**Аннотация**

Программный продукт "Kitchen Boss" представляет собой систему автоматизации бизнес-процессов ресторана, разработанную на платформе .NET Framework 4.7.2 с использованием языка программирования C#, платформы пользовательского интерфейса Windows Forms и Microsoft SQL Server 18. Приложение предназначено для управления персоналом, клиентами, заказами, блюдами и столиками, с разграничением прав доступа для различных должностей. В пояснительной записке приведены описание функциональности, структуры программы, руководство пользователя и методика тестирования.

**Содержание**

[Введение 4](#_Toc198207796)

[1 Назначение и область применения 5](#_Toc198207797)

[2 Постановка задачи 6](#_Toc198207798)

[3 Описание программы 7](#_Toc198207799)

[3.1 Общие сведения 7](#_Toc198207800)

[3.2 Функциональное назначение 7](#_Toc198207801)

[3.3 Описание логической структуры 7](#_Toc198207802)

[3.4 Используемые технические средства 8](#_Toc198207803)

[3.5 Вызов и загрузка 9](#_Toc198207804)

[3.6 Входные и выходные данные 9](#_Toc198207805)

[4 Программа и методика испытаний 9](#_Toc198207806)

[4.1 Объекты испытаний 9](#_Toc198207807)

[4.2 Цель испытаний 9](#_Toc198207808)

[4.3 Требования к программе 9](#_Toc198207809)

[4.4 Методы испытаний 10](#_Toc198207810)

[4.5 Тестовый пример 10](#_Toc198207811)

[5 Руководство оператора 12](#_Toc198207812)

[5.1 Выполнение программы 12](#_Toc198207813)

[5.2 Сообщение оператору 12](#_Toc198207814)

[Заключение 13](#_Toc198207815)

[Приложение А 14](#_Toc198207816)

[Приложение Б 14](#_Toc198207817)

[Источники, использованные при разработке 15](#_Toc198207818)

# Введение

Разработка выполнена в рамках курсового проекта по профессиональному модулю ПМ.01. Программа "KitchenBoss" автоматизирует ключевые процессы управления рестораном: учет заказов, управление персоналом, ведение меню и контроль столиков. В качестве СУБД используется Microsoft SQL Server 18.

# 1 Назначение и область применения

Назначение:

* Автоматизация процессов обслуживания клиентов
* Управление персоналом и контроль прав доступа
* Ведение учета заказов и столиков
* Управление меню и ингредиентами

Область применения:

* Рестораны,
* Кафе
* Другие предприятия общепита

Аналоги:

* "iiko" – комплексная система автоматизации ресторанов.
* "R-Keeper" – решение для управления заведениями общепита.
* "1C: Общепит" – решение от компании «1С-Рарус».
* "1C: Ресторан" – решение от компании «1С-Рарус».

# 2 Постановка задачи

Разработать desktop-приложение со следующим функционалом:

* + 1. Система авторизации с ролевой моделью (менеджер, шеф-повар, официант и др.)
    2. CRUD-операции для сущностей:
       - Сотрудники и должности.
       - Клиенты.
       - Блюда, категории и ингредиенты.
       - Заказы и столики.
    3. Автоматический расчет стоимости заказов.
    4. Контроль статусов заказов.

# 3 Описание программы

## 3.1 Общие сведения

* Технологии: язык программирования C#, платформа пользовательского интерфейса Windows Forms, .NET Framework 4.7.2, Microsoft SQL Server 18.
* Архитектура: клиент-серверное приложение с монолитной архитектурой.
* Сущности БД: 15 таблиц (Employee, Dish, Order и др.)

## 3.2 Функциональное назначение

* Основные модули:

1. Авторизация (fmLogin.cs):
   * + Вход по email/телефону и паролю.
     + Хеширование паролей с солью (SHA512) (см. Приложение А)

Главная форма (fmMain.cs):

* + Навигация между модулями.
  + Контроль прав доступа по должности.

1. Универсальный просмотрщик таблиц (fmTableViewer.cs):
   * Режимы работы для разных сущностей.
   * Динамическая настройка DataGridView.
2. Бизнес-логика:
   * Триггеры базы данных для автоматического расчета сумм заказов.
   * Валидация данных при сохранении.

## 3.3 Описание логической структуры

Алгоритм программы: переделать



Используемые методы:

* HashPassword() – хеширование паролей.
* ConfigureAccessRights() – контроль прав доступа.
* SetupDataGridView() – динамическая настройка таблиц.
* Триггеры базы данных для автоматического пересчета сумм заказов.
* Паттерн “Façade” для более удобной работы с приложением.
* В качестве оптимизации все таблицы реализованы через одну форму.

Схема базы данных:



Структура программы:

1. Слой данных:

* Entity Framework для работы с Microsoft SQL Server.
* Классы-сущности (EmployeeDto, DishDto и др.).

1. Бизнес-логика:

* Валидация данных.
* Расчет стоимости заказов.

1. Интерфейс:

* fmMain – главная форма.
* fmTableViewer – универсальный редактор.
* fmLogin – авторизация.

## 3.4 Используемые технические средства

* Операционная система: Microsoft Windows 10/11 или выше.
* .NET Framework 4.7.2.
* Microsoft SQL Server 2018 или выше.

## 3.5 Вызов и загрузка

Способ запуска:

* Двойной клик по KitchenBoss.exe.
* Запуск через добавленный пользователем ярлык в меню «Пуск».

Технические параметры:

* Объем оперативной памяти: минимально 8 ГБ.
* Дисковое пространство: 30 ГБ.

Точки входа:

* Program.Main() – инициализация приложения.
* fmLogin.loginButton\_Click() – обработка входа.

## 3.6 Входные и выходные данные

Входные данные:

|  |  |  |
| --- | --- | --- |
| Тип данных | Формат | Пример |
| Учётные данные | Текстовый, числовой | Логин: ivanov  Пароль: qwerty123 |
| Данные заказа | Текстовый, числовой | Блюдо: Цезарь  Категория: Салаты  Описание: Салат с курицей, сухариками и соусом  Цена: 300.00 |

# 4 Программа и методика испытаний

## 4.1 Объекты испытаний

* Модуль авторизации.
* Расчет стоимости заказов.

## 4.2 Цель испытаний

Проверка:

* + - Корректности работы алгоритмов расчета.
    - Соответствия интерфейса ТЗ.
    - Устойчивости к ошибочному вводу.
    - Производительности при 100+ одновременных заказах.

## 4.3 Требования к программе

Проверяемые параметры:

* Время отклика ≤ 1 сек.
* Поддержка 25+ активных подключений.
* Корректное отображение на разрешениях 1280x720 или выше
* Защита от SQL-инъекций.

## 4.4 Методы испытаний

1. Модульное тестирование:

* NUnit-тесты для классов.
* Проверка граничных значений.

1. Интеграционное тестирование:

* Сквозные тесты "от заказа до чека".

## 4.5 Тестовый пример

Таблица 1 – Тест – кейс 01 «Авторизация»

|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейса | 01 | | |
| Название: | Авторизация | | |
| Действие | | Ожидаемый результат | Результат теста:  Пройден |
| Предусловие: | | | |
| Запустить сервер с БД, открыть приложение | | Приложение запущено без ошибок | Пройден |
| Шаги теста: | | | |
| Ввод корректных данных | | Открытие главной панели | Пройден |
| Ввод некорректных данных | | Сообщение об ошибке | Пройден |
| Постусловие: | | | |
| Закрыть приложение, выключить БД | | Приложение закрыто, БД выкл | Пройден |

Таблица 2 – Тест – кейс 02 «Создание заказа»

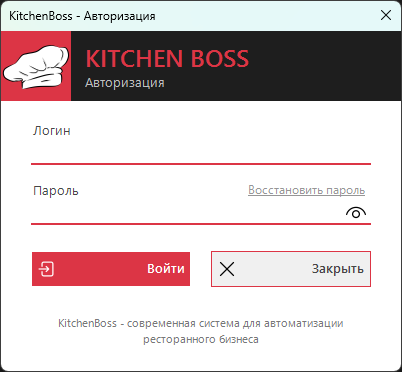
|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейса | 02 | | |
| Название: | Создание заказа | | |
| Действие | | Ожидаемый результат | Результат теста:  Пройден |
| Предусловие: | | | |
| Запустить сервер с БД, открыть приложение | | Приложение запущено без ошибок | Пройден |
| Шаги теста: | | | |
| Ввод корректных данных | | Открытие главной панели | Пройден |
| Открыть раздел “Заказы” | | Загружается список заказов | Пройден |
| Выбрать заказ, у которого не будет статуса “Отменен” или “Закрыт” и нажать “Позиции заказа” | | Загружается список позиций заказа клиента | Пройден |
| Добавить 2 блюда и нажать “Сохранить” | | Корректный расчёт суммы | Пройден |
| Постусловие: | | | |
| Закрыть приложение, выключить БД | | Приложение закрыто, БД выкл | Пройден |

Таблица 3 – Тест – кейс 03 «Создание блюда»

|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейса | 03 | | |
| Название: | Создание блюда | | |
| Действие | | Ожидаемый результат | Результат теста:  Пройден |
| Предусловие: | | | |
| Запустить сервер с БД, открыть приложение | | Приложение запущено без ошибок | Пройден |
| Шаги теста: | | | |
| Открыть раздел “Блюда” | | Загружается список блюд | Пройден |
| Ввести данные в новую строку | | Данные успешно введены | Пройден |
| Сохранить | | Новая запись в таблице | Пройден |
| Постусловие: | | | |
| Закрыть приложение, выключить БД | | Приложение закрыто, БД выкл | Пройден |

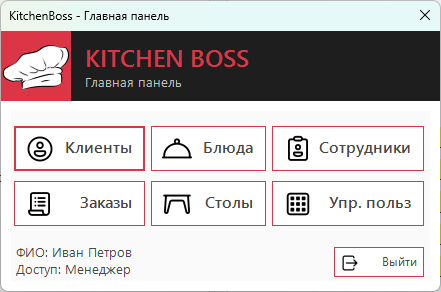
# 5 Руководство оператора

## 5.1 **Выполнение** программы



1. Запуск приложения через exe файл / ярлык (сервер баз данных должен работать).

2. Ввод учётных данных.

3. Главная Панель.

В зависимости от уровня доступа некоторые кнопки/функции могут быть ограничены или недоступны.

Завершение программы – закрыть окно Главной Панели, а затем закрыть окно Авторизации.

## 5.2 Сообщение оператору

Ошибки, которые могут возникнуть:

* "Ошибка подключения к БД" → проверить сервер.
* Какие-то функции недоступны → проверьте под какой учётной записью вы вошли.

# Заключение

В ходе разработки приложения "Kitchen Boss" для автоматизации ресторанного бизнеса был реализован комплексный программный продукт, решающий ключевые задачи управления заведениями общественного питания. Система демонстрирует следующие ключевые характеристики:

1. Гибкость архитектуры:

* Модульная структура позволяет легко расширять функционал.
* Разделение на слой данных, бизнес логику и пользовательский интерфейс обеспечивает поддержку кода.
* Использование Entity Framework упрощает поддержку миграцию на другие СУБД.

1. Безопасность и надежность:

* Многоуровневая система авторизации с хешированием паролей.
* Встроенные механизмы валидации входных данных.
* Транзакционная обработка критических операций.

1. Производительность:

* Оптимизированные SQL-запросы к базе данных.
* Кэширование часто используемых данных.
* Асинхронная загрузка интерфейсов.

1. Удобство использования:

* Единый интерфейс для всех бизнес-процессов.
* Контекстные подсказки и валидация ввода.
* Адаптивный дизайн под разные разрешения экранов.

Перспективы развития системы:

* Внедрение мобильного приложения.
* Интеграция с онлайн-платежными системами.
* Разработка модуля аналитики и прогнозирования.
* Поддержка мультиязычного интерфейса.

Рекомендации по внедрению:

* Провести обучение персонала по работе с системой.
* Разработать регламент резервного копирования данных.
* Обеспечить регулярное обновление ПО.

# Приложение А

**Функция хеширования пароля**

private string HashPassword(string password, string salt)

{

string passwordWithSalt = salt + password;

using (var sha512 = SHA512.Create())

{

byte[] bytes = Encoding.UTF8.GetBytes(passwordWithSalt);

byte[] hashBytes = sha512.ComputeHash(bytes);

return Convert.ToBase64String(hashBytes);

}

}

# Приложение Б

**Enum, использующийся в программе для ограничения доступа по уровню доступа**

public enum Positions

{

[Display(Name = "Менеджер")]

Manager,

[Display(Name = "Шеф-повар")]

ChefCook,

[Display(Name = "Официант")]

Waitor,

[Display(Name = "Повар")]

Cook,

[Display(Name = "Повар-стажер")]

CookTrainee

}

public static class PositionExtensions

{

public static string GetPositionDisplayName(this Enum enumValue)

{

var fieldInfo = enumValue.GetType().GetField(enumValue.ToString());

var displayAttribute = fieldInfo?.GetCustomAttributes(typeof(DisplayAttribute), false)

.OfType<DisplayAttribute>()

.FirstOrDefault();

return displayAttribute?.Name ?? enumValue.ToString();

}

}

# Источники, использованные при разработке

1. ГОСТ Р ИСО\_МЭК 25051-2017 Требования к качеству готового к использованию программного продукта (RUSP) и инструкции по тестированию.
2. ЕСПД Единая система программной документации.
3. Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов, Дэвид Скляр, 2017
4. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, Никсон Робин, 2019
5. Изучаем JavaScript. Руководство по созданию современных веб-сайтов, Этан Браун, 2017

**Internet – ресурсы**

1. PHP: Hypertext preprocessor [Электронный ресурс] – [http://php.net](http://php.net/)
2. JavaScript Tutorial [Электронный ресурс] – <http://w3schools.com/js>
3. MySQL [Электронный ресурс] - <https://www.mysql.com/>
4. htmlbook.ru | Для тех кто делает сайты [Электронный ресурс] -http://htmlbook.ru/