МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование и стандарты проектирования»

Тема Программное средство «Ресторан»

Исполнитель

Студента 2 курса группы 6 Белько Никиты Валерьевича

(Ф.И.О.)

Руководитель ассистент Радиванович Дмитрий Александрович

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Пацей Н. В.

(подпись)

Минск 2021

**Содержание**

[**Введение** 3](#_Toc73149274)

[**1. Аналитический обзор литературы** 4](#_Toc73149275)

[1.1. Анализ прототипов 4](#_Toc73149276)

[**2.** **Моделирование предметной области и разработка функциональных требований** 7](#_Toc73149280)

[2.1. Описание инструментов разработки программного средства 7](#_Toc73149281)

[2.2. Описание функций, реализуемых программой 9](#_Toc73149282)

[**3. Проектирование программного средства** 12](#_Toc73149283)

[3.1. Проектирование базы данных 13](#_Toc73149287)

[**4.** **Создание программного средства** 16](#_Toc73149288)

[**5.** **Тестирование программного средства** 18](#_Toc73149289)

[**6 Руководство по использованию** 22](#_Toc73149290)

[**Список использованных источников** 23](#_Toc73149291)

**Введение**

Ресторан – предприятие широкого питания с широким ассортиментом блюд сложного приготовления, включая заказные и фирменные. Блюда, как правило, подаются и съедаются на месте в ресторане. Рестораны сильно различаются по внешнему виду и предложениям, включая большое разнообразие кухонь и видов обслуживания, от недорогих ресторанов быстрого питания до семейных ресторанов со средней ценой и дорогих заведений класса люкс.

Программное средство «Ресторан» сможет сделать процесс заказа блюд более простым и удобным. Ведь, как было сказано ранее, – намного проще заказать в онлайне, нежели ходить по множеству ресторанов в поисках нужных блюд.

Программное средство «Ресторан» реализует работу с блюдами. «Ресторан» имеет несколько режимов доступа: пользователь и администратор. Функциями приложения являются: ведение базы данных заказов и блюд, просматривание информации о блюдах и заказах, добавление их в раздел «корзина», есть возможность добавлять новые блюда (добавлять может только администратор.) и оставлять отзывы о ресторане (добавлять может как пользователь, так и администратор).

Целью курсового проекта является разработка программного обеспечения для удобства заказа блюд и места в ресторане, связанная с ведением базы данных.

Базой данных называется набор логически связанных данных, который предназначен для удовлетворения информационных потребностей организации/предприятия. Для выполнения задания, поставленного в курсовой работе, необходимо освоить работу по проектированию баз данных и работу с базами данных в выбранной среде программирования.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс — Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом — С#, разработанный для создания приложений на платформе Microsoft .NET Framework.

Кроме того, используемая в WPF модель разделения кода и дизайна предполагала работу с декларативным языком описания интерфейса — eXtensible Application Markup Language (XAML).

**1. Аналитический обзор литературы**

* 1. Анализ прототипов

На сегодняшний день на рынке представлено большое разнообразие систем, автоматизирующих процессы принятия и обработки заказов. Которые индивиду­ально настраиваются под потребности определенного клиента. Данные решения являются платными. В свою очередь, это затрудняет анализ и обзор подобных прототипов. Несмотря на это, я постарался в полном объеме проанализировать найденные решения.

Для обзора были выбраны наиболее популярные системы:

1. «Смартомато» – система управления доставкой для ресторанов.

Как указывает производитель, данное программное средство нацелено на рестораны любого типа. Также указано, что у каждого сотрудника ресторана – своя роль в цепочке заказа.

Интерфейс «Смартомато» представлен на рисунке 1.1.

|  |
| --- |
|  |

Рисунок 1.1 - Интерфейс «Смартомато»

К вышеперечисленному, решение предоставляет возможность детального анализа различных показателей бизнеса, сопровождающеюся разного рода диаграм­мами и графиками для более удобного рассмотрения. Также возможна интеграция с уже существующими сервисами по заказу еды.

1. «MobiDel» – программа для автоматизации службы доставки.

Производитель говорит, что данное программное средство позволяет организовать оперативную работу на каждом этапе, начиная от оформления заказа до ведения складского учёта.

Интерфейс «MobiDel» представлен на рисунке 1.2.

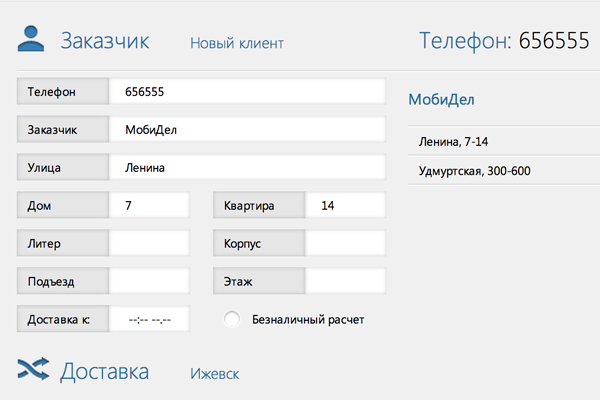


Рисунок 1.2 - Интерфейс «MobiDel»

Помимо выше представленного рисунка, данное решение удивляет обилием интерфейсов для различного типа пользователей. Благодаря которым реализованы возможности оформления заказов, управления предприятием, просмотра различных отчетов и контроля процесса доставки.

