# ВВЕДЕНИЕ

В условиях быстро меняющихся запросов клиентов и потребности в оптимизации процессов управления рестораном, автоматизация учёта и изменений меню становится важной задачей.

Основной целью данной курсовой работы является разработка программного модуля для управления ресторанным меню. Разработанная система должна позволить управлять ассортиментом блюд, их категориями, весом и ценами. Важным аспектом является возможность автоматического обновления меню, создание отчётности и интеграция с базой данных для сохранения информации о блюдах и их характеристиках. Кроме того, система должна учитывать потребности различных ролей пользователей, включая гостей и менеджеров, предлагая функциональность, подходящую для каждой роли.

Важно, чтобы программа оптимизировала рабочие процессы, сокращая время на внесение изменений в меню, обеспечивая актуальность данных и возможность быстро получать отчёты о состоянии меню и заказов.

Объектом исследования является работа ресторана как комплексный процесс.

Предметом исследования является процесс автоматизации управления ресторанным меню, включая взаимодействие с базой данных, интерфейс для пользователей и механизмы обработки данных о блюдах, относящихся к разным категориям.

Для решения данных задач были использованы такие инструменты, как язык программирования Python, база данных SQLite, библиотека PyQt6 для создания графического интерфейса и openpyxl для генерации чека и отчета по продажам в формате Excel. Система проектировалась с учётом масштабируемости и возможности её дальнейшего расширения.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Информационное обеспечение задачи

Для эффективной реализации системы управления ресторанным меню требуется четко определить источники, структуру и способы обработки данных, связанных с меню ресторана, а также функционал, обеспечивающий автоматизацию управления меню.

Основным источником данных является база данных SQLite, которая хранит следующую информацию:

* + Блюда: название, вес, цена, изображение блюда, категория.
  + Категории блюд: название категории.
  + Ингредиенты: название ингредиента, единица измерения.
  + Меню: название типа меню.
  + Пользователи: имя пользователя (логин), пароль, статус.
  + Заказы: id пользователя (ссылаемся на текущего гостя), дата заказа, полная стоимость.
  + Позиции заказа: id заказа (ссылаемся на заказ), id блюда (ссылаемся на блюдо), количество, цена позиции.

Эти данные вводятся и обновляются пользователями системы, что позволяет поддерживать актуальность информации о меню.

Основными пользователями системы являются:

* + Менеджер ресторана, который управляет меню, обновляет данные о блюдах и категориях, а также формирует отчеты.
  + Гости ресторана, которые взаимодействуют с интерфейсом для просмотра меню и оформления заказов.

Система автоматизирует рутинные операции, связанные с управлением меню, минимизируя вероятность ошибок и сокращая время на обработку данных.

Программный модуль для системы управления ресторанным меню должен предоставлять следующий функционал:

Управление блюдами: добавление новых блюд с указанием их названия, категории, веса, цены и изображения, удаление блюд, которые больше не входят в меню, и редактирование данных о блюдах, включая изменения цены, веса.

Управление категориями блюд: создание категорий для структурирования меню, таких как "Супы", "Салаты", "Напитки", с возможностью их удаления и редактирования.

Создание и подтверждение заказа гостем с возможностью отменить заказ или заплатить за выбранные позиции.

Формирование меню: создание меню на основе категорий и блюд, а также отображение текущего меню для гостей и сотрудников ресторана.

Формирование отчетности: генерация чека для гостей и создание отчета по продажам в окне менеджера.

## Обзор и анализ существующих программных решений

Существующие на рынке системы автоматизации ресторанного бизнеса предлагают широкий спектр возможностей: от управления меню до генерации аналитических отчетов. Однако многие из них ориентированы на крупные заведения с высоким бюджетом, что делает их избыточными для небольших ресторанов.

Рассмотрим некоторые из них:

* R-Keeper предназначен для автоматизации ресторанного бизнеса и включает функции управления меню, обработки заказов, учета складских остатков и генерации отчетов о продажах.[[1]](#footnote-1)
* iiko автоматизирует ресторанные процессы, включая управление меню, учет складских остатков, анализ продаж и поддержку онлайн-заказов.[[2]](#footnote-2)
* Poster POS фокусируется на простоте использования для небольших заведений. Функционал включает управление меню, учет заказов и генерацию отчетов.[[3]](#footnote-3)

Эффективное управление ресторанным меню играет ключевую роль в успешной работе заведения, независимо от его масштаба. Анализ существующих решений, таких как R-Keeper, Iiko и Poster POS, показывает, что автоматизация процессов позволяет значительно оптимизировать работу ресторана.

## Постановка задачи

Целью данной работы является разработка программного модуля для управления ресторанным меню, который обеспечит автоматизацию процессов добавления, редактирования, удаления блюд, формирования чеков, а также генерацию отчетов о продажах. Модуль должен быть интуитивно понятным для пользователей, устойчивым к ошибкам и обеспечивать актуальность данных.

Задачи программного модуля “Система управления ресторанным меню“ для ресторана “Семейный уголок”:

* Разработка архитектуры:

Архитектура должна быть гибкой и масштабируемой для удобства дальнейших улучшений. По этой причине были выбраны язык программирования Python, фреймворк PyQT6 для создания графического интерфейса и база данных SQLite для хранения информации.

* Создание ролевой структуры:

Определение ролей пользователей (менеджер и гость) с соответствующими правами доступа к функциям системы.

* Просмотр текущего меню:

Возможность просмотра актуального меню.

* Добавление новых блюд:

Функция добавления новых блюд с указанием названия, цены, веса, категории.

* Редактирование информации о существующих блюдах:

Возможность изменения информации о блюде, включая название, цену, вес и категорию.

* Удаление блюд из меню:

Удаление блюд, например, в случае отсутствия ингредиентов.

