страницы заведения

На рисунке 13 изображен прототип главной страницы. Можно сделать вывод о будущем расположении элементов и структуры страницы.

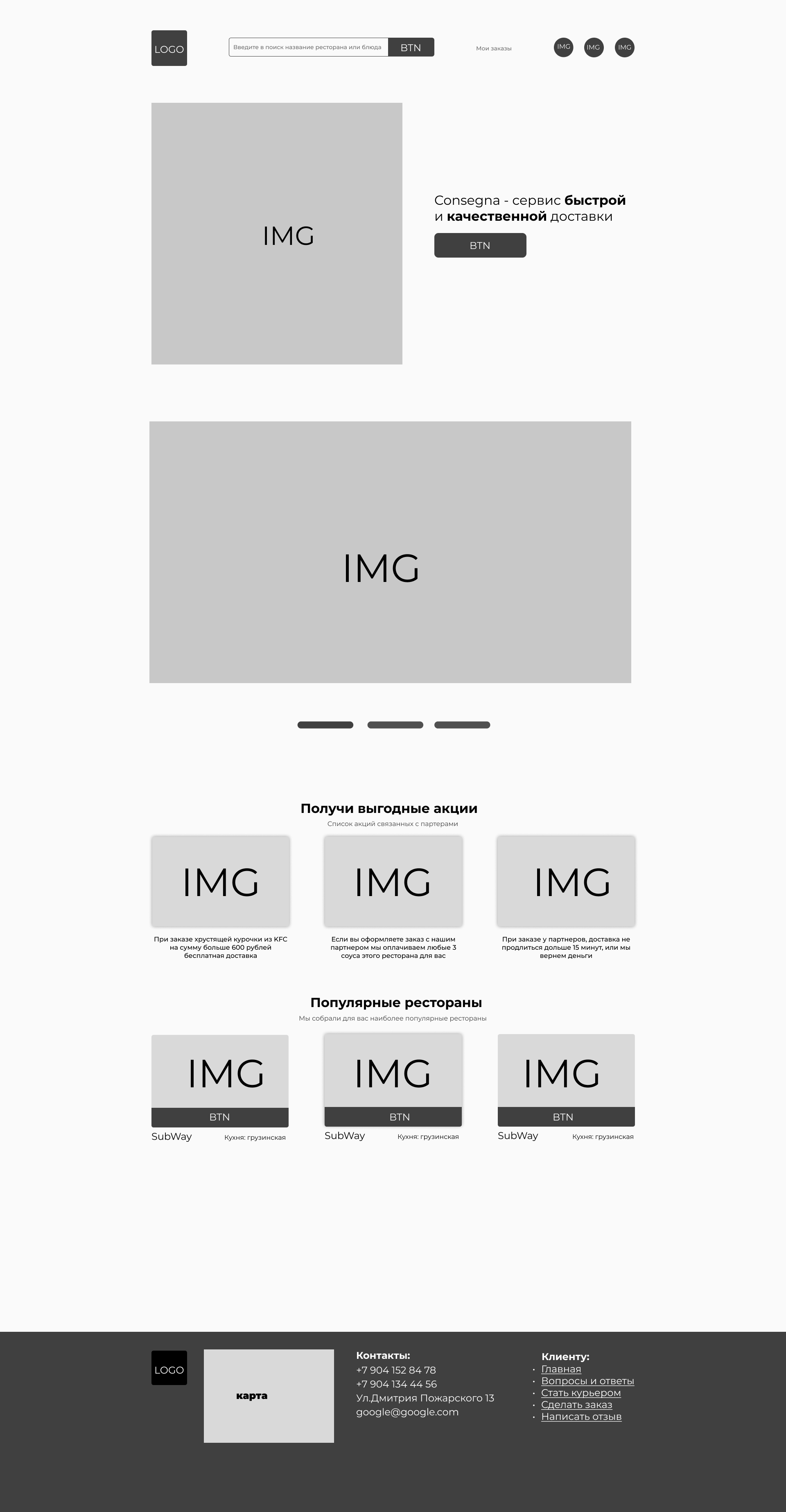


Рисунок 13 – Прототип главной страницы

### 4.4.2 Выбор цветовой гаммы и шрифтов

Шрифт: Montserrat шрифт без засечек, современный, легко читаемый, приятный для пользователя.

Основные цвета:

Цвет 1: FFD22D оттенок желтого. Желтый ассоциируется с удовольствием и оптимизмом. Этот цвет может повысить настроение и стимулировать креативность

Цвет 2: 404040 оттенок серого. Серый это благородный цвет, нейтральный.

Цвет 3: 000000 Черный цвет использован для текста.

Цвет 4: D91D1D оттенок красного. Призван акцентировать внимание на наиболее важных элементах сайта таких как кнопки выхода, удаление заказа.

### 4.4.3 Разработка элементов интерфейса

На рисунке 14 изображена карточка товара на странице заведения. Картинка и текст меняются в зависимости от выбранного ресторана.

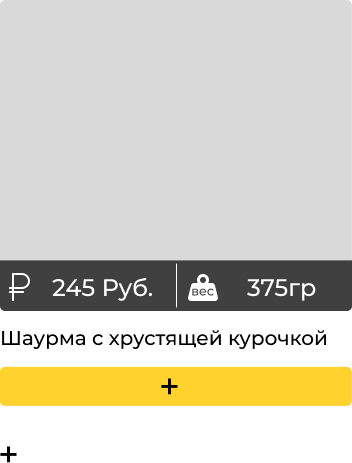


Рисунок 14 – Карточка товара

На рисунке 15 изображена карточка ресторана на главной странице.



Рисунок 15 – Карточка ресторана

### 4.4.4 Разработка дизайн макетов

На рисунке 16 изображена страница входа в информационную систему. Для входа используется номер телефона и пароль от аккаунта.



Рисунок 16 – Страница входа в информационную систему

На рисунке 17 изображена страница регистрации. Для регистрации используется Имя, Фамилия, Телефон, E-mail, Пароль и Повторный пароль.

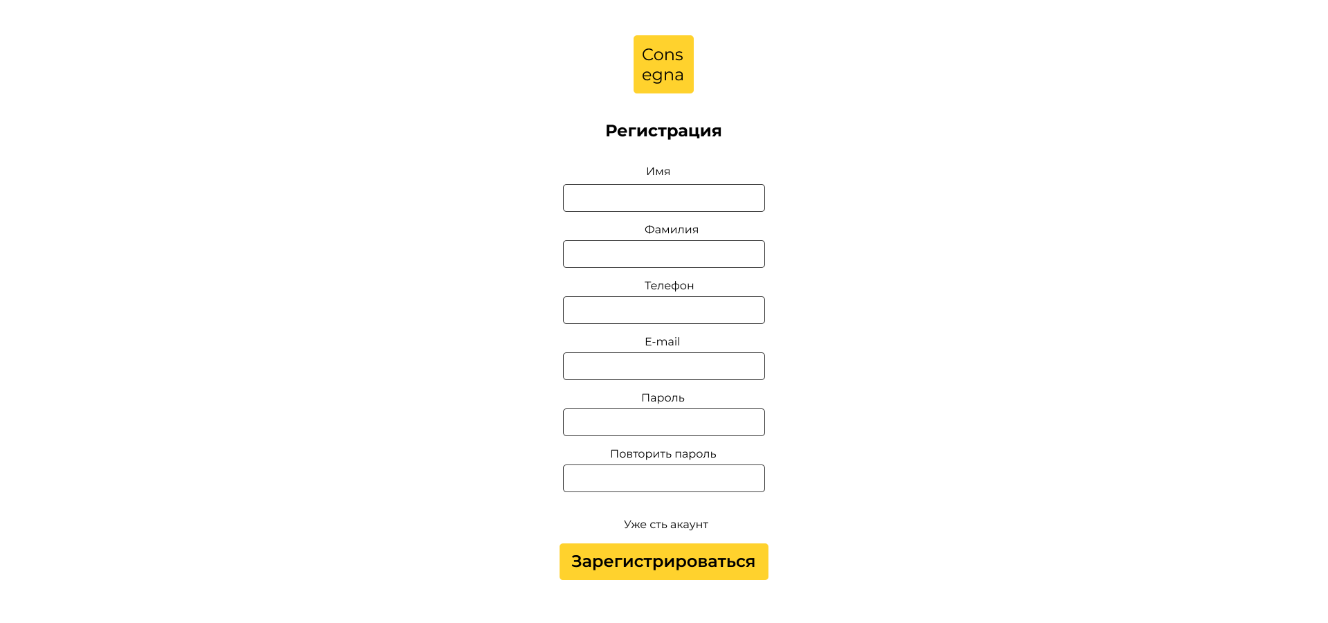


Рисунок 17 – Страница регистрации

На рисунке 18 изображена страница заведения. На этой странице можно заказать понравившиеся блюда.