1. «FastOperator» – система автоматизации ресторана.

Аналитика, контроль кухни и сборки заказа, управление заказами, контроль курьера и маршрутизация – все это обещает вам производитель при покупке платной подписки. Самое интересное, что бесплатная версия попросту отсутствует.

Интерфейс «FastOperator» представлен на рисунке 1.3.

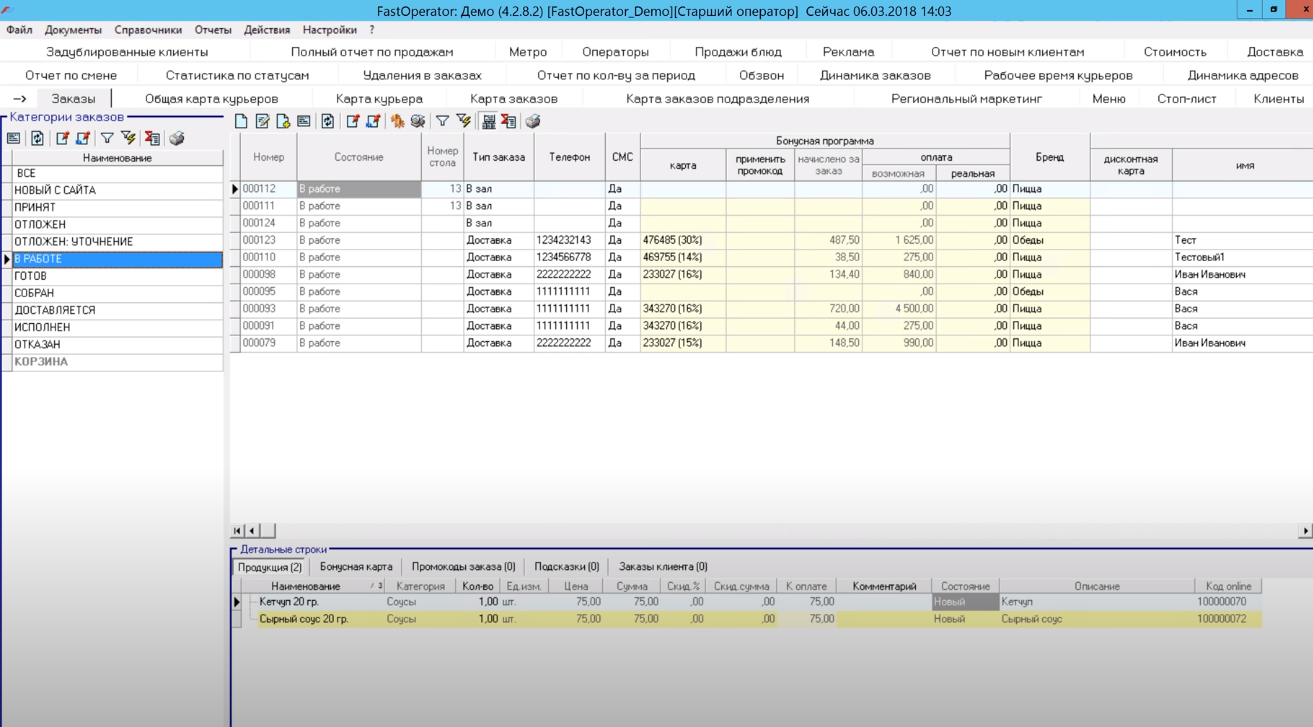


Рисунок 1.3 – Интерфейс «FastOperator»

Производитель дает возможность выбора тарифов клиенту от которых зависит набор предоставляемых услуг. Вплоть до того, что предоставляется возможность закупки специализированного оборудования с установкой на него специального программного обеспечения.

Проанализировав существующие на рынке решения, был сделан следующий вывод:

Несмотря на обилие представленных средств, которые позволяют максимально автоматизировать процессы заказа и доставки еды. Стоит учесть, что данные предложения являются платными и хорошо подходят для больших и популярных сетевых точек общественного питания. Но являются труднодоступными с финансовой точки зрения для небольших заведений, перед которыми также стоит потребность удовлетворения спроса посетителей. Также, несмотря на уверения производителя о быстрой и беспроблемной интеграции в уже существующую бизнес-модель, на практике дело обстоит иначе.



# **Моделирование предметной области и разработка функциональных требований**

## 2.1. Описание инструментов разработки программного средства

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2021;
* программная платформа .NET Framework 5.0;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* ADO.NET C#;
* MySQLConnetion Library;
* MAMP

*C#* — объектно-ориентированный язык программирования. Разработан в 1998 — 2001 под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

*XAML* — это декларативный язык разметки. С точки зрения модели программирования .NET Framework язык XAML упрощает создание пользовательского интерфейса для приложения .NET Framework.

WPF (Windows Presentation Foundation) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Причем важно отметить, что систем управления баз данных может быть множество. В своей сущности они могут различаться. MS SQL Server, например, для создания запросов использует язык T-SQL, а MySQL и Oracle применяют язык PL-SQL. Разные системы баз данных могут иметь разные типы данных. Также могут различаться какие-то другие моменты. Однако функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов: Connection, Command, DataReader, DataSet и DataAdapter. С помощью объекта Connection происходит установка подключения к источнику данных. Объект Command позволяет выполнять операции с данными из БД. Объект DataReader считывает полученные в результате запроса данные. Объект DataSet предназначен для хранения данных из БД и позволяет работать с ними независимо от БД. И объект DataAdapter является посредником между DataSet и источником данных. Главным образом, через эти объекты и будет идти работа с базой данных.

MAMP - это, если коротко, веб-сервер с дистрибутивами Apache и MySQL, предоставляющий разработчикам надежный инструмент тестирования, на который они могут рассчитывать в любое время дня и ночи. Подобно другим приложениям из этой области, софт позволяет без особых усилий устанавливать Apache, PHP и MySQL. Эти три основных компонента, которые лежат в основе веб-сайта, развертываются без необходимости утомительного процесса настройки и настройки каждого из них.  
  
Программа сама позаботится обо всем, что позволит даже начинающим разработчикам быстро и без проблем запустить свой веб-сервер. Обратите внимание, что интерфейс состоит из понятного и удобного в использовании главного окна, в котором отображается состояние серверов Apache и MySQL, а также параметров для доступа к локальному хосту и остановки серверов.  
  
Благодаря интуитивно понятным функциям серверы можно легко запускать и останавливать, изменять используемые порты, выбирать версию и кэш-дистрибутив PHP или управлять веб-сервером. Другими словами, MAMP сам по себе является удобным способом тестирования веб-страниц и веб-приложений локально, без фактического доступа к Интернету. Вы можете управлять базами данных MySQL и выполнять операции SQL через PHPMyAdmin, а дистрибутив PHP позволяет веб-разработчикам создавать свои веб-сайты. В отличие от конкурентов с открытым исходным кодом, MAMP довольно беден с точки зрения диапазона установленных компонентов.

Библиотека MySqlConnection используется для установления соединения с сервером MySQL. Следующие значения по умолчанию установлены для отсутствующих параметров: server = 'localhost:3306', username = имя пользователя владельца процесса сервера и password = пустой пароль.

Параметр server может также включать номер порта, к примеру "hostname:port" или путь к сокету, к примеру ":/path/to/socket" для локального сервера.

Замечание: При указании параметру server значения "localhost" или "localhost:port" клиентская библиотека MySQL будет пытаться соединиться с локальным сокетом. Если вы всё же хотите использовать TCP/IP, используйте адрес "127.0.0.1" вместо "localhost". Если клиентская библиотека пытается подключиться не к тому локальному сокету, это можно исправить через указание директивы mysql.default\_host в конфигурации PHP, после чего можно оставлять параметр server пустым.

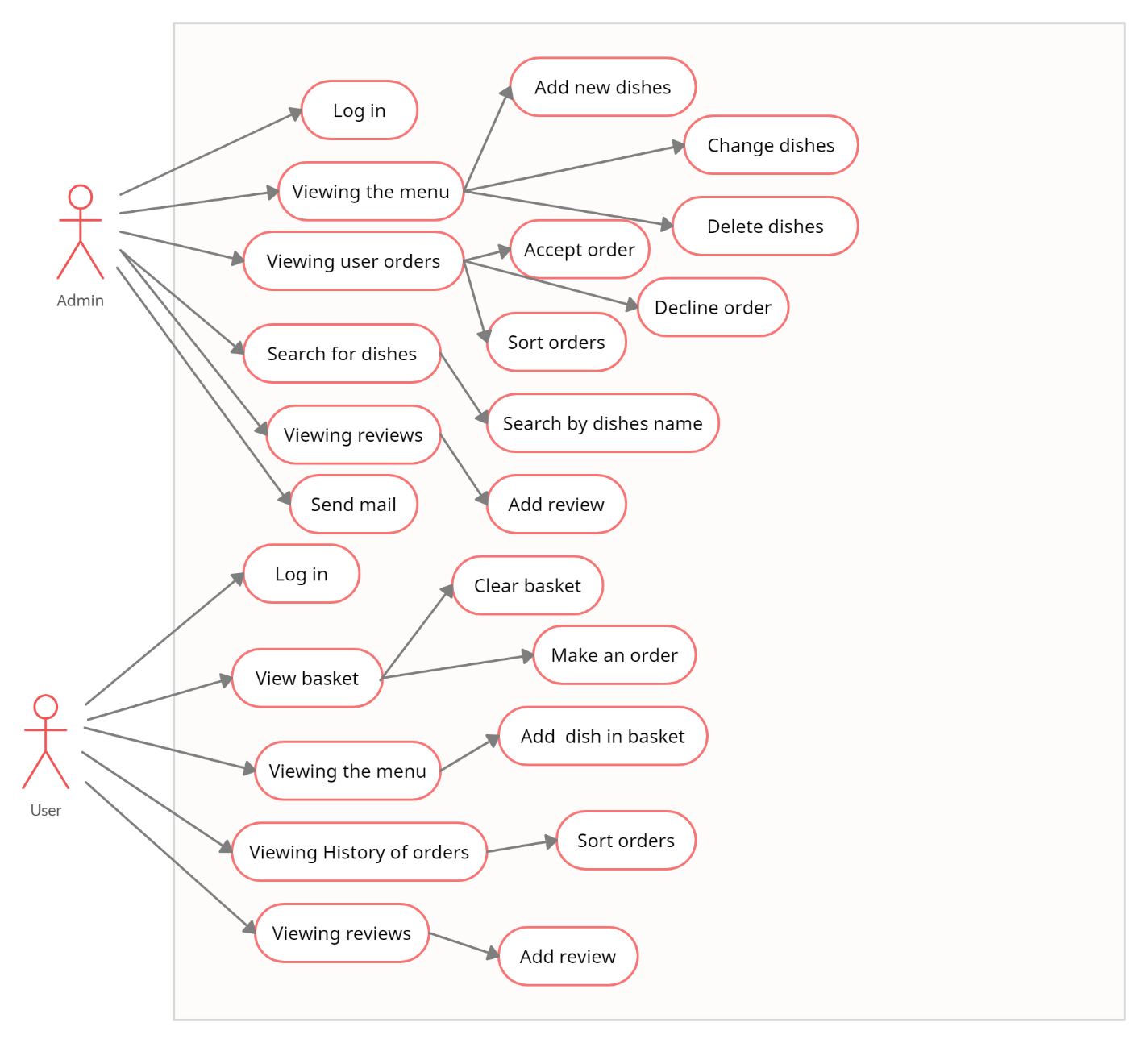


Рисунок 2.1 – Диаграмма вариантов использования (UML Case)

## 2.2. Описание функций, реализуемых программой

Варианты использования программного средства показаны на диаграмме на рисунке 2.1.

Сейчас же хотелось бы рассмотреть более подробно каждый из прецендентов.

Прецедент «View Basket» заключается в просмотре списка выбранных блюд.

Прецедент «Log in» заключается в авторизации (т.е. выбора роли администратора/пользователя) в случае успешного авторизации в учетную запись.

Если мы зашли как «Администратор»:

Прецедент «Viewing the menu» заключается в просмотре блюд, находящихся в системе.

Прецедент «Add New Dishes» заключается в добавлении в систему нового блюда.

Прецедент «Change Dishes» заключается в изменении в системе информации о блюде.

Прецедент «Delete Dishes» заключается в удалении блюда из системы.

Прецедент «Viewing Users Orders» заключается в просмотре заказов всех пользователей.

Прецедент «Search for Dishes» заключается в поиске блюда по названию.

Прецедент «Delete Dishes» заключается в удалении блюда из системы.

Прецедент «Viewing Reviews» заключается в просмотре отзывов в системе.

Прецедент «Add Review» заключается в добавлении отзыва в систему.

Прецедент «Sort Orders» заключается в сортировке заказов по статусу.

Прецедент «Send Email» заключается в отправке зарегистрированным пользователям сообщения на почту.

Если мы зашли как «Пользователь»:

Прецедент «View Basket» заключается в просмотре выбранных блюд.

Прецедент «Viewing History of orders» заключается в просмотре истории заказов пользователя.

Прецедент «Clear Basket» заключается в очистке корзины.

Прецедент «Make an Order» заключается в осуществлении заказа по заданной форме.

Прецедент «Add Dish in Basket» заключается в добавлении блюда в раздел «Basket».

Из всего вышесказанного следует, что программное средство работает следующим образом: изначально запускается главное окно MainWindow, в котором будет окно авторизации/регистрации. После авторизации, в зависимости от роли пользователя, ему станет доступен основной функционал, предоставляемый программным средством. Итак, рассмотрим более подробно каждый из сценариев: администратора и пользователя.

* Администратор

Если пользователь вошёл в систему в качестве администратора, то на главном окне Menu появляются кнопки «Рассылка», «Отзывы» и «Заказы».

В окне добавления новых блюд администратор может добавить новое блюдо, его название, граммовки, стоимость, а также изображение блюда. В этом же окне есть кнопка удаления блюда (крестик), которая позволяет удалить ненужное блюдо

В окне поиска блюд мы можем ввести текст, а программа отобразит в текущем окне все имеющиеся блюда. Искать блюда мы можем только по названию. Помимо вышеперечисленного есть кнопка «Рассылка», которая позволяет отправлять зарегистрированным пользователям сообщение на почту. Кнопка «Заказы» позволяет посмотреть список заказов, которые администратор может как принять, так и отклонить. В зависимости от выбранного варианта пользователю будет отправлено сообщение на почту.

* Пользователь

Зайдя в систему в качестве стандартного пользователя, на базовом окне Menu становятся активными следующие кнопки: «Отзывы», «Заказы» и «Корзина». Работа панели поиска и все последующие действия у пользователя аналогичны действиям с этой же кнопкой у администратора.

Если же пользователь нажмёт кнопки «Корзина», расположенную в окне Menu, то появится панель, содержащая информацию о выбранных блюдах. В этом же окне есть кнопки «Очистить» и «Заказать», с помощью которых пользователь сможет очистить «Корзину» и оформить заказ по заданной форме соответственно.

Также пользователю доступна кнопка «Отзывы», при нажатии на которую открывается окно, содержащее все отзывы о данном ресторане. Можно нажать на кнопку «Оставить отзыв». В этом случае откроется окно, в котором по заданной форме можно оставить отзыв о ресторане. Сам отзыв имеет следующие поля : «Тема», «Текст», «Недостатки» и «Преимущество». После заполнения данных полей следует нажать на кнопку «Оставить отзыв», после чего пользователь получит сообщение об успешной публикации своего отзыва.

* Функционал, доступный как пользователю, так и администратору

Помимо вышеперечисленного, при успешной авторизации пользователь и администратор имеют доступ к главной панели, которая содержит в себе следующие кнопки : «Меню», «Новое», «История», «Работа» и «Рестораны».

При нажатии на «Меню» появляется дополнительная панель, которая содержит в себе кнопки : «Закуски», «Салаты», «Супы» и «Напитки». В зависимости от выбранного варианта вы получить список блюд, удовлетворяющих данному типу. При нажатии кнопок «Новое», «История», «Работа» и «Рестораны» вы получите краткую информацию об новостях, трудовых буднях истории ресторана и его геолокации.

# **3. Проектирование программного средства**



На этом этапе было разработано, спроектированное в предыдущем разделе, программное средство. Которое состоит из 4 модулей: web-приложение и отдельно вынесенная библиотека, содержащая реализацию взаимодействия сервера и базы данных, desktop-приложение и библиотека, в которой реализованы вспомогательные классы взаимодействия с сервером.

Таким образом, в последующих подразделах будет описан процесс разработки. Основное внимание будет уделено особенностям реализации, трудностям, с которыми пришлось столкнуться на этапе разработки, и вариантам их решения, которые были выбраны по тем или иным причинам.

Также в проекте есть 17 классов окон. Классы окон представлены на рисунке 3.1.

В данных классах представлен весь основной функционал программного средства, включающее в себя добавление новых блюд в базу данных, добавление новых отзывов, добавление блюда пользователем в раздел «Корзина», очистка корзины пользователем.

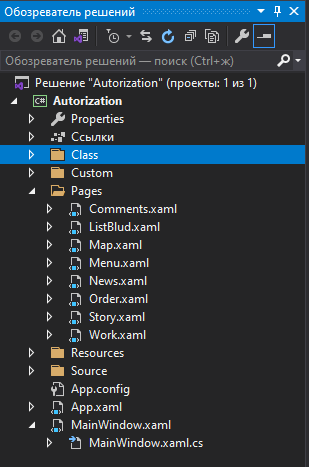


Рисунок. 3.1 – Классы окон

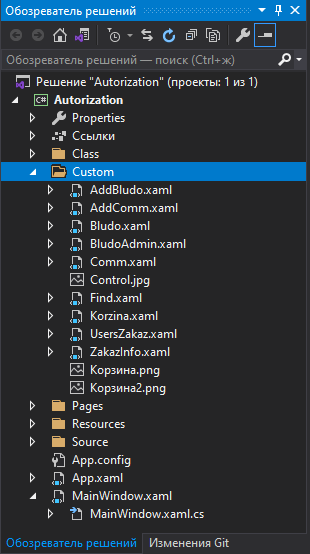


Рисунок. 3.2 – Классы окон

Также в проекте использовался класс SMTP:

SMTP — это широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.

SMTP впервые был описан в RFC 821 (1982 год); последнее обновление в RFC 5321 (2008) включает масштабируемое расширение — ESMTP (англ.  Extended SMTP). В настоящее время под «протоколом SMTP», как правило, подразумевают и его расширения. Протокол SMTP предназначен для передачи исходящей почты с использованием порта TCP 25.

## Проектирование базы данных

Сам термин база данных появился в начале 1960-х годов, и был введён в употребление на симпозиумах, организованных компанией SDC в 1964 и 1965 годах, хотя понимался сначала в довольно узком смысле, в контексте систем искусственного интеллекта. В широкое употребление в современном понимании термин вошёл лишь в 1970-е годы

База данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных

Диаграмма полученной базы данных представлена на рисунке 3.3.

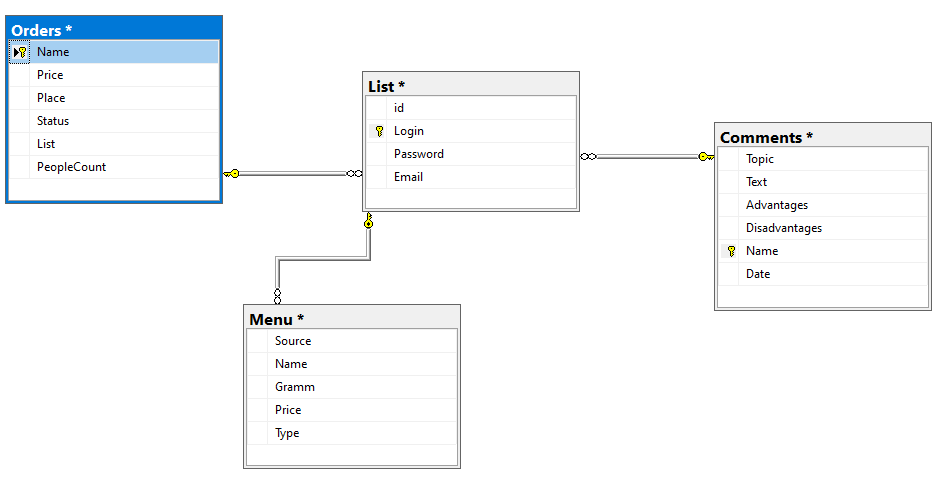


Рис. 3.3 – Диаграмма базы данных

Почему используют реляционную базу данных:

Она поддерживают требования ACID (по крайней мере транзакционная БД) Это единый синтаксис SQL, который используется повсеместно.

Требования ACID:

ACID — это аббревиатура из требований, которые обеспечивают сохранность ваших данных:

* Atomicity — Атомарность
* Consistency — Согласованность
* Isolation — Изолированность
* Durability — Надёжность

Если база данных не поддерживает их, то могут быть печальные последствия из серии «Деньги с одного счета ушли, на другой не пришли»

Всего в базе данных 4 таблиц. В таблице List хранятся все пользователи, зарегистрированные в программе, Menu – информация о блюдах, Orders – информация о заказах пользователей, Comments – текст отзывов, оставленных пользователями

Таблица List состоит из четырех столбцов, иллюстрирующих полную информацию о пользователе:

* Id– хранит идентификатор пользователя;
* Login – хранит имя пользователя;
* Password – хранит пароль пользователя;
* Email – хранит электронную почту пользователя;

Таблица Menu состоит из пяти столбцов, иллюстрирующих полную информацию о блюде:

* Name – хранит название блюда;
* Gramm – хранит информацию о граммовках блюда;
* Price – хранит стоимость блюда;
* Type – хранит тип блюда;
* Source – хранит изображение книги (строку-путь на расположение картинки на диске);

Таблица Orders состоит из шести столбцов, иллюстрирующих полную информацию об заказах пользователей:

* Name - хранит имя пользователя, сделавшего заказ;
* Price – хранит сумму заказа;
* Place – хранит адрес ресторана;
* Status – хранит состояние заказа;
* List – хранит список заказанных блюд;
* PeopleCount – хранит информацию об количество людей;

Таблица Comments состоит из шести столбцов, содержащих информацию об оставленных отзывах:

* Topic – хранит тему отзыва;
* Text – хранит текс самого отзыва;
* Advantages – хранит текст, посвященный достоинствам ресторана;
* Disadvantages – хранит текст, посвященный недостаткам ресторана;
* Name – хранит имя пользователя, оставившего отзыв;
* Date – хранит дату, когда был опубликован отзыв;

# **Создание программного средства**

В данной главе будет рассмотрен базовый функционал классов.

В данном программном средстве все классы можно условно разделить на три группы: классы Pages(Page), Custom(ЭУ) и вспомогательные классы. Детально рассмотрим все классы из данных групп.

Pages классы:

* Comments – предназначенная для отображения и создания отзывов.
* ListBlud – окно, предназначенное для отображения списка блюд у пользователя.
* Map – предназначен для информирования пользователя о местоположении ресторанов и их времени работы.
* Menu – предназначен для отображения меню блюд.
* News–предназначен для информирования пользователя о свежих новостях ресторана.
* Order – предназначен для формирования заказа по заданной форме.
* Story – предназначен информирования пользователя о истории появления ресторана.
* Work – предназначен информирования пользователя о рабочем коллективе в ресторане.

Custom классы:

* AddBludo – предназначен для добавления блюда при авторизации под профилем админа.
* AddComm – предназначен для добавления комментария.
* Bludo – пользовательский элемент управления, предназначенный для отображения блюда в меню.
* BludoAdmin – пользовательский элемент управления, предназначенный для редактирования блюда администратором.
* Comm – предназначен для отображения комментария, оставленного пользователем/администратором.
* Find – пользовательский элемент управления, предназначенный для поиска блюд по названию.
* Korzina – пользовательский элемент управления, предназначенный для отображения блюда в корзине.
* UsersZakaz – пользовательский элемент управления, предназначенный для отображения заказов пользователей у администратора.
* ZakazInfo – пользовательский элемент управления, предназначенный для отображения истории заказов у пользователя.

Вспомогательные классы:

* Data – класс, в котором реализуется подключение к базе данных. Методы данного класса показаны на Рис.4.1
* Korb – класс, содержащий статический список (используется для реализации корзины), статическую переменную sum (она же сумма заказа) и статическая функция SendMessage(), в которой реализована отправка сообщений на почту пользователю. Реализация показана на Рис.4.2
* User – класс, в котором хранится логин и пароль пользователя после успешной авторизации.
* Zakaz– класс, в котором хранится информация о блюде (стоимость, количество блюд и название)

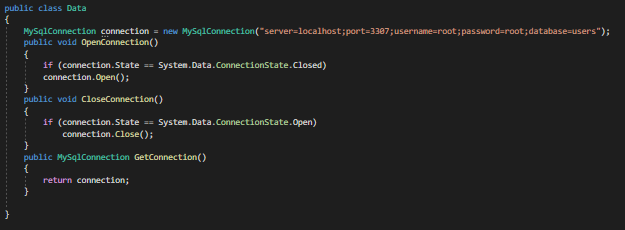


Рис. 4.1 – Реализация класса Data

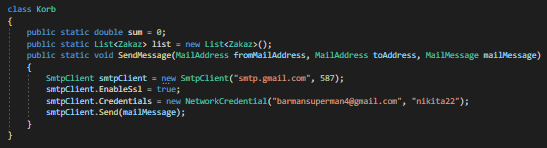


Рис. 4.2 – Реализация класса Korb

Также для удобства все используемые стили были помещены в словарь ресурсов «Style», который используется с самого старта приложения. На Рис.4.3 показано подключение словаря ресурсов.

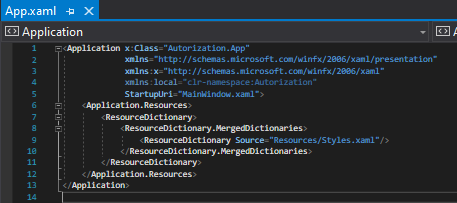


Рис. 4.3 – Подключение словаря ресурсов

# **Тестирование программного средства**

При тестировании данного приложения были проведены следующие тесты на ситуации, которые могли бы привести к ошибке:

* тесты авторизации и регистрации;
* Попытка введения неверного логина;
* Регистрация пустого заказа;
* Попытка регистрация пользователя с логином, который уже занят;
* Попытка введения неверного пароля:

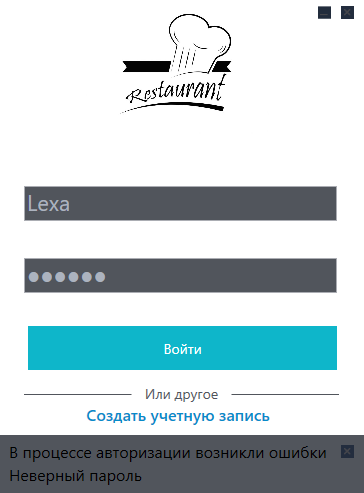
****

Рис. 5.1 – Пользователь ввёл неверный пароль

* Попытка введения неверного логина:

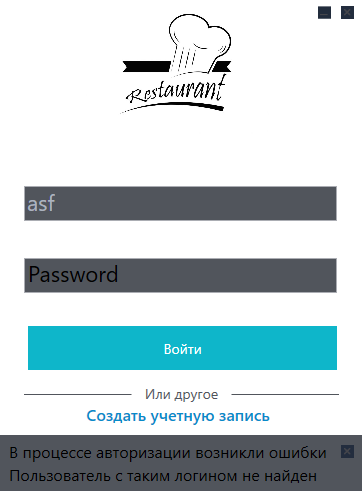


Рис. 5.2 – Пользователь ввёл неверный логин

* Регистрация пустого заказа:

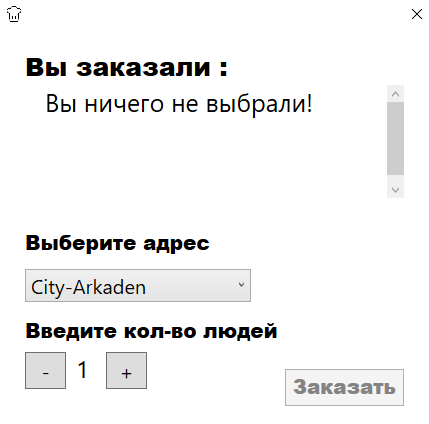
****

Рис. 5.3 – Регистрация пустого заказа

При попытке оформления пустого заказа кнопка «Заказать» становится неработоспособной до тех пор, пока в корзине не появятся блюда.

* Попытка регистрации пользователя с логином, который уже занят:

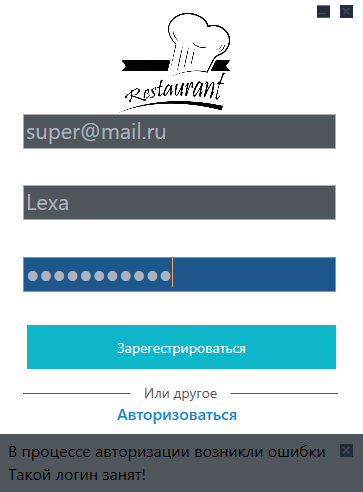
****

Рис. 5.4 – Регистрация c логином, который уже занят

При регистрации нового пользователя каждое поле тщательно проверяется благодаря регулярным выражениям и методам. Пример применяемого регулярного выражения на Рис. 5.5

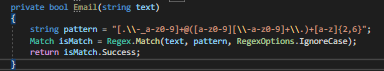


Рис. 5.5 Проверка Email с помощью Regex

# **Руководство по использованию**

При запуске приложения у нас появляется окно авторизации, которое требует ввода логина и пароля для начала работы.

В случае, если вас нету своей учётной записи, её можно создать, нажав на кнопку «Регистрация» в окне авторизации. Вы будете направлены в окно регистрации, где вам нужно будет заполнить все поля. После успешной регистрации, вам нужно будет вернутся в окно авторизации и ввести ваши данные, указанные при регистрации.

Нажав на соответствующие кнопки в окне авторизации, у вас появится окно, где нужно будет ввести данные для входа в учётную запись, через которую производится вход.

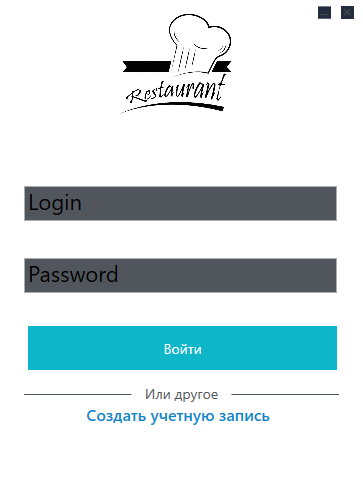


Рис. 6.1 – «Окно авторизации»

После успешного входа, мы попадаем в главное окно, которое изображено на рисунке 6.2.

В этом окне мы может выбрать одну из страниц, которую хотим посетить. Также может добавить блюда в корзину, для этого нужно перейти в «Меню», после чего появятся типы блюд.

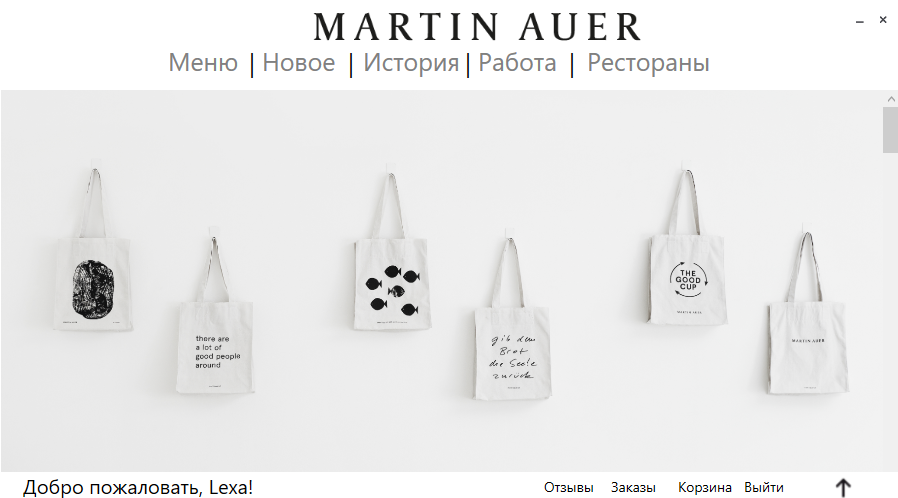


Рис. 6.2 – «Главное окно»

Для добавления блюда нужно выбрать нужно блюда, далее количество и нажать на картинку корзинки. После этого блюдо успешно добавлено. Чтобы проверить список блюд, которые вы выбрали, перейдите на вкладку «Корзина». Рис.6.3



Рис. 6.3 – «Корзина и меню блюд»

После того, как вы определились с вашим заказом, на вкладке «Корзина» нажмите «Заказать», после этого появится окно с формой заказа. Выберите нужные вам параметры и нажмите заказать. Окно заказа показано на Рис 6.4

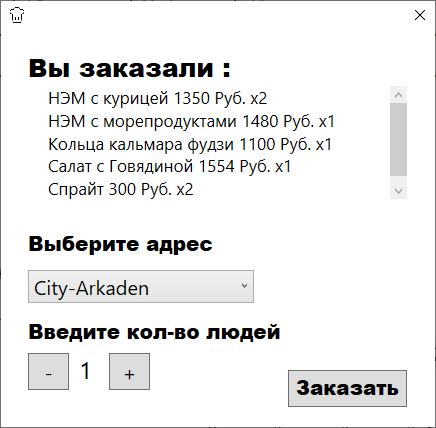


Рис. 6.3 – «Окно заказа»

Также вы можете оставить отзыв о нашем ресторане, для этого перейдите в «Отзыв», после этого нажмите на появившуюся кнопку «Оставить отзыв». Панель отзывов показана на Рис.6.4

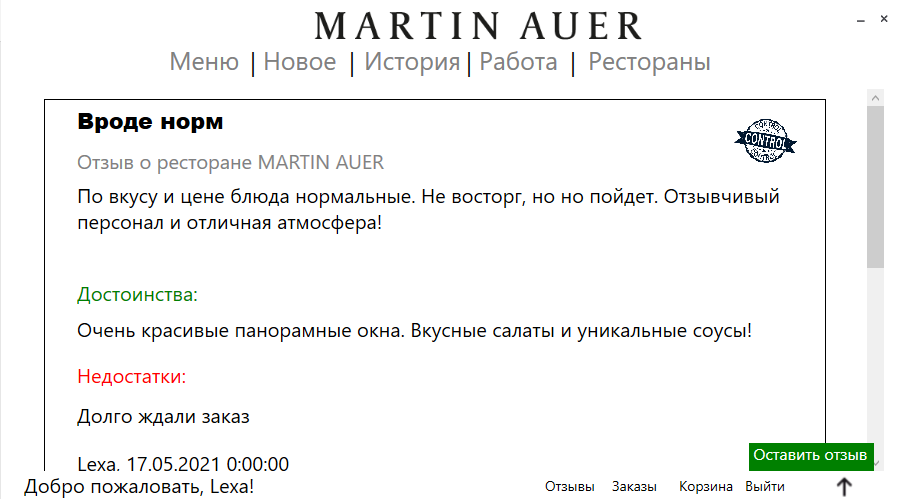


Рис.6.4 – «Страница с отзывами»

**Заключение**

В процессе решения поставленной задачи была достигнута поставленная цель по созданию программного средства «Ресторан». При разработке были выполнены все пункты из указанного списка предполагаемого основного функционала приложения.

В программном средстве были реализованы следующие функции: Администратор:

* + Поддержка работы с базой данных;
  + Регистрация заказов;
  + Оповещение клиентов о новых скидках(почта);
  + Выполнение поисковых запросов, фильтрация;
  + Пополнение/удаление/редактирование блюд в меню;

Пользователь:

* Выполнение регистрации и авторизации;
* Заполнение формы заказа по заданной форме;
* Оставлять отзывы;
* Выполнение поисковых запросов, фильтрацияl
* Просмотр меню;

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

**Список использованных источников**

1. Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.
2. Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с.
3. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/>. Дата доступа: 20.04.2018
4. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com>. Дата доступа: 20.05.2018
5. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> Дата доступа: 13.04.2018
6. Youtube.com C# WPF Material Design UI Menu [Электронный ресурс]/ Режим доступа: <https://www.youtube.com/watch?v=YQ1EJJZBHyE&list=LLJYz_lF4d1mOUKPGhpU53AQ&index=2&t=584s>