* Автоматическое создание отчета по продажам за определенный период времени:

Менеджер создает отчет по продажам со всеми заказами за определенный диапазон времени с учетом цен каждой позиции, даты совершения заказа, именем пользователя и общей стоимостью заказов.

* Выбор блюд гостем для заказа:

Гость выбирает блюда из меню и добавляет их в заказ с отображением стоимости как отдельных блюд, так и общей стоимости заказа.

* Подтверждение заказа:

Окно подтверждения с итоговой суммой и списком выбранных блюд.

* Генерация чеков для гостей с указанием выбранных блюд и общей стоимости:

Автоматическое создание чеков с указанием выбранных блюд и общей стоимости заказа.

* Хранение и обновление информации о блюдах, категориях и ингредиентах в базе данных после изменений в меню:

Обновление информации о блюдах, категориях и ингредиентах в базе данных после редактирования.

* Разработка интерфейса:

Удобный и интуитивно понятный интерфейс для пользователей с гармоничным дизайном.

* Тестирование:

Проверка всех функций системы на корректность и удобство использования.

Структура входной и выходной информации:

* Входная информация:
* Логин и пароль для авторизации пользователей с разными правами доступа (менеджер и гость);
* Данные о блюдах, заказах, категории.
* Выходная информация:
* Результаты выполненных операций: редактирование, удаление и добавление блюд в меню;

Вывод чека для гостя с информацией о заказе: наименования блюд, цены и общая стоимость заказа.

Вывод отчета по продажам за определенный диапазон времени.

# 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ

## 2.1 Эскизное проектирование

На этапе эскизного проектирования были определены основные функциональные требования к разрабатываемому модулю.

Для роли "Менеджер":

* Вход в систему:

Пользователь выбирает роль "Менеджер".

Вводит логин и пароль.

При успешной авторизации менеджер попадает в главное меню, где доступны функции редактирования меню, просмотра текущего меню и просмотра отчетов.

Изменение меню:

* Добавление блюда:

Менеджер нажимает кнопку "Добавить блюдо".

Вводит название, цену, вес, категорию, описание.

Нажимает "Сохранить".

Система обновляет базу данных.

* Редактирование блюда:

Менеджер нажимает кнопку "Редактировать блюдо".

Выбирает блюдо из таблицы меню и вносит его название в строку.

Вносит изменения (например, изменяет цену или вес – возможно изменять несколько критериев за один раз).

Нажимает "Сохранить".

Система обновляет базу данных.

* Удаление блюда:

Менеджер нажимает кнопку "Удалить блюдо".

Система запрашивает подтверждение.

При подтверждении блюдо удаляется из базы данных.

* Просмотр текущего меню:

Менеджер нажимает кнопку "Показать текущее меню".

Отображается таблица со всеми блюдами.

* Просмотр отчетов:

Нажимает кнопку "Журнал заказов".

В следующем окне нажимает кнопку "Отчет по продажам".

Система генерирует файл Excel с данными о заказах (названия блюд, цены, дата оформления заказа, общая сумма) за определенный период.

Отчет сохраняется, и менеджер получает уведомление о завершении операции.

Для роли "Гость":

* Вход в систему:

Пользователь выбирает роль "Гость".

Вводит только логин (пароль не требуется).

После успешной авторизации попадает в окно просмотра меню.

* Просмотр меню:

Отображается список блюд.

Пользователь может видеть название, вес, цену и категорию блюда.

* Выбор блюд для заказа:

Гость выбирает блюда, ставя флажки рядом с ними.

Нажимает кнопку "Заказать".

Система рассчитывает общую сумму заказа и отображает окно подтверждения.

* Подтверждение заказа:

В окне подтверждения отображается список выбранных блюд и их стоимость.

Гость нажимает "Да", чтобы подтвердить и завершить заказ, или "Отмена", чтобы вернуться к выбору блюд.

* Генерация чека:

После подтверждения заказа система создает файл Excel с названием чека (order\_receipt.xlsx).

Чек содержит названия выбранных блюд, их цены и общую стоимость.

Гость получает уведомление об успешной оплате и завершении заказа.

Связь окон и действий:

Менеджер: Каждое действие обновляет данные в базе и интерфейсе. Итоговые действия приводят к изменению меню или генерации отчетов.

Гость: Действия связаны с формированием заказа. Чек создается только после завершения заказа.

В блок-схеме описан принцип работы программы с точки зрения двух ролей – менеджера и гостя: (см. Рисунок 1)