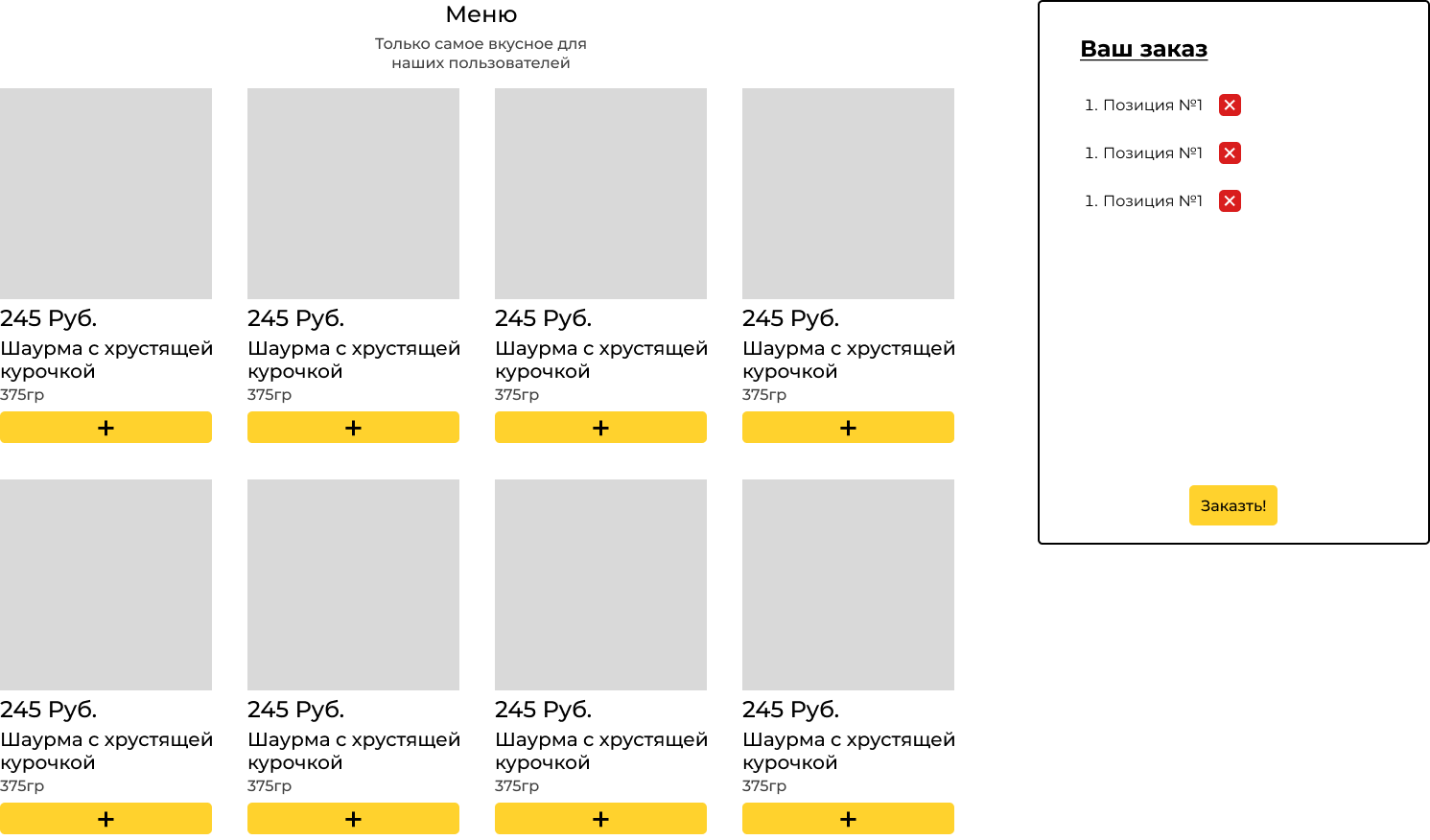


Рисунок 18 – Страница заведения

На рисунке 19 изображена главная страница. На этой странице можно ознакомится с ресторанами, которые зарегистрированы в приложении.

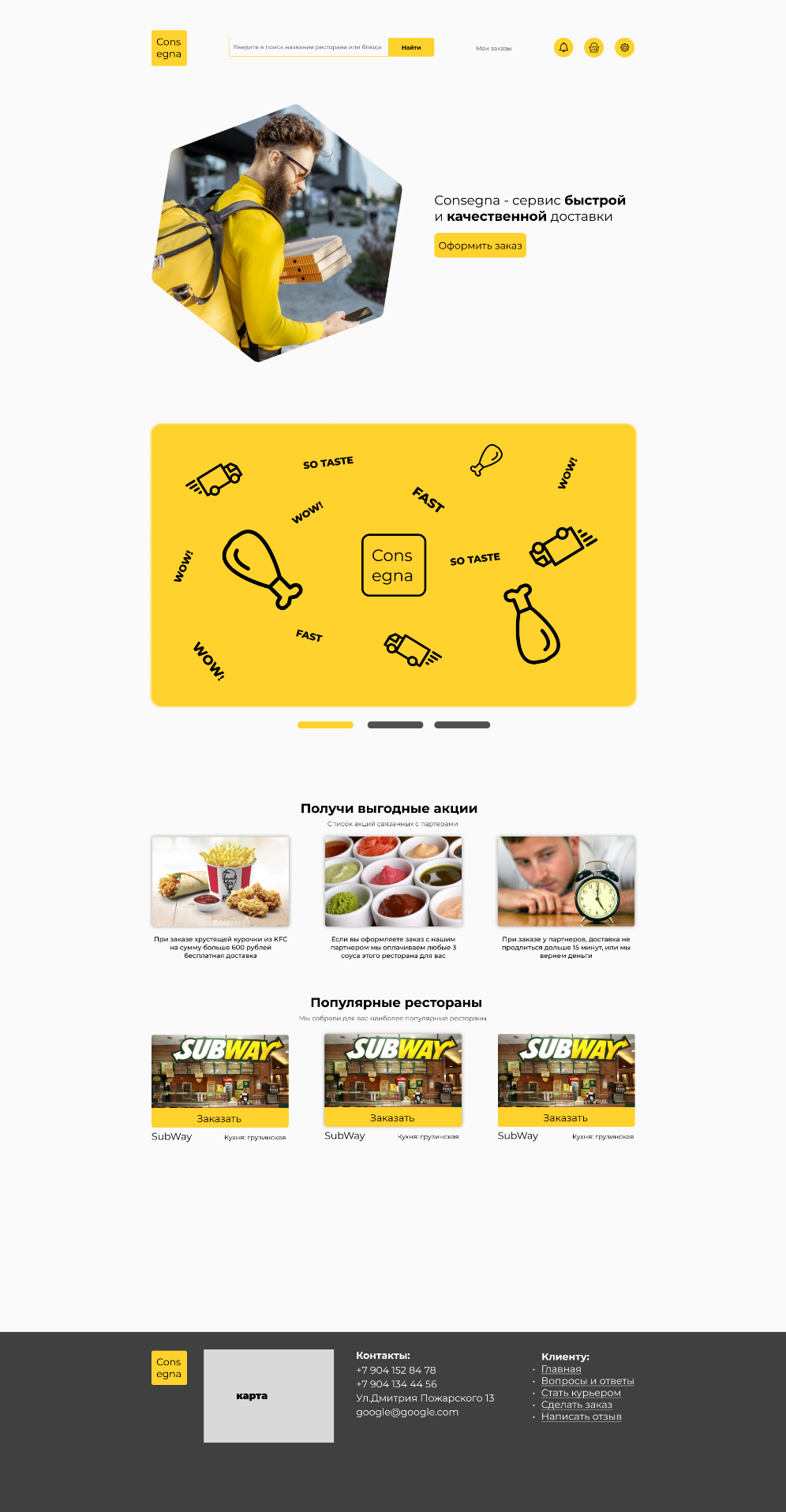


Рисунок 19 – Главная страница

На рисунке 20 изображена страница заказа. На этой странице можно ознакомится с вашим заказом, изменить его и выбрать способ оплаты.

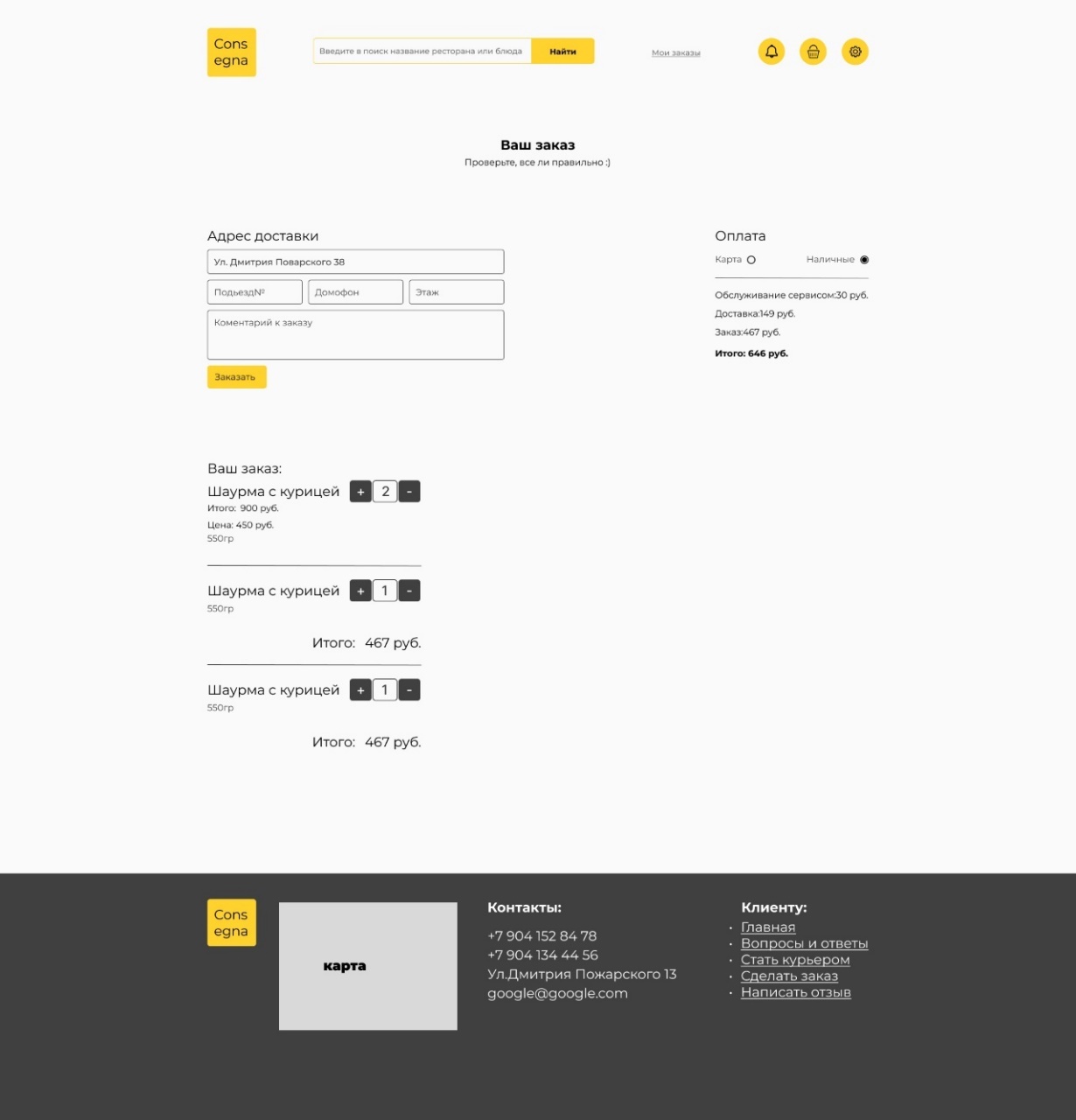


Рисунок 20 – Страница заказа

На рисунке 21 изображен футер. В секции футера находятся кнопки для перехода в AppStore и Google Play. Блок «Клиенту», в котором находиться навигационное меню по сайту. Логотип компании. Qr код для загрузки приложения с RuStore.



Рисунок 21 – Футер

На Рисунке 22 изображен хедер. В секции хедера находиться логотип. Поисковая строка с кнопкой для поиска по сайту и кнопка для настроек.

Рисунок 22 – Хедер

### 5 Разработка информационной системы

### 5.1 Разработка интерфейса информационной системы

На рисунке 23 расположен блок кода шапки, в котором находиться форма поиска, ссылка на страницу с заказами, кнопка чтобы посмотреть уведомления, кнопка чтобы посмотреть корзину и кнопка для модального окна.

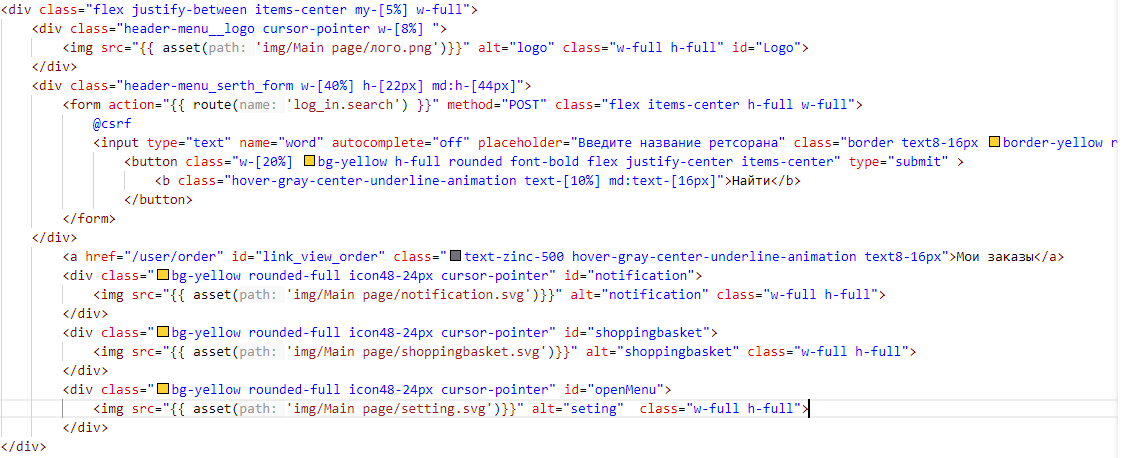


Рисунок 23 – Блок кода шапки сайта

На рисунке 24 расположен блок кода шапки, в котором расположен контент модульного окна для дополнительных возможностей пользователя.

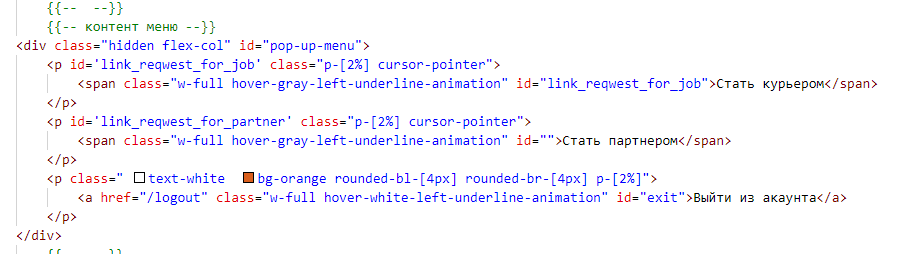


Рисунок 24 – Блок кода модульного окна

На рисунке 25 расположен блок кода файла конфигурации tailwind в котором находиться пользовательские настройки.

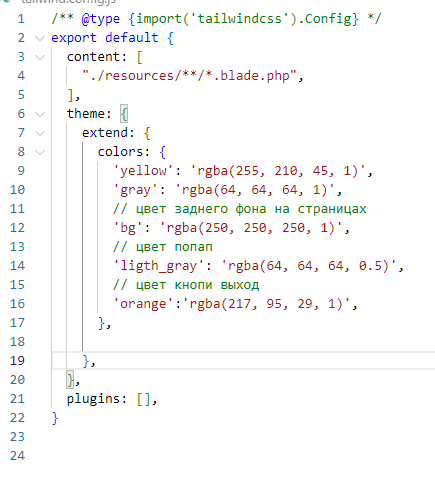


Рисунок 25 – Блок кода файла конфигурации tailwind

На рисунке 26 расположен блок кода c пользовательскими переменными, которые используются для ускорения разработки верстки.



Рисунок 26 – Блок кода c пользовательскими переменными

### 5.2 Разработка базы данных информационной системы

На рисунке 27 расположен блок кода с кодом создания таблицы пользователей в котором расположены атрибуты будущей таблицы.



Рисунок 27 – Блок кода с кодом создания таблицы пользователей

На рисунке 28 расположен блок кода с кодом модели для таблицы пользователей, в ней расположены связи и атрибуты, которые можно создавать и заполнять.

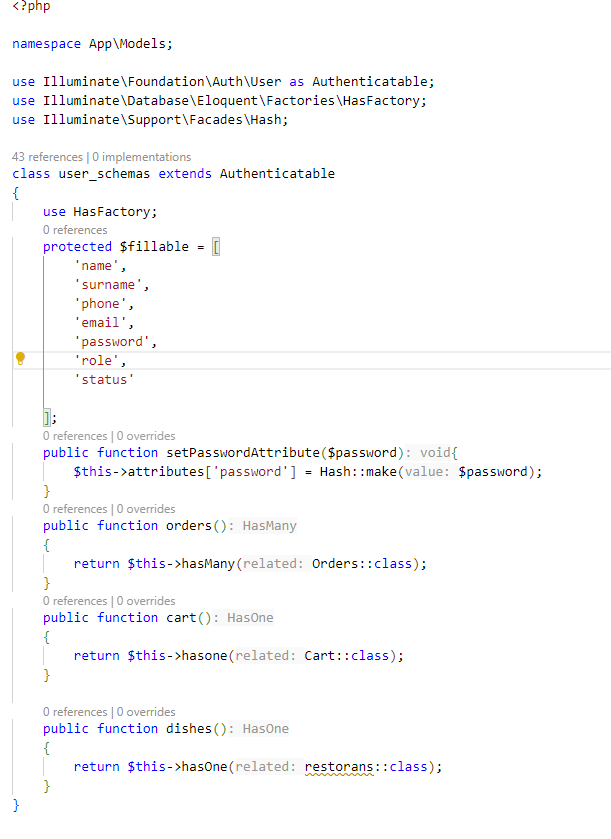


Рисунок 28 – Блок кода с кодом модели для таблицы пользователей

На рисунке 29 расположен блок кода с кодом посредника, который для каждого запроса проверяет аутентификацию и позволяет обращаться к таблице заказов чтобы можно было визуализировать заказы.

****

Рисунок 29 – Блок кода с кодом посредника

На рисунке 30 расположен блок кода, на котором расположен контроллер и функция для редактирования ресторана, проверка полей вводимых данных, и обновление старых.



Рисунок 30 – Блок кода с кодом посредника

### 5.3 Разработка информационной системы

На рисунке 31 расположен блок кода с меню CRUD, на котором расположены кнопки удаления, редактирования, удаления. Эти кнопки находиться на странице ресторана, на котором расположены все рестораны.



Рисунок 31 – Блок кода с меню CRUD

На рисунке 32 расположен блок кода контроллера, в котором находиться функции для редактирования и удаления ресторана.

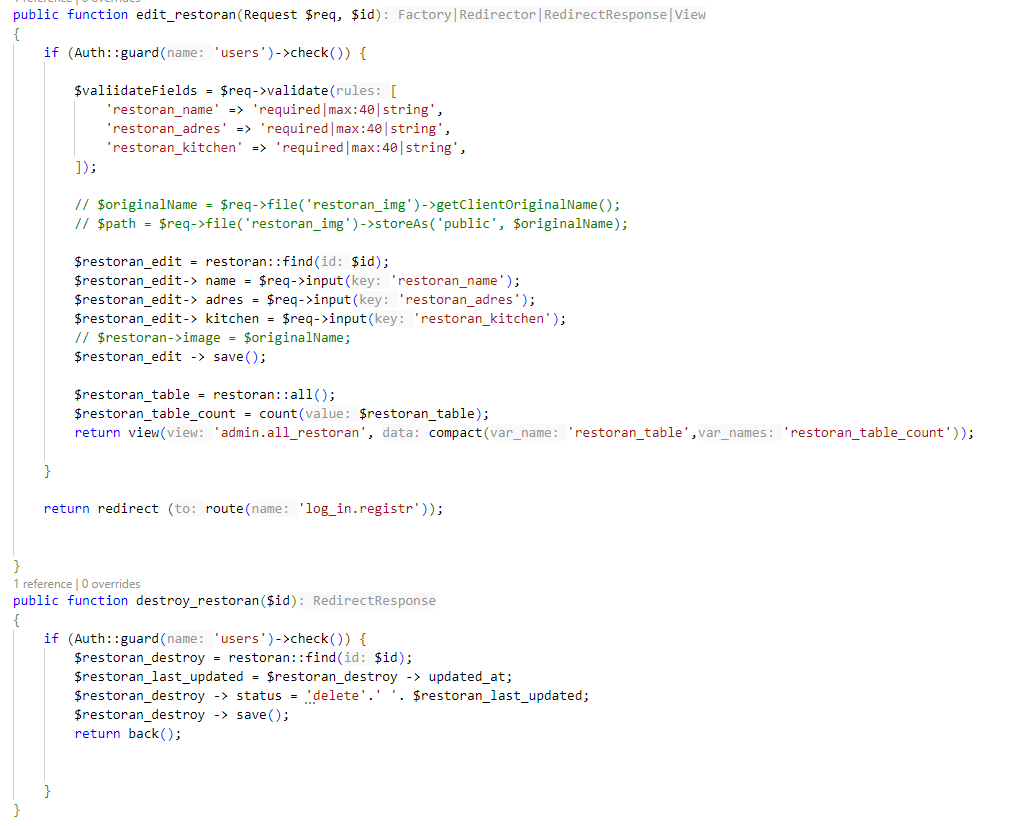


Рисунок 32 – Блок кода контроллера

На рисунке 33 расположен блок кода модульного окна, которое появляется и позволяет внести изменения в блюдо.

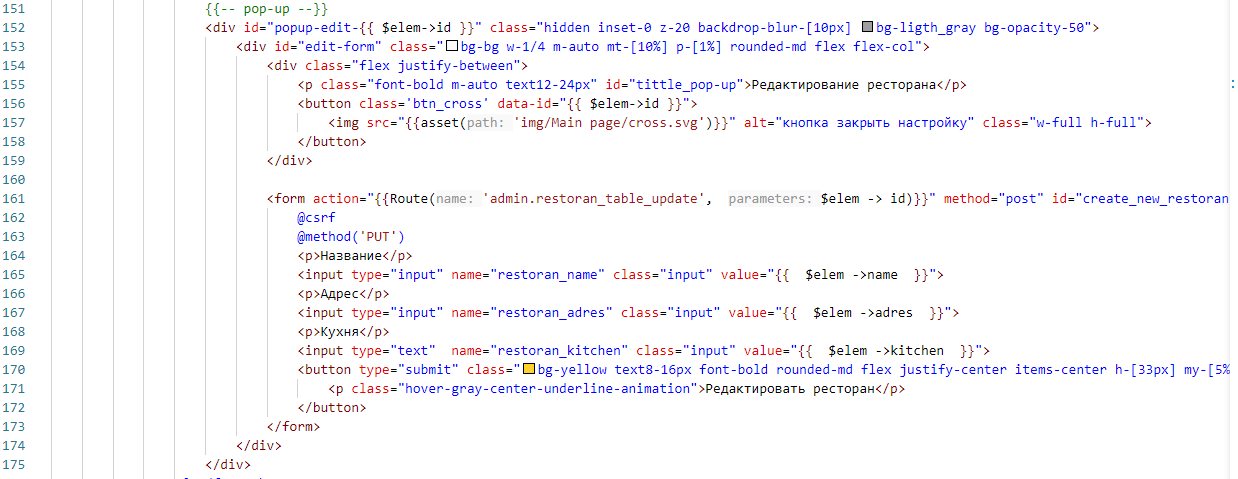


Рисунок 33 – Блок кода модульного окна

**5.4 Тестирование информационной системы**

На рисунке 34 представлен блок кода с тестом создания пользователя с неправильными данными. Данный тест провален так как данные неправильные.

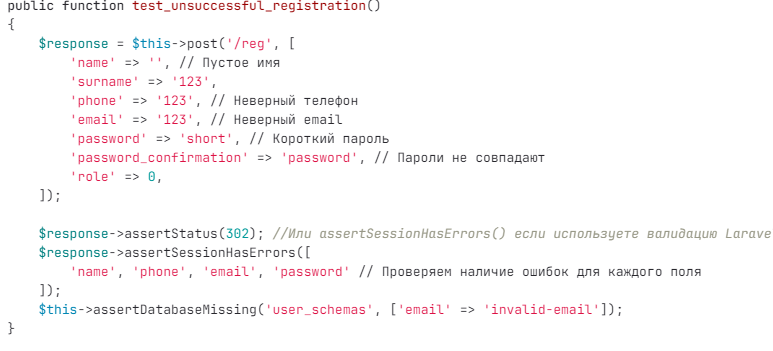


Рисунок 34 – Блок кода с тестом создания пользователя

На рисунке 35 представлен блок кода с тестом создания новой записи в корзине при помощи неавторизованного пользователя, с неправильными данными. Данный тест провален так как данные неправильные и пользователь не авторизован.

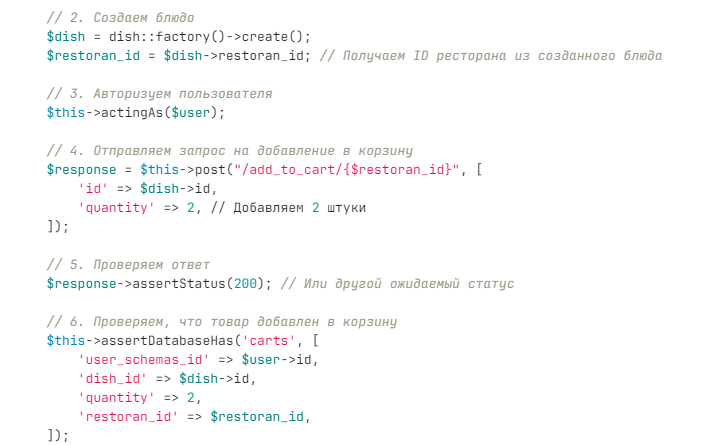


Рисунок 35 – Блок кода с тестом создания пользователя

**6 Документирование программного продукта**

**6.1 Руководство пользователя информационной системой**

Руководство пользователя – руководство, назначение которого – помощь пользователям в использовании информационной системой.

Информационная система сервис доставки еды предназначено для автоматизации бизнес-процесса формирования и осуществление доставки заказов еды из ресторанов.

Применение система способствует автоматизации процесса создания заказа, редактирования данных через администратора.

Основными возможностями сервис доставки еды являются:

Регистрация пользователей, выбор поправившегося ресторана и выбор блюда, формирование заявки с указанием адреса доставки и комментария к заказу.

Пользователи сервис доставки еды должны иметь навыки в работе с применением технических и программных средств уровня Windows 7 и выше или их аналогов.

Для работы с сервис доставки еды необходим один из браузеров:

1) Google Chrome,

2) Яндекс.Браузер,

3) Firefox,

4) или любой другой браузер на основе Chromium.

Для доступа к сервису доставки еды требуется подключение к интернету.

При запуске приложения проходит проверка на авторизацию, если пользователь не авторизован или выходил из приложения ранее ему загружается станица аутентификации рисунок 38

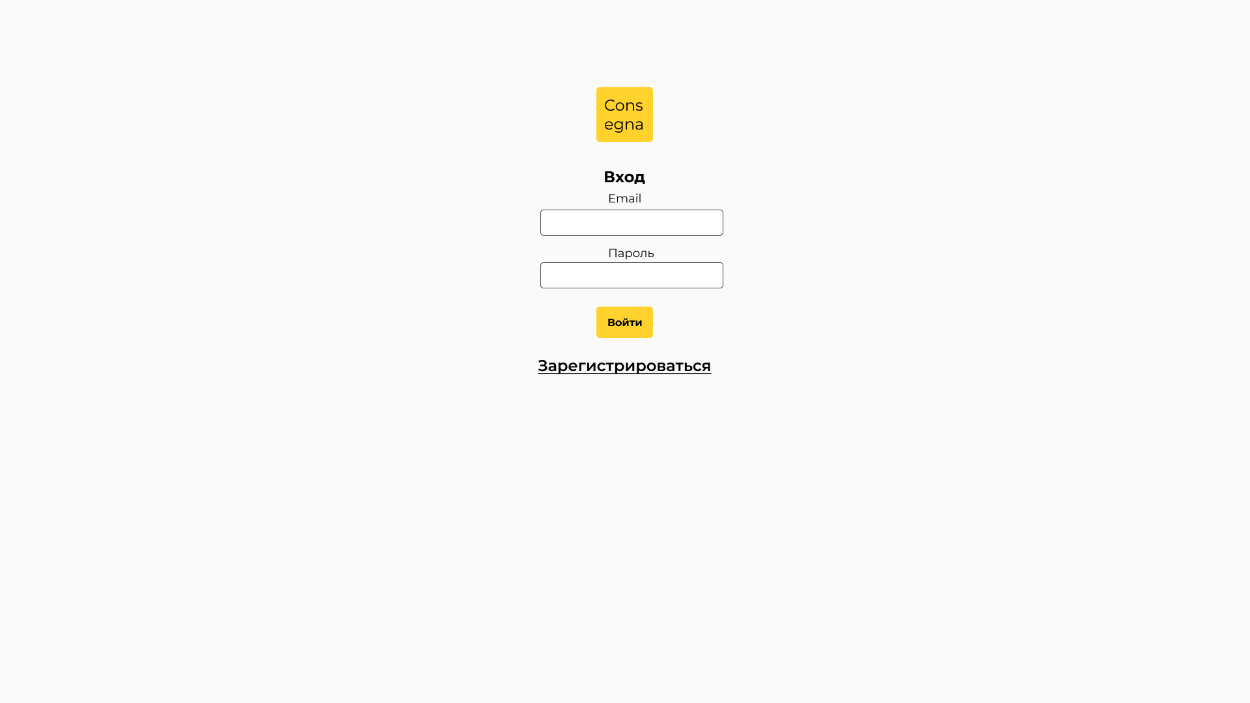


Рисунок 38 – Станица аутентификации

После успешной аутентификации пользователь попадает на главную страницу рисунок 39.

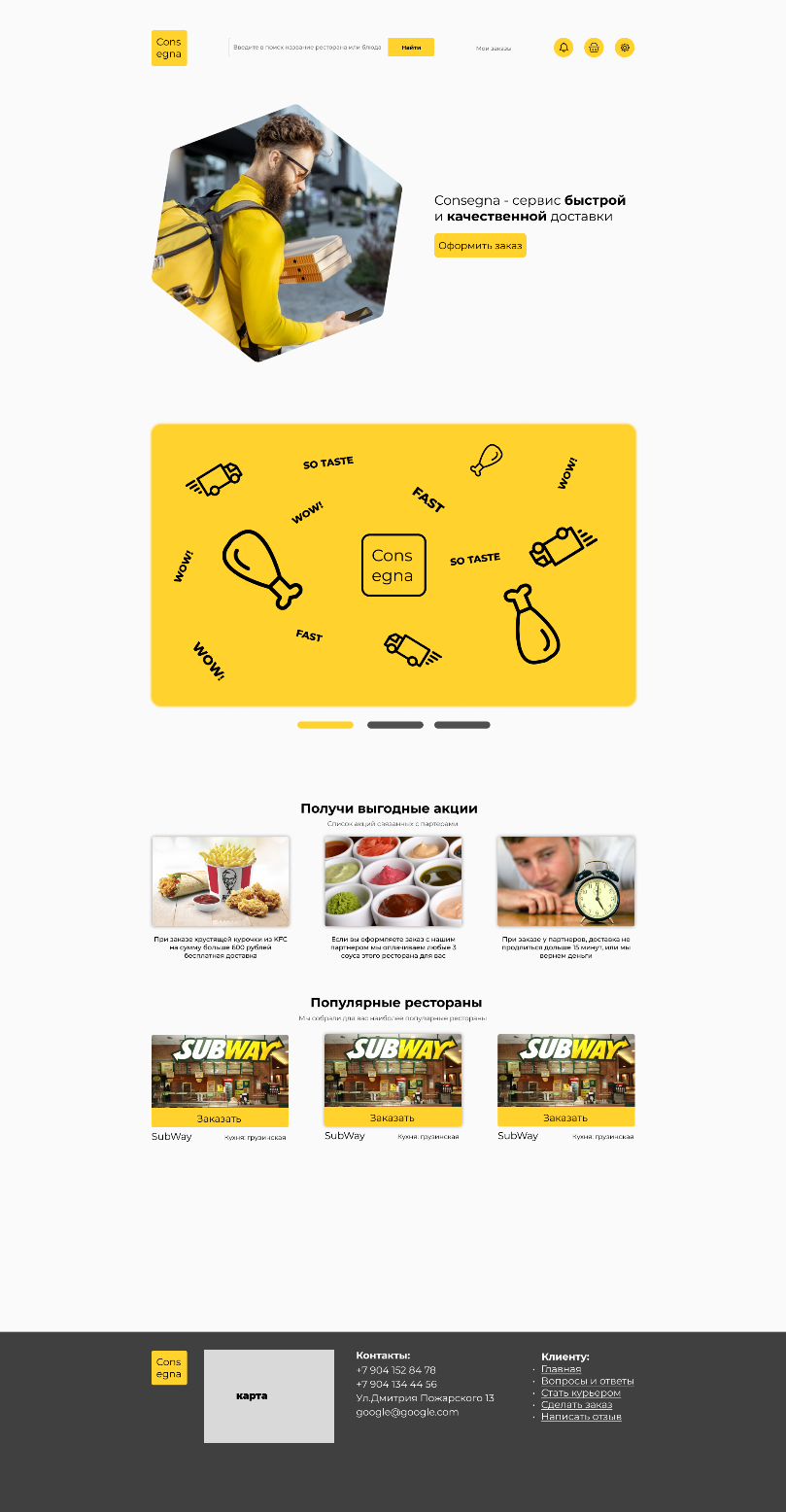


Рисунок 39 – Главная страница

После выбора ресторана пользователь попадает на страницу заведения рисунок 40.

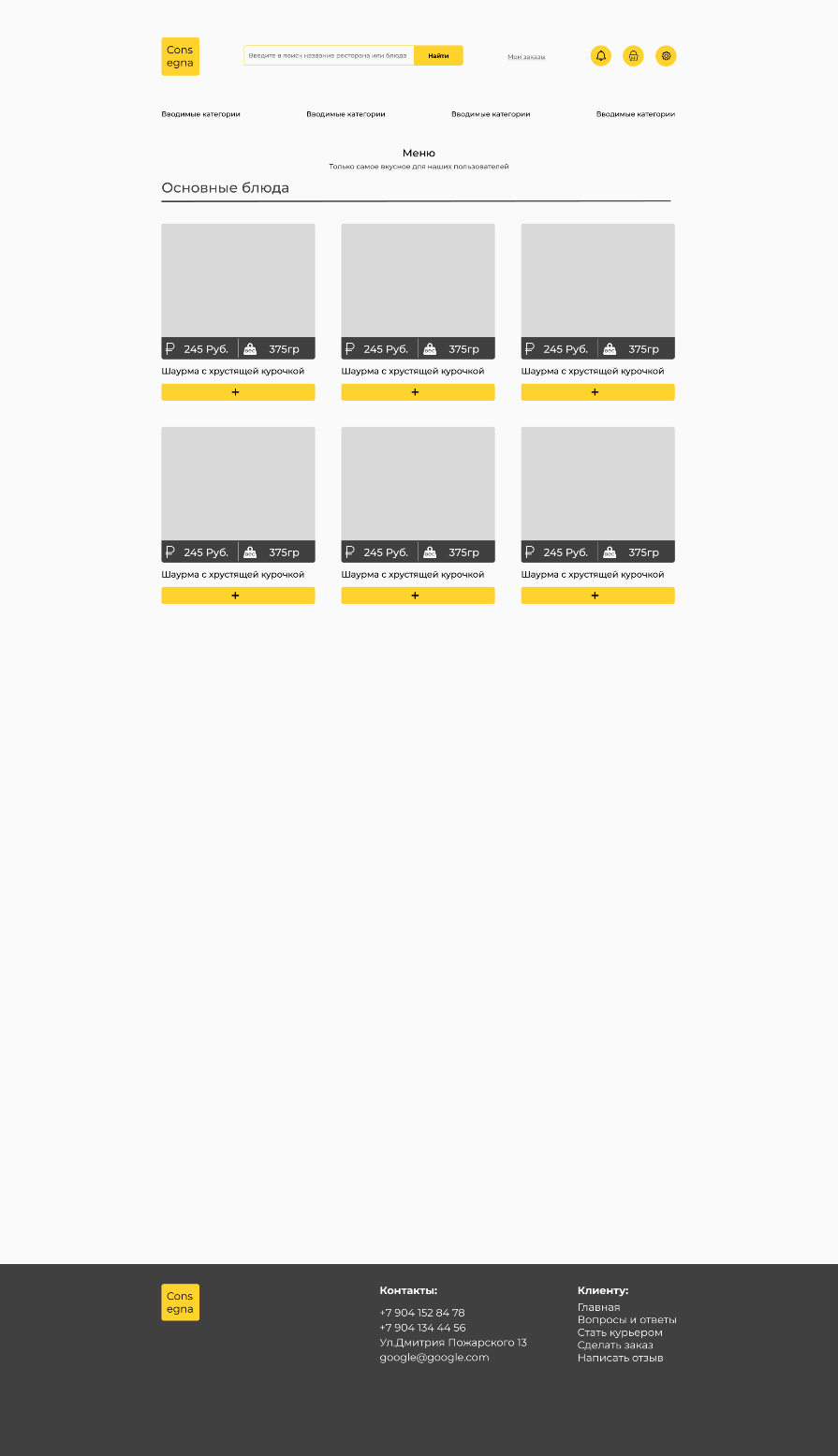


Рисунок 40 – Страница заведения

После выбора ресторана пользователь попадает на страницу формирования заказа рисунок 41.

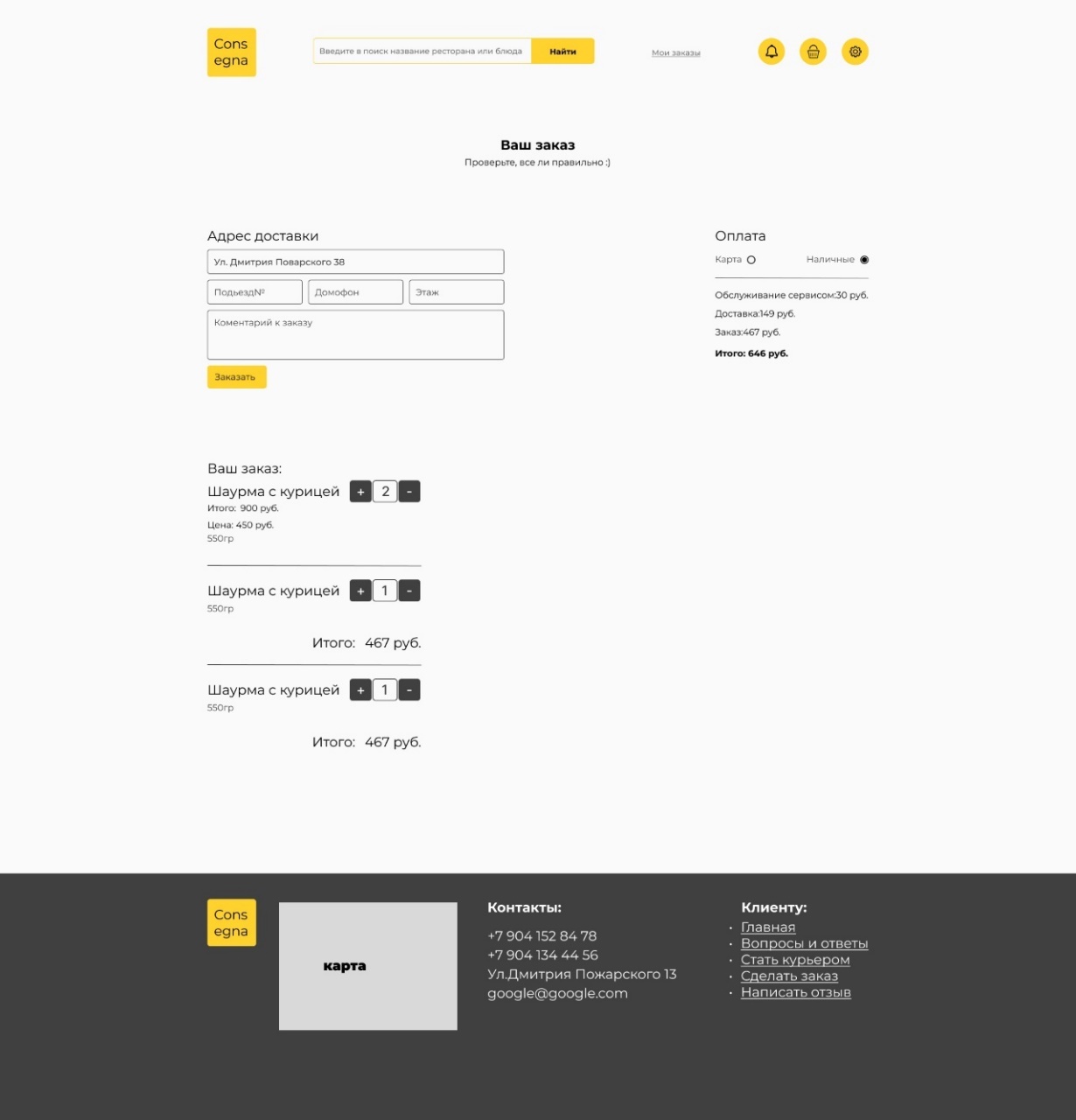


Рисунок 41 – Страница формирования заказа

На рисунке 42 представлено модальное окно с возможностью выйти из аккаунта, посмотреть заказы и стать курьером.

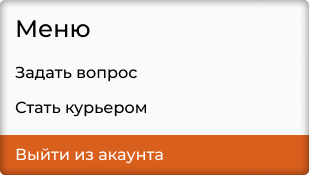


Рисунок 42 – Модальное окно

**6.2 Руководство администратора веб-приложения**

Руководство администратора – руководство, назначение которого – помощь администратору в использовании веб-приложения.

Веб-приложение сервис доставки еды предназначено для формирования и осуществление доставки заказов еды из ресторанов.

Применение приложения способствует автоматизации процесса создания заказа, редактирования данных через администратора.

Основными возможностями сервис доставки еды являются:

Регистрация пользователей, выбор поправившегося ресторана и выбор блюда, формирование заявки с указанием адреса доставки и комментария к заказу.

Пользователи сервис доставки еды должны иметь навыки в работе с применением технических и программных средств уровня Windows 7 и выше или их аналогов. При работе с базой данных приложения необходимо придерживаться правил безопасности и конфиденциальности, а именно: не разглашать личную информацию пользователей другим лицам, производить удаление, изменение данных только после установки режима отладки на веб-приложении, не производить изменение в базе данных напрямую без разрешения владельца, страницы для администратора специально не адаптивны так как работа с базой данных разрешается исключительно на компьютере во избежание случайных удалений.

Для работы с сервис доставки еды необходим один из браузеров:

5) Google Chrome,

6) Яндекс.Браузер,

7) Firefox,

8) или любой другой браузер на основе Chromium.

Для доступа к сервису доставки еды требуется подключение к интернету.

При запуске приложения проходит проверка на авторизацию, если администратор не авторизован или выходил из приложения ранее ему загружается станица аутентификации рисунок 43

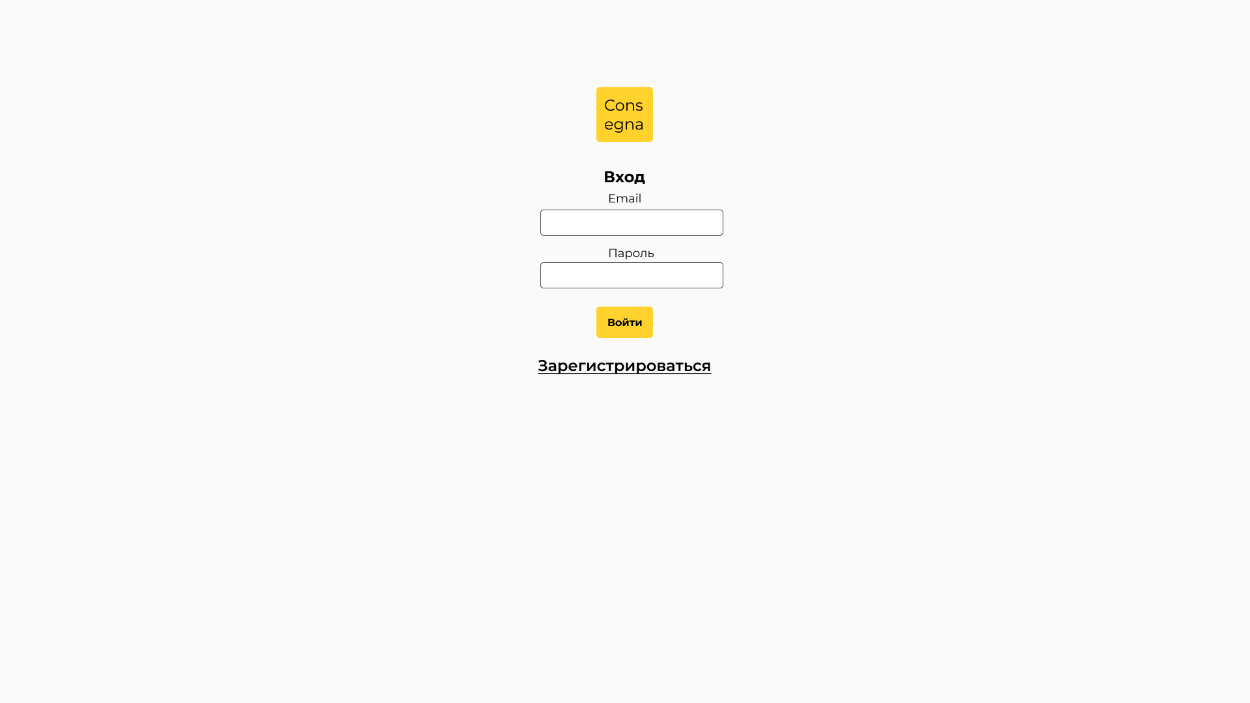


Рисунок 43 – Страница входа администратора

После успешной аутентификации администратор попадает на главную страницу администратора рисунок 44.

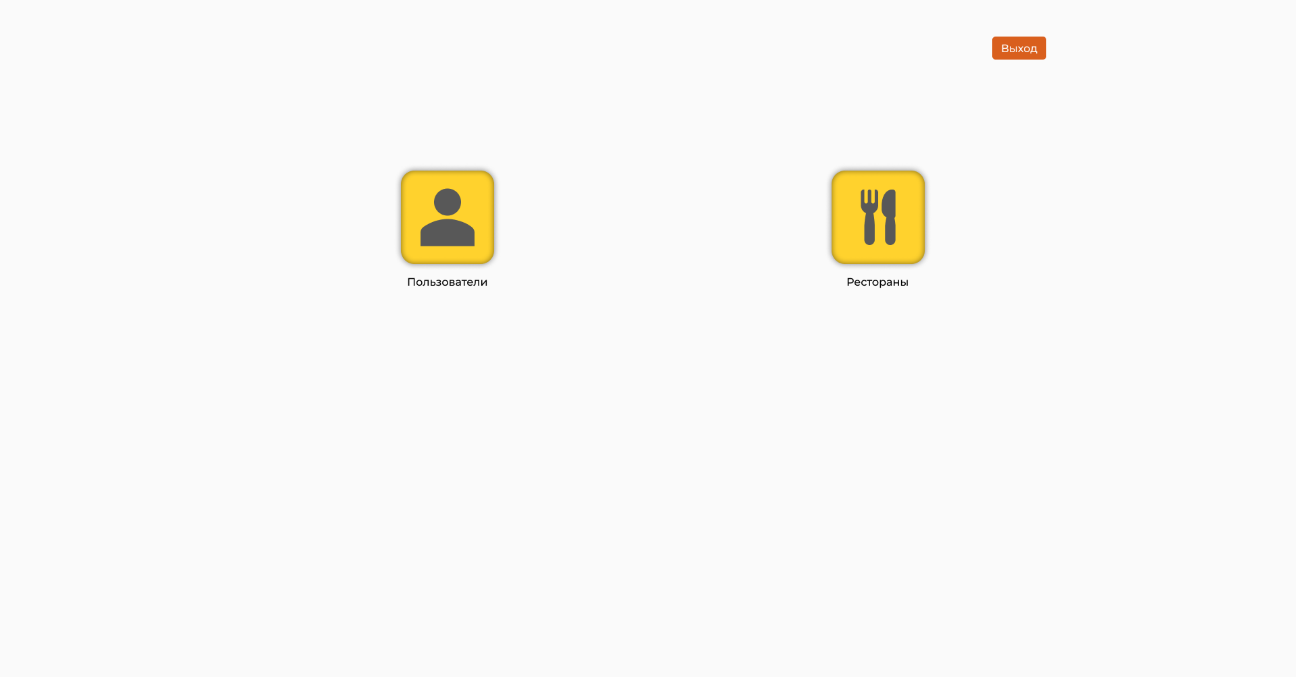


Рисунок 44 – Главная страница администратора

Далее администратор выбирает с чем он будет взаимодействовать с пользователями, или ресторанами. Если администратор решает взаимодействовать с ресторанами рисунок 45

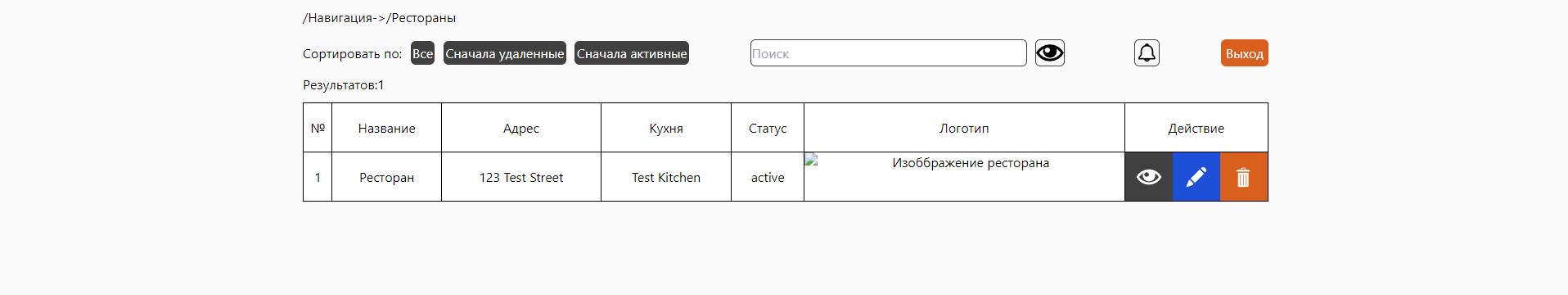


Рисунок 45 – База ресторанов

Администратор решает добавить новое блюдо в уже существующий ресторан рисунок 46

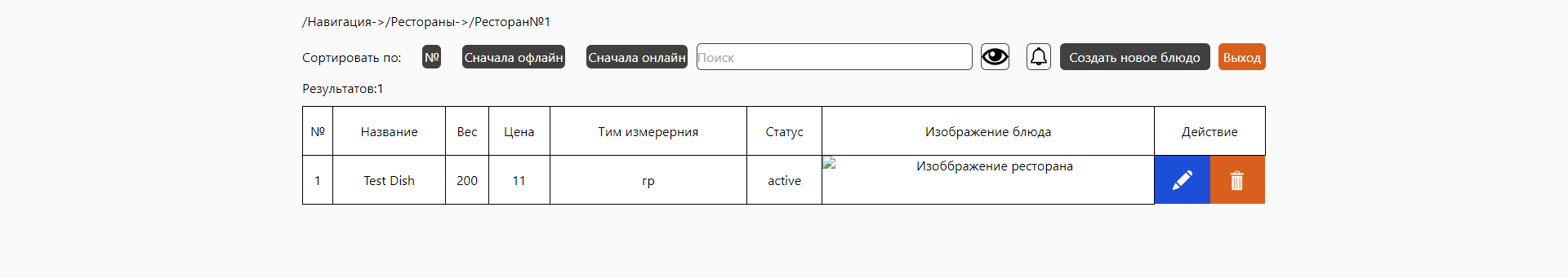


Рисунок 46 – рисунок базы данных блюд

**6.3 Руководство курьера веб-приложения**

Руководство курьера – руководство, назначение которого – помощь курьера в использовании веб-приложения.

Веб-приложение сервис доставки еды предназначено для формирования и осуществление доставки заказов еды из ресторанов.

Применение приложения способствует автоматизации процесса создания заказа, редактирования данных через администратора, работа с заказами через курьера.

Основными возможностями сервис доставки еды являются:

Регистрация пользователей, выбор поправившегося ресторана и выбор блюда, формирование заявки с указанием адреса доставки и комментария к заказу.

Пользователи сервис доставки еды должны иметь навыки в работе с применением технических и программных средств уровня Windows 7 и выше или их аналогов. При работе с базой данных приложения необходимо придерживаться правил безопасности и конфиденциальности, а именно: не разглашать личную информацию пользователей другим лицам, доставлять заказ в целостности и сохранности.

Для работы с сервис доставки еды необходим один из браузеров:

9) Google Chrome,

10) Яндекс.Браузер,

11) Firefox,

12) или любой другой браузер на основе Chromium.

Для доступа к сервису доставки еды требуется подключение к интернету.

При запуске приложения проходит проверка на авторизацию, если курьера не авторизован или выходил из приложения ранее ему загружается станица аутентификации рисунок 47

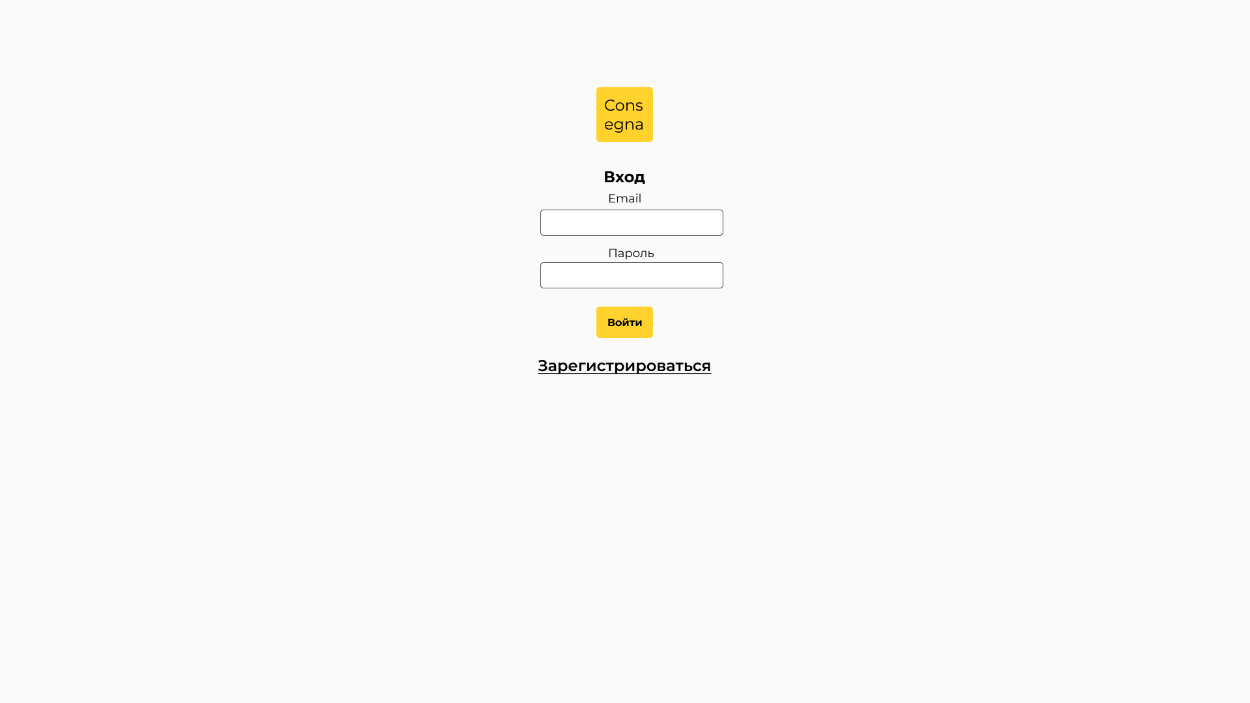


Рисунок 47 – Страница входа курьера

После успешной аутентификации курьер попадает на главную страницу курьера рисунок 48.

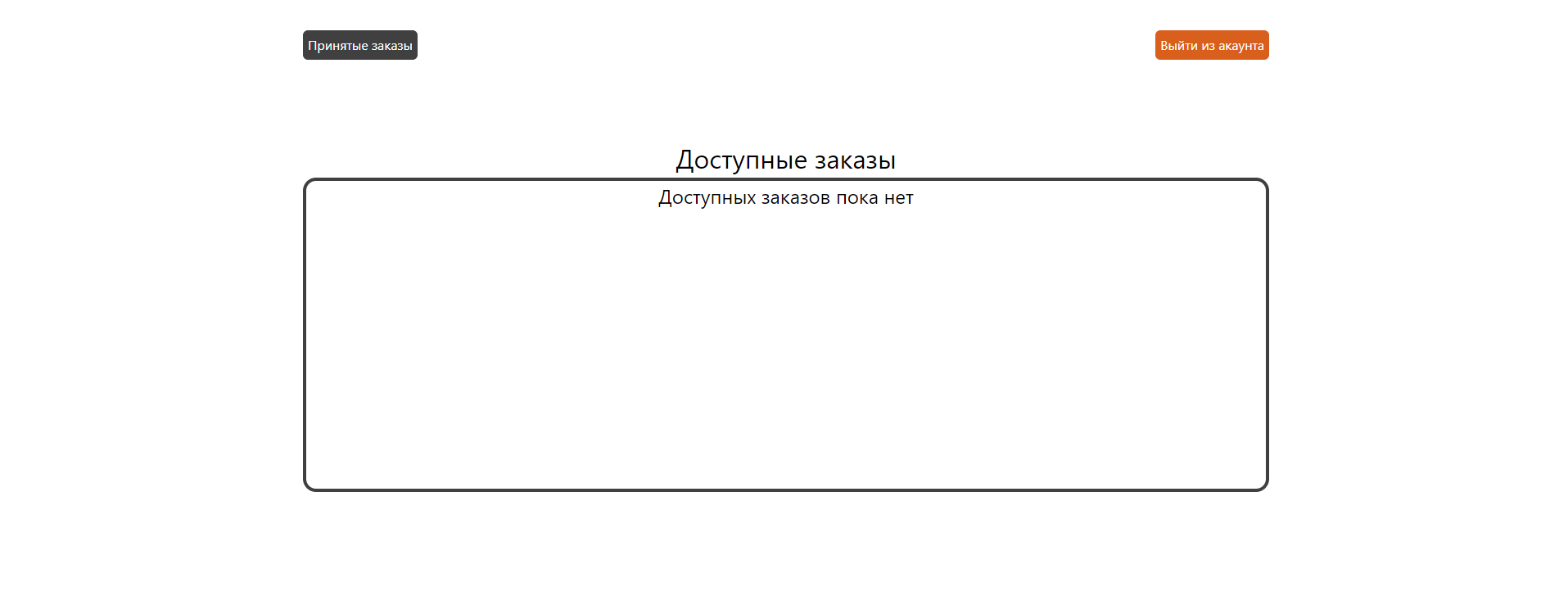


Рисунок 48 – Главная страница курьера

Далее курьер выбирает заказ из доступных. Если заказов нет или курьер уже выбрал для себя заказ и не закончил его, то доступных заказов не будет. рисунок 49 секция «принятые заказы»

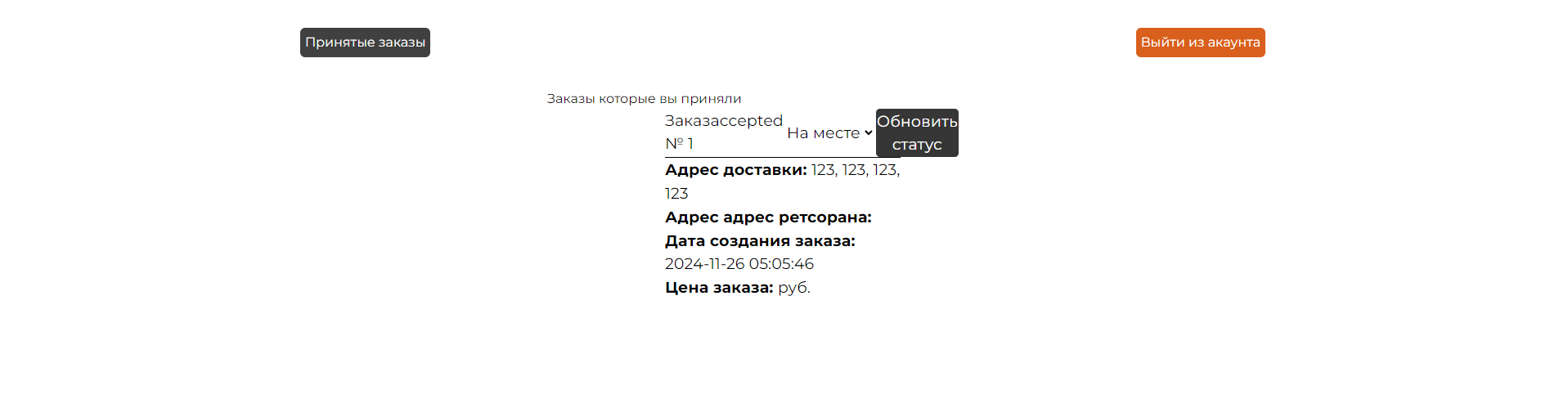


Рисунок 49 – Принятые заказы

Курьер решает обновить статус заказа рисунок 50

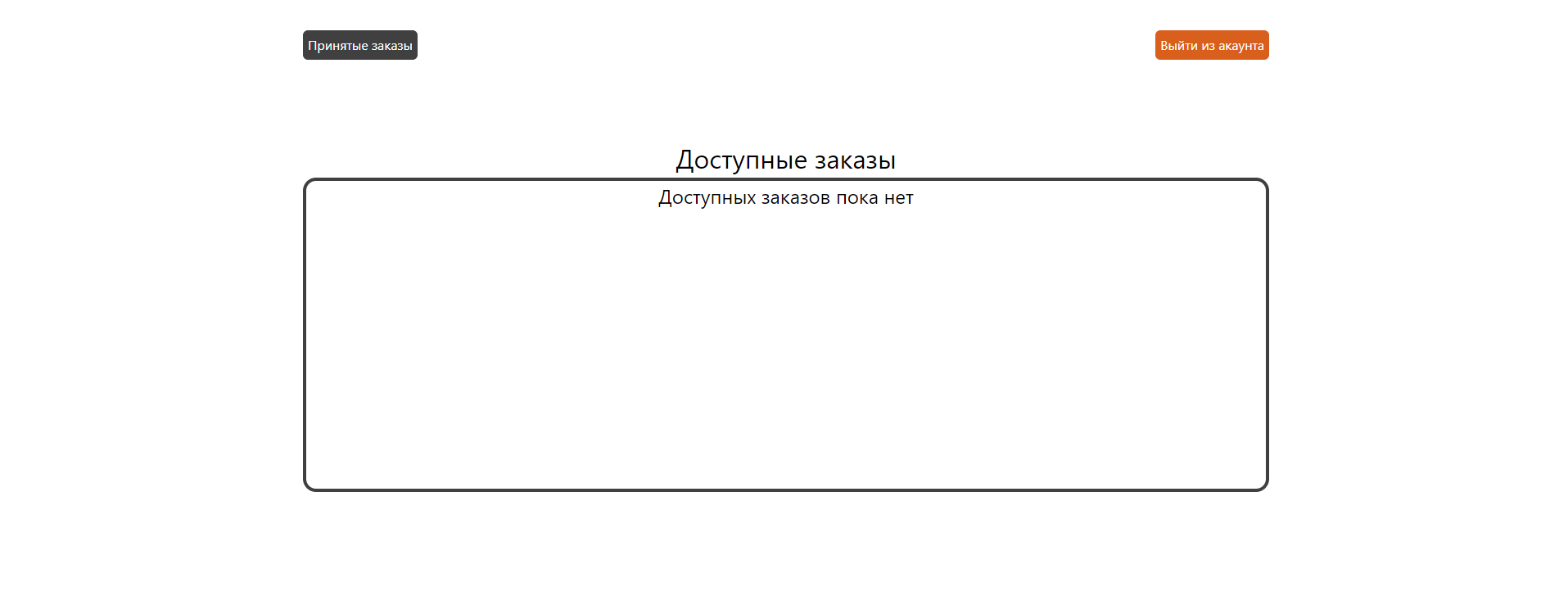


Рисунок 50 – Пустое поле так как заказ был доставлен

# Заключение

В процессе создания веб-приложения доставки еды, было проведено исследование предметной области, были выполнены цели и задачи, которые были поставлены в начале работы, а именно:

– создание дизайна для веб-приложения доставка еды

– проектирование и создание базы данных для веб-приложения доставка еды

– верстка дизайна веб-приложения

– добавление функционала

– привязка базы данных к веб-приложению

Задачи были выполнены в полном объеме, с использованием современных средств веб разработки.

Целью, данной курсового проекта, является создание веб-приложения по доставке еды, которая реализует процесс регистрации, аутентификации пользователя, выход из акаунта, процесс формирования заказа, а также его регистрация со стороны пользователя. Со стороны администратора редактирование, удаление данных пользователей, добавление, изменение удаление, добавление, данных ресторанов и блюд для меню этого ресторана, а также просмотр данных о заказах конкретного пользователя. Со стороны курьера выбор из доступных не занятых заказов. Просмотр информации о выбранном заказе.

В дальнейшем, во время выполнения проекта уже как дипломной работы планируются изменения относительно имеющегося функционала, добавление категории клиентов рестораторов, чтобы снизить нагрузку на администратора, также планируется улучшение формы поиска для отображения не только ресторанов, содержавших подходящее названия, но и ресторанов, содержащих подходящие блюда. Настройка системы партнеров, добавление возможности редактировать акции, категорию партнеров, карусель на главной странице. Реализовать связь межу пользователем и курьером, который занят доставкой заказа. Повысить безопасность приложения.

# Список используемых источников

1. dasreda.ru–Деловаясреда–URL:https://dasreda.ru/learn/blog/article/ 2398-kak -organizovat-biznes-po-dostavke-edy1e131748661 (дата обращения: 17.03.2024). – Текст: электронный.
2. Laravel – Routing – URL: https://laravel.com/docs/11.x/routing (дата обращения: 18. 03.2024). – Текст: электронный.
3. https://laravel.su/. – Controllers– URL: https://laravel.com/docs/11.x/ Controllers (дата обращения: 18. 03.2024). – Текст: электронный.
4. https://laravel.su/. – Blade Templates– URL: https://laravel.com/docs/11.x/ Blade Templates (дата обращения: 18. 03.2024). – Текст: электронный.
5. <https://habr.com/>.–Хабр.–URL:https://habr.com/ru/articles/291818/(дата обращения: 19.04.2024). – Текст: электронный.
6. https://laravel.su/. – Middleware– URL: https://laravel.com/docs/11.x/ Middleware (дата обращения: 19. 03.2024). – Текст: электронный.
7. https://laravel.su/. – CSRF Protection– URL: https://laravel.com/docs/11.x/ CSRF Protection (дата обращения: 24. 03.2024). – Текст: электронный.
8. https://laravel.su/. – Views– URL: https://laravel.com/docs/11.x/ Views (дата обращения: 10. 04.2024). – Текст: электронный.
9. https://laravel.su/. – HTTP Session – URL: https://laravel.com/docs/11.x/ HTTP Session (дата обращения: 11. 04.2024). – Текст: электронный.
10. https://laravel.su/. – Validation – URL: https://laravel.com/docs/11.x/ Validation (дата обращения: 13. 04.2024). – Текст: электронный.

# Приложение А Техническое задание

**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
|  |  |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ИНФОРМАЦИОННАЯ СИСТЕМА “ДОСТАВКА ЕДЫ”

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.С. Александрова) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (В.А. Кузнецов) |

Иркутск 2024

**1. Общие сведения**

Наименование работы: информационная система сервис по доставке еды.

Исполнитель: студент иркутского авиационного техникума, отделения ИКТ, группы ВЕБ-21-2, Кузнецов.В.А.

Разработка информационной системы проходит в рамках курсового проекта по МДК.05.02 «Разработка кода информационных систем» на основании приказа №53-у от 18.09.2024 г.

Сроки разработки информационной системы с 20.09.2024 по 04.12.2024 года.

**1.2 Цели и задачи**

Назначение информационной системы «доставка еды» заключается в автоматизации бизнес-процесса формирования заказа и подбора персонала для доставки еды до получателя. Предоставлении пользователю возможности выбирать понравившийся ресторан в интерфейсе системы, выбирать блюда для формирования личной корзины пользователя, для последующего оформления заказа и оплаты.

* Мониторинг выполнения заказов
* Автоматизация процессов создания нового ресторана
* Автоматизация процесса наполнения ресторана блюдами

**2 Основания для разработки**

**2.1. Нормативные документы**

Документ основывается на следующих нормативных документах:

ГОСТ 34.602-2020 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы".

**2.2. Проектные документы**

Проектные документы включают:

* Пояснительную записку.

**3. Назначение системы**

**3.1. Общее описание**

Информационная система «Доставки еды» предназначенная для автоматизации бизнес-процессов формирования заказа и создания новых ресторанов, формирования отчетов для администрации ресторана. Создаётся для преимущественно средних и крупных организаций которые могут позволить себе большую рекламную компанию для привлечения новых курьеров.

**3.2. Преимущества и новизна**

* Интуитивно понятный интерфейс
* Возможность составлять отчеты для администрации ресторана
* Интеграция со сторонними сервисами

**4.Требования к системе**

**4.1. Функциональные требования**

* Формирование заявки
* Возможность поиска ресторана
* Управление пользователями
* Управление ресторанами
* Управление блюдами
* Составления отчетов
* Мониторинг выполнения заказов
* Управление заказами
* Уведомления об статусе выполнении заказа
* Уведомления об одобрения заказа
* Отправлять заявки на партнерство
* Управлять заявками на партнёрство

**4.2. Технические требования**

1. Производительность:
   * Обработка до 100 задач одновременно.
   * Время отклика системы не более 2 секунд при загрузке данных.
2. Надежность:
   * Доступность системы не менее 99,5% в год.
3. Безопасность:
   * Аутентификация пользователей через LDAP.
   * Шифрование данных на уровне хранения.

**4.3. Эксплуатационные требования**

1. Удобство использования:
   * Дружественный пользовательский интерфейс.
2. Интеграция:
   * Возможность интеграции с сторонними сервисами для авторизации в системе.

**5. Требования к техническому обеспечению**

**5.1. Оборудование**

Минимальные системные требования для сервера:

1) i7-8700T 2.4-4.0 ГГц, 6 ядер,

2) 12 потоков, 16 Гб RAM,

3) 512 Гб NVME SSD

4) Операционная система: Windows

Минимальные системные требования для клиента:

1) i7-8700T 2.4-4.0 ГГц, 6 ядер,

2) 12 потоков, 4 Гб RAM,

3) Операционная система: Windows

4) Интернет 5 мб\сек

5) Включенная обработка JavaScript

**5.2. Сетевые требования**

Сеть: Доступ в Интернет со скоростью не менее 5 Мбит/с.

Сетевые протоколы: Поддержка TCP/IP, HTTP.

**6. Требования к программному обеспечению**

**6.1. Программные компоненты**

* Операционная система: Серверная версия ОС Windows Server.
* Базы данных: MySQL версии не ниже 10.0.
* Программное обеспечение: Веб-сервер Apache , интерпретатор PHP 7.4 или выше.

**6.2. Интерфейсы**

API: RESTful API для интеграции с другими системами.

Интерфейс пользователя: Веб-интерфейс с поддержкой браузеров Chrome, Firefox, Edge.

В процессе формирования заказа пользователю предлагается заполнить поля: дом, улица, если есть, номер квартиры и подъезда. выбрать способ оплаты: наличными или картой, и написать поясняющий комментарий к заказу в котором пользователь может вписать комментарий который увидит курьер, (все поля должны проходить проверку (перед отправкой в базу и созданий новой записи данные должны проходить проверку на заполненность, отсутствия символов которые могут навредить системе:<>, &, “”, ‘, пробелы в начале и в конце строк, \ / ? : | заполненность обязательных полей таких как улица, дом либо квартира и подъезд.) в случае нарушения правила должно появляться уведомления с ошибкой о том что система принимает только разрешенные символы (Латинские A-a, 0-9, , . , +) а также ознакомиться с его заказом который берётся из личной корзины пользователя, внести последние изменения в заказ, удалить позиции, увеличить количество позиций заказа, выбрать способ оплаты, наличные или картой. После подтверждения заказа со стороны пользователя, данные заказа, адрес, комментарий, общий вес заказа, блюда и способ оплаты попадают в базу данных заказа, а база данных корзина обнуляется для конкретного пользователя, который совершил заказ.

Также клиент должен иметь возможность составлять свой заказ. Изменять, добавлять и удалять разные позиции заказа, а также изменять адрес доставки и способ оплаты до непосредственного оформления заказа и передачу его в ресторан.

Предусмотрена возможность для пользователя смотреть историю заказов за последнюю неделю. В разделе «мои заказы», размещены все заявки, при нажатии на кнопку «подробнее», в заказах указаны блюда, итоговая стоимость, адрес доставки, адрес ресторана, статус заказа, а также дата заказа.

Администратор должен иметь возможность изменять, удалять, смотреть пользователей, а также смотреть их заказы. Добавлять, изменять, удалять рестораны, и их меню. При удалении блюда, ресторана, пользователя, меняется статус и больше он не отображается в системе, так же администратору нужно указать причину удаления или изменения, которая сохраниться в журнале изменений, для этого необходимо заполнить форму, написать поясняющий комментарий и выбрать причину из предложенного списка. В журнале изменений указан id администратора, который совершил изменения, указана таблица, в которой были проведены имения, тип изменений, и id измененной записи, так же дата изменения и кнопка чтобы перейти непосредственно к измененной записи чтобы отменить изменения в случае ошибки или необходимости.

Для администратора необходимо реализовать функции аутентификации для предоставления прав администратора и редактирования данных. Функции администратора, просмотр базы данных клиентов, ресторанов, изменение данных и удаление.

Просмотр данных о конкретном пользователе, просмотр заказов пользователя. Просмотр базы данных ресторанов реализация функционала добавление, изменения, удаления, просмотр меню конкретного ресторана. При добавлении нового ресторана необходимо заполнить поля формы: названия, адрес, кухня ресторана, логотип ресторана. (перед отправкой в базу и созданий новой записи данные должны проходить проверку на заполненность, отсутствия символов которые могут навредить системе:<>, &, “”, ‘, пробелы в начале и в конце строк, \ / ? : | так же логотип должен проходить проверку на тип файла jpg, так как jpg больше подходит к системе из-за меньшего веса благодаря алгоритмам сжатия и такие картинки будут весить меньше, что обеспечит их более высокую скорость загрузки на странице) Просмотр базы данных блюд реализация функционала добавление, изменения, удаления.

Для курьера должна быть реализована функция регистрации и авторизации на сайте.

Курьер должен иметь возможность выбирать из списка доступных заказов тот который хочет доставить. В доступных заказах должны быть заявки, которые в данный момент доступны для доставки и не заняты другими курьерами. Курьер также имеет возможность отменять заказ с описанием причины, и просматривать подробности выбранного заказа, также есть возможность изменять статус заказа или отменять его с объяснением причины. После того как курьер выбрал заказ он не может выбрать другую заявку пока не завершит уже принятую или не откажется от него с объяснением причины, также для курьера необходимо реализовать возможность авторизоваться в системе и выбирать заказы, не занятые курьерами. Просмотр информации о приятом заказе. Изменение статуса заказа.

Администратор ресторана должен иметь возможность добавлять новые блюда, изменять старые, удалять, в ресторане, который принадлежит им. Так же иметь возможность получать статистику по заказам, в статистке необходимо в форме графиков показывать 10 самых успешных блюд, количество заказов из этого ресторана, доход ресторана, так же возможность посмотреть данные за определенный период, просмотр данных об количестве заказов за последнюю неделю.