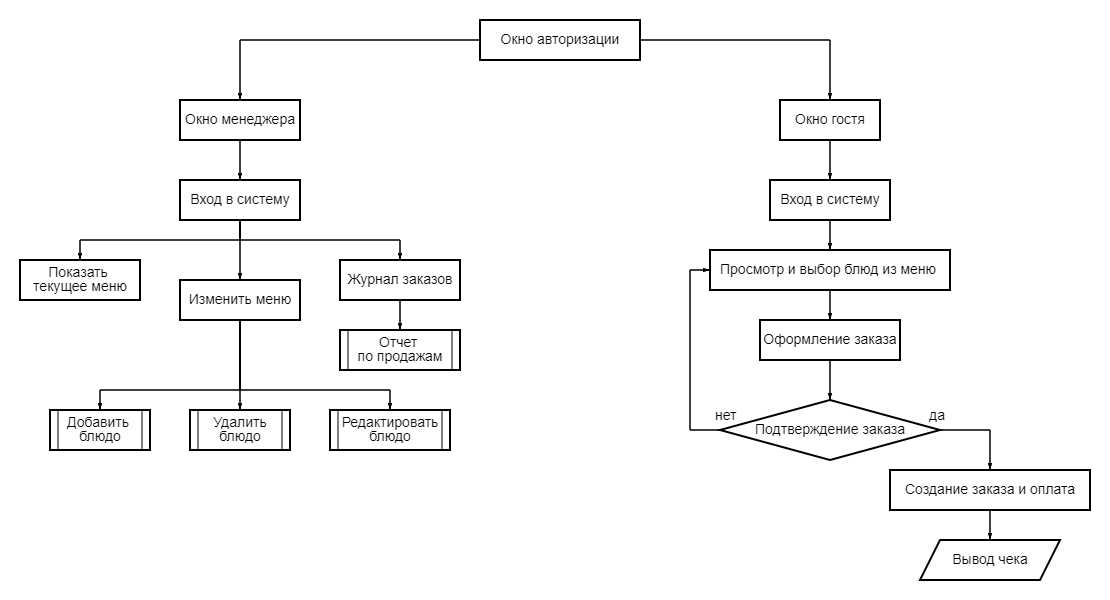


Рисунок 1 – Блок-схема работы программы

В системе роли представлены двумя категориями: менеджер и гость. Диаграмма наглядно показывает, какие функции доступны для каждой роли: (см. Рисунок 2)

* Менеджер

Просмотр текущего меню в виде таблицы.

Добавление, удаление и редактирование блюд.

Генерация отчета о продажах в Excel.

* Гость

Просмотр меню с блюдами и ценами.

Выбор блюд через установку флажков.

Оформление заказа через окно подтверждения.

Получение чека.

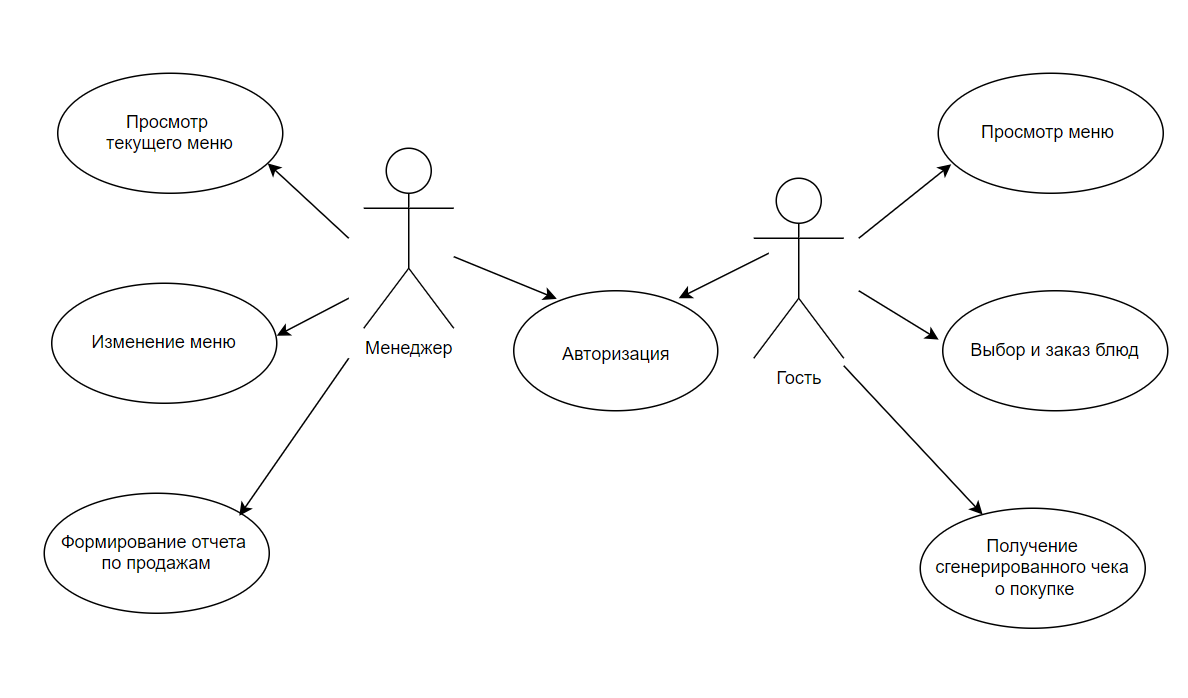
****

Рисунок 2 – Диаграмма прецедентов

### 2.1.1 Выбор архитектуры

Для реализации программы была выбрана одноуровневая монолитная архитектура с локальным хранилищем данных. Логика приложения разделена на следующие уровни:

Уровень данных (база данных): используется SQLite, что обеспечивает простоту интеграции, автономность и отсутствие необходимости настраивать серверную СУБД.

Логический уровень: реализован с помощью языка Python, в котором размещены функции для взаимодействия с базой данных и обработки данных.

Пользовательский интерфейс: создается с помощью PyQt6, что позволяет реализовать удобный и интуитивно понятный графический интерфейс.

### 2.1.2 Выбор инструментальных средств реализации

Python: основной язык программирования, предоставляющий библиотеки для работы с базами данных и создания интерфейсов.

SQLite: база данных, используемая для хранения информации о блюдах, категориях, заказах и пользователях.

PyQt6: библиотека для разработки графического интерфейса пользователя.

openpyxl: библиотека для генерации отчетов в формате Excel.

## 2.2 Проектирование базы данных

Программа взаимодействует с базой данных SQLite, которая содержит 9 таблиц. Ниже представлено краткое описание каждой таблицы БД “MENU”: (см. Рисунок 3)

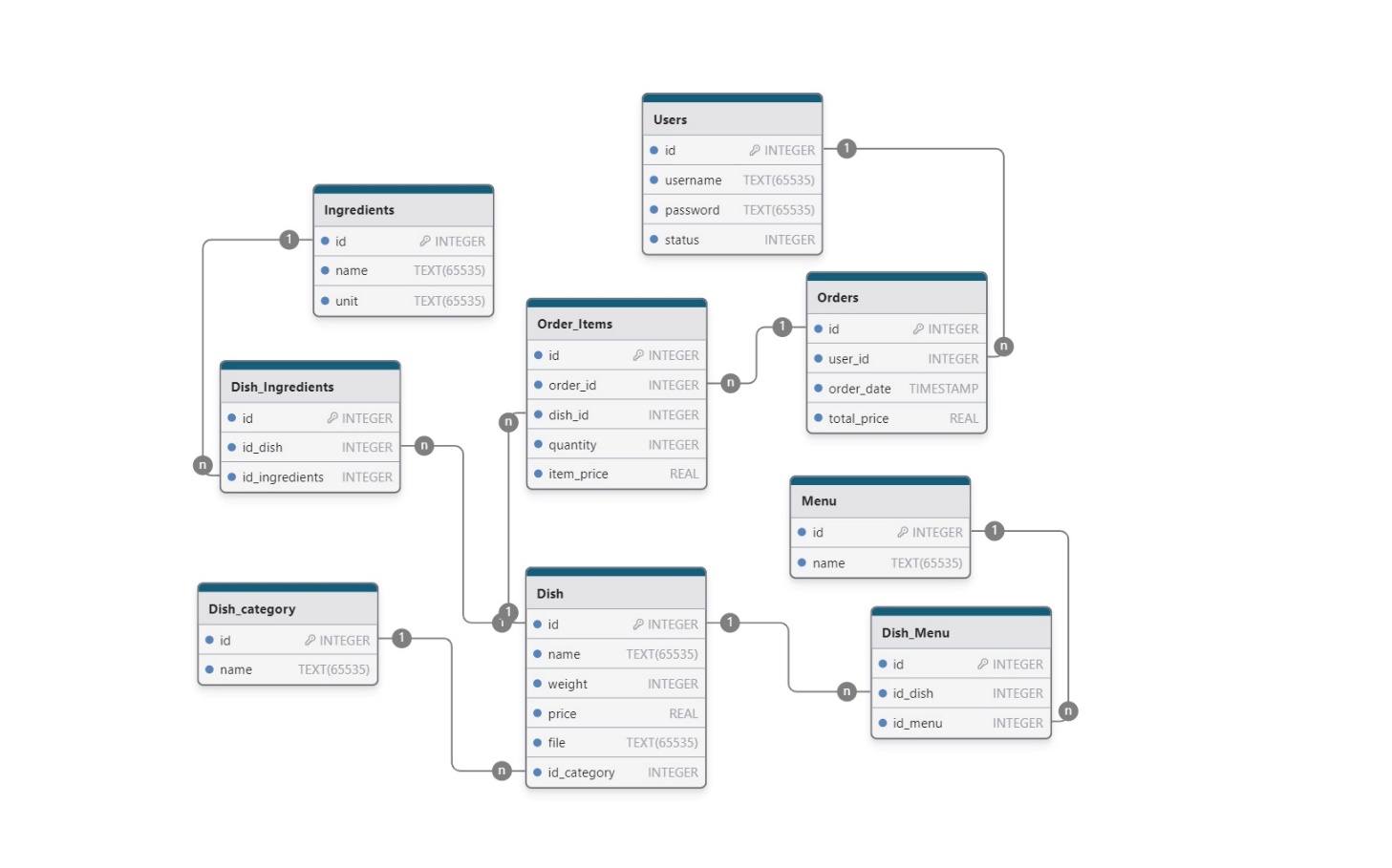


Рисунок 3 – ER-модель БД “MENU”

Таблица Users (Пользователи):

id (INTEGER) — уникальный идентификатор пользователя.

username (TEXT) — имя пользователя.

password (TEXT) — пароль пользователя.

status (INTEGER) — статус пользователя (0 — гость, 1 — менеджер).

Таблица Dish\_category (Категории блюд):

id (INTEGER) — уникальный идентификатор категории.

name (TEXT) — название категории.

Таблица Dish (Блюда):

id (INTEGER) — уникальный идентификатор блюда.

name (TEXT) — название блюда.

weight (INTEGER) — вес блюда в граммах.

price (REAL) — цена блюда.

file (TEXT) — путь к изображению блюда (опционально).

id\_category (INTEGER) — идентификатор категории блюда, ссылающийся на таблицу Dish\_category.

Таблица Ingredients (Ингредиенты):

id (INTEGER) — уникальный идентификатор ингредиента.

name (TEXT) — название ингредиента.

unit (TEXT) — единица измерения ингредиента.

Таблица Dish\_Ingredients (Связь блюд и ингредиентов):

id (INTEGER) — уникальный идентификатор записи.

id\_dish (INTEGER) — идентификатор блюда, ссылающийся на таблицу Dish.

id\_ingredient (INTEGER) — идентификатор ингредиента, ссылающийся на таблицу Ingredients.

Таблица Menu (Меню):

id (INTEGER) — уникальный идентификатор меню.

name (TEXT) — название меню (например, Завтрак, Обед, Ужин).

Таблица Dish\_Menu (Связь блюд и меню):

id (INTEGER) — уникальный идентификатор записи.

id\_dish (INTEGER) — идентификатор блюда, ссылающийся на таблицу Dish.

id\_menu (INTEGER) — идентификатор меню, ссылающийся на таблицу Menu.

Таблица Orders (Заказы):

id (INTEGER) — уникальный идентификатор заказа.

user\_id (INTEGER) — идентификатор пользователя, оформившего заказ, ссылающийся на таблицу Users.

order\_date (TIMESTAMP) — дата и время оформления заказа.

total\_price (REAL) — общая стоимость заказа.

Таблица Order\_Items (Позиции заказа):

id (INTEGER) — уникальный идентификатор записи.

order\_id (INTEGER) — идентификатор заказа, ссылающийся на таблицу Orders.

dish\_id (INTEGER) — идентификатор блюда, ссылающийся на таблицу Dish.

quantity (INTEGER) — количество блюд.

item\_price (REAL) — цена одной позиции блюда.

## 2.3 Разработка интерфейса

Интерфейс разработан для двух типов пользователей — гостей и менеджера. Он обеспечивает понятную навигацию между основными функциями программы, такими как авторизация, просмотр меню, оформление заказа, управление меню и создание отчетов.

Главное окно программы — окно входа (LoginWindow), позволяет пользователю выбрать свою роль (гость или менеджер) через выпадающий список и авторизоваться с вводом имени пользователя и, при необходимости, пароля. (см. Рисунок 4)

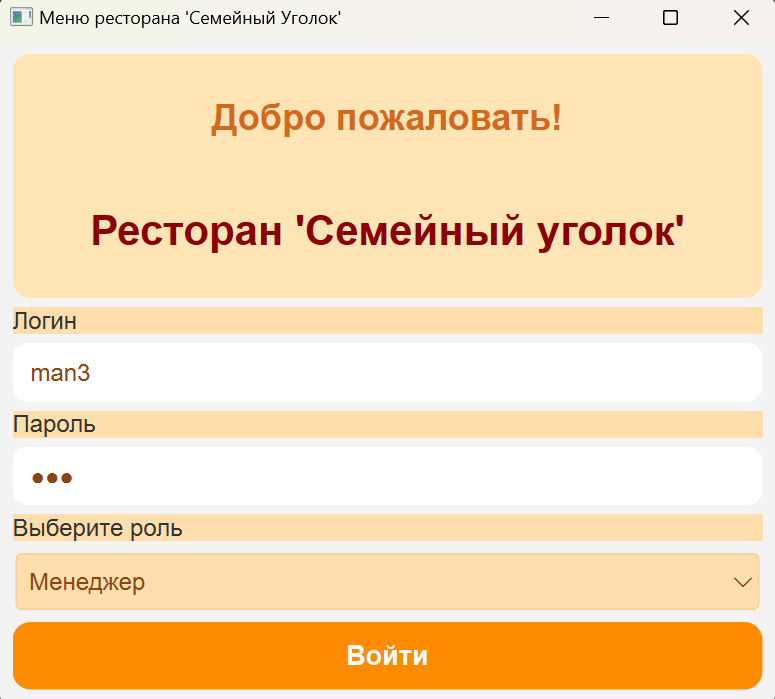


Рисунок 4 – Окно авторизации ресторана “Семейный уголок”

После успешной авторизации гостя перенаправляют в окно для оформления заказа (GuestWindow), а менеджера — в окно управления меню (ManagerWindow).

В окне гостя (GuestWindow) реализован функционал просмотра текущего меню ресторана и оформления заказа. Таблица, расположенная в центре окна, содержит такие колонки, как название блюда, категория, вес, цена, изображение блюда и флажок для выбора блюда. (см. Рисунок 5)



Рисунок 5 – Окно гостя для выбора блюд

После нажатия на кнопку "Заказать" открывается окно подтверждения заказа (OrderConfirmationWindow), где отображается список выбранных блюд, их стоимость и общая сумма. Окно подтверждения заказа представляет собой диалоговое окно с двумя кнопками: "Да" и "Нет". (см. Рисунок 6)

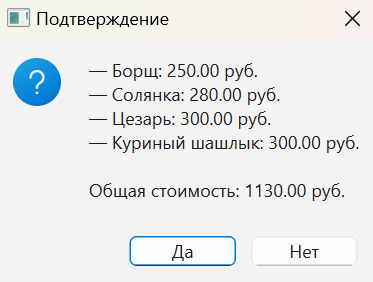


Рисунок 6 – Окно подтверждения заказа

При подтверждении заказа данные сохраняются в базе данных, а чек автоматически формируется в формате Excel. (см. Рисунки 7.1 и 7.2)

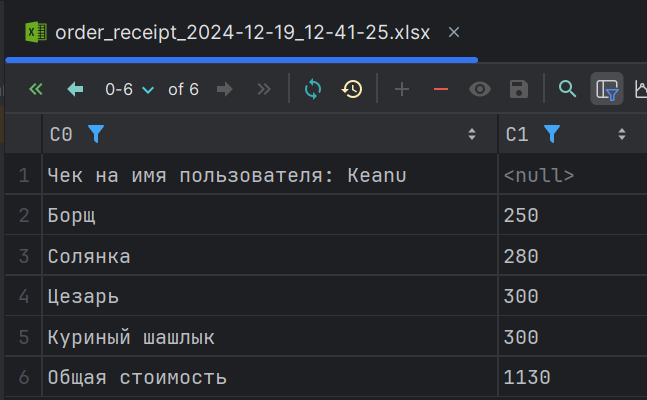


Рисунок 7.1 – Вывод чека в PyCharm

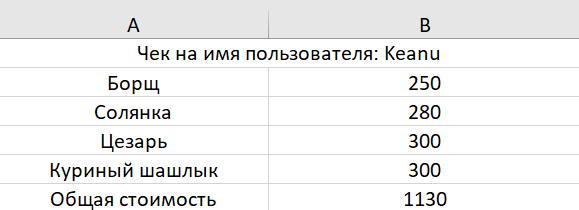


Рисунок 7.2 – Вывод чека в Excel

В окне менеджера (ManagerWindow) также прописан функционал для просмотра текущего меню ресторана, но без изображений. Именно здесь сосредоточены функции управления меню ("Изменить меню") и работы с отчетами ("Журнал заказов"). (см. Рисунок 8)



Рисунок 8 – Просмотр текущего меню в окне менеджера

Кнопки "Добавить блюдо", "Редактировать блюдо" и "Удалить блюдо" вызывают соответствующие диалоговые окна, где можно внести изменения в меню. Для добавления блюда пользователю необходимо указать его название, цену, вес, категорию и путь к изображению. Добавление новых категорий реализовано через выпадающий список, который поддерживает ввод новых значений. (см. Рисунки 9, 10, 11, 12)

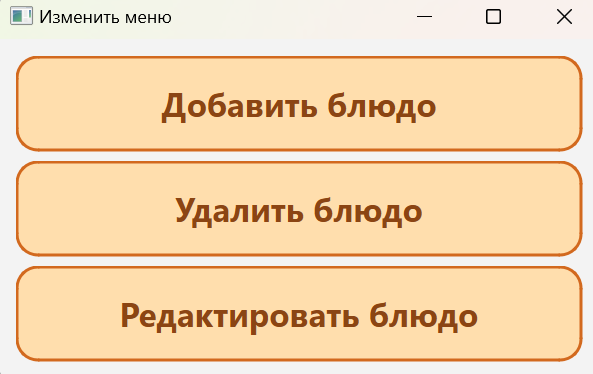


Рисунок 9 – Окно для изменения меню

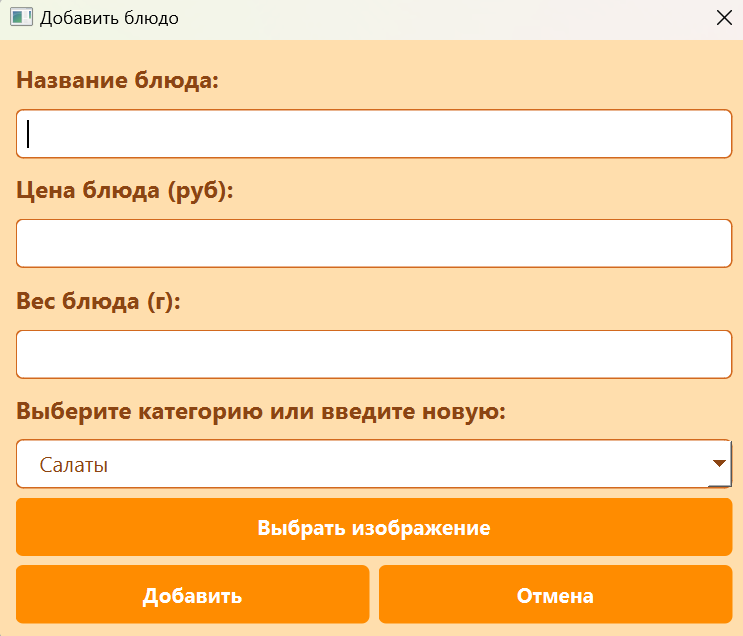


Рисунок 10 – Окно для добавления блюда в меню

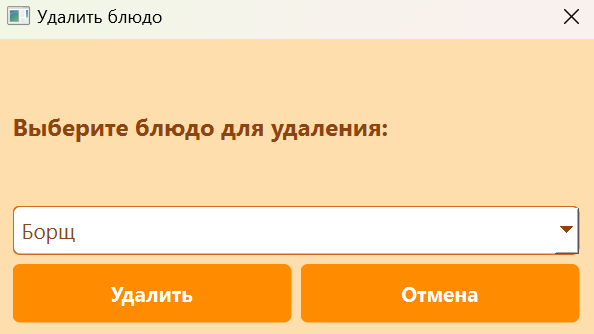


Рисунок 11 – Окно для удаления блюда из меню

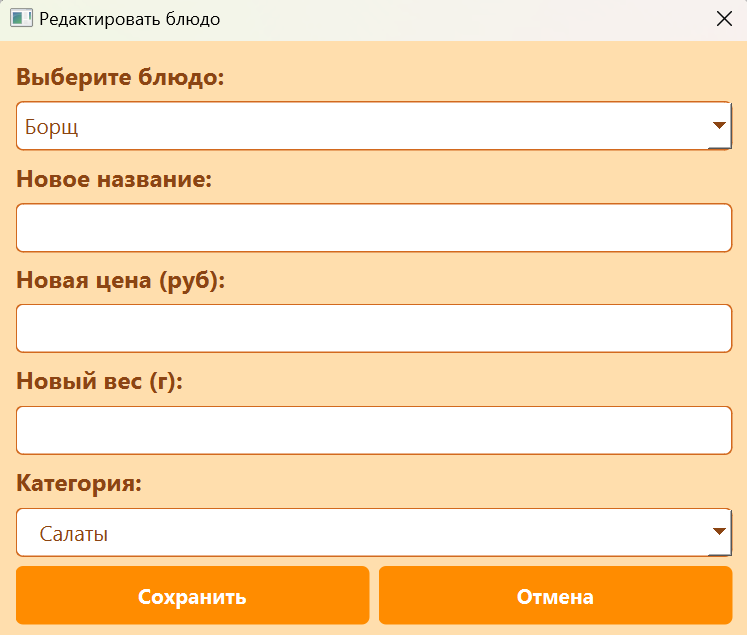


Рисунок 12 – Окно для редактирования блюда в меню

Кроме того, окно менеджера предоставляет возможность создания отчета по продажам с возможностью указания временного промежутка (по умолчанию — одна неделя, где конечная дата отчета совпадает с текущим днем). (см. Рисунок 13)

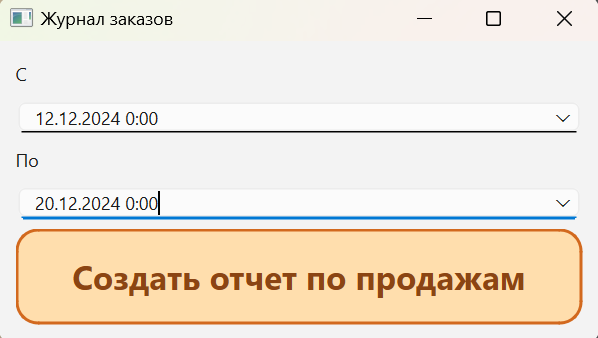


Рисунок 13 – Окно для создания отчета по продажам

Функция "Создать отчет по продажам" формирует Excel-документ с перечнем заказов, их детализацией и общей выручкой. (см. Рисунки 14.1, 14.2)

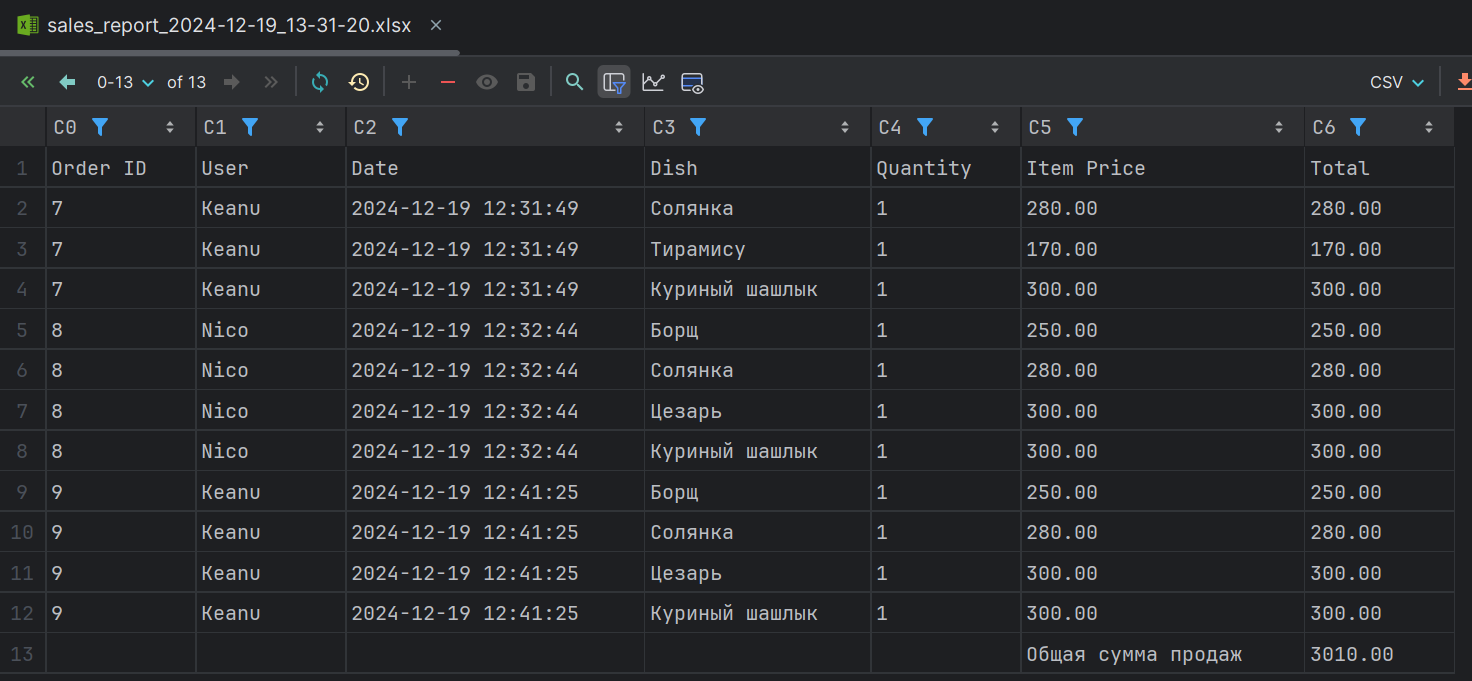


Рисунок 14.1 – Вывод отчета по продажам в PyCharm

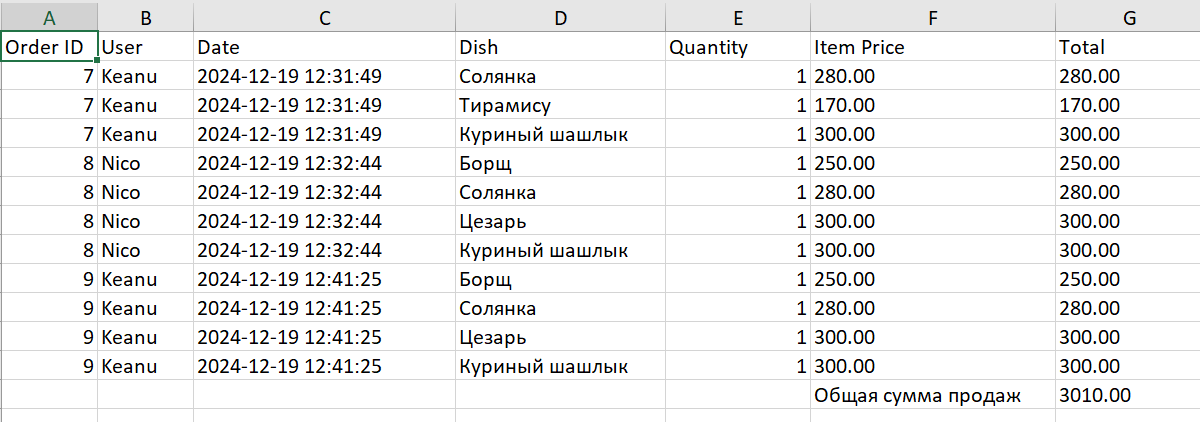


Рисунок 14.2 – Вывод отчета по продажам в Excel

Интерфейс не только упрощает взаимодействие пользователя с программой, но и делает процесс управления ресторанным меню логичным и интуитивным.

## 2.4 Написание программного кода

При запуске программы происходит инициализация базы данных, если таковая отсутствует. Это включает создание всех необходимых таблиц, таких как Users, Dish, Dish\_category, Orders и других. После этого открывается главное окно программы, которое предлагает пользователю выбрать роль: гость или менеджер. (см. Приложение 1)

В случае выбора роли менеджера пользователь вводит логин и пароль, которые проверяются на основе данных из базы. Авторизация реализована в методе on\_login. Этот метод находится в классе LoginWindow, который отвечает за запуск окна авторизации. Пользователь выбирает свою роль через ComboBox, а затем нажимает кнопку «Войти». Если выбран «Гость», проверяется логин и открывается окно гостя, создавая экземпляр класса GuestWindow. Если выбран «Менеджер», проверяются логин и пароль, после чего создается экземпляр класса ManagerWindow. (см. Приложение 2)

При успешной авторизации в роли менеджера открывается окно, где предоставляются функции просмотра и управления меню, а также работа с журналом заказов. Реализация показа текущее меню отображается в классе MenuWindow, в котором выполняется выборка данных из базы данных и заполняется таблица. Меню доступно только для чтения, а поле ID скрыто. (см. Приложение 3)

Редактирование меню реализовано в классах AddDishDialog, EditDishDialog, DeleteDishDialog в модуле modify.py. Необходимые методы в этих классах вызываются соответствующими кнопками и взаимодействуют с базой данных, добавляя, обновляя или удаляя записи о блюдах. Логика кнопки "Добавить" позволяет ввести название, цену, вес, категорию блюда и путь к изображению. Важно, что категория может быть как выбрана из существующих, так и добавлена вручную. При редактировании блюда менеджер может обновить любую информацию о нем, а при удалении — полностью удалить его из базы данных. (см. Приложение 4)

Журнал заказов содержит в себе кнопку «Создать отчет по продажам», которая связана с модулем report.py и прописана в методе generate\_sales\_report, который вызывается в классе ManagerWindow. Метод формирует Excel-файл с отчетом о продажах, используя библиотеку openpyxl. (см. Приложение 5)

Для гостей доступно окно меню, где они могут видеть список блюд с указанием их цены, веса, категории и кнопкой выбора. Отображение меню для гостя реализовано в методе load\_menu\_data класса GuestWindowAdmin. Этот метод выполняет выборку данных из базы данных и создает элементы интерфейса с картинками, названиями блюд, ценами и флажками для выбора.

Логика кнопки "Заказать" прописана в классe GuestWindowAdmin, а создание заказа на основе выбранных блюд прописана в методе create\_order того же класса. (см. Приложение 6)

Этот метод собирает данные о выбранных блюдах и вызывает окно подтверждения заказа, создавая экземпляр класса OrderConfirmationWindow (модуль guest\_order\_conf). При подтверждении заказа данные о нем заносятся в таблицы Orders и Order\_Items в базе данных. (см. Приложение 7)

Кроме того, для гостя формируется чек в формате Excel, включающий название блюда, его цену и общую стоимость заказа. Генерация чека прописана в модуле report.py через метод generate\_order\_receipt и вызывается в классе GuestWindowAdmin. (см. Приложение 5)

# 3. ОТКЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ

Тестирование программы управления ресторанным меню необходимо для проверки правильности её работы и обеспечения надежности.

Важно проверять такие функции, как:

Авторизация пользователей: Проверка правильности входа в систему как для менеджера, так и для гостя. Это включает в себя проверку правильности логина и пароля у менеджера и корректного отображения интерфейса для гостей.

Управление меню: Проверка возможностей менеджера добавлять, редактировать и удалять блюда в меню, а также изменение цен. Тестирование также включает в себя проверку работы кнопки для добавления блюда в стоп-лист и отображение недоступных блюд для гостей.

Выбор блюд гостями: Проверка работы интерфейса выбора блюд гостями, включая правильность отображения блюд с картинками и ценами, а также возможность отмечать блюда с помощью флажков и оформления заказа.

Журнал заказов: Проверка корректности отображения заказов в журнале менеджера и возможности создания отчетов по продажам. Тестирование включает в себя генерацию отчета в Excel, который должен содержать правильные данные о суммах заказов.

Окно подтверждения заказа: Проверка работы окна подтверждения заказа, чтобы убедиться, что при нажатии на кнопку "ОК" заказ фиксируется, а при нажатии на кнопку "Отмена" происходит возврат в меню выбора.

Тестирование этих функций важно для того, чтобы убедиться, что все ключевые элементы программы работают корректно, данные обрабатываются правильно, а пользователи (менеджеры и гости) могут взаимодействовать с системой без ошибок. Это гарантирует, что программа будет удобной и надежной в эксплуатации. (см. Приложение 8)

# ЗАКЛЮЧЕНИЕ

В рамках разработки программного модуля "Система управления ресторанным меню" был создан функциональный программный продукт, который отвечает современным требованиям к удобству и эффективности управления меню ресторана.

В процессе работы были созданы база данных, включающая ключевые сущности, разработан графический интерфейс пользователя с использованием библиотеки PyQt6, обеспечивающий удобство работы для менеджера и гостя, реализованы основные функции приложения: управление блюдами и меню, формирование заказов гостями, а также создание отчетов для анализа продаж.

Внедрены механизмы авторизации для различных ролей (гость и менеджер), что гарантирует доступ к необходимому функционалу в зависимости от уровня пользователя.

Результаты работы подтверждают практическую применимость разработанного модуля в реальных условиях ресторанного бизнеса. Он позволяет автоматизировать управление меню, упростить процесс формирования заказов и обеспечить прозрачность учета продаж.

Таким образом, все этапы проекта успешно выполнены, и разработанное программное обеспечение полностью соответствует поставленным задачам и требованиям.

1. [R-Kepper](https://rkeeper.ru/) [↑](#footnote-ref-1)
2. [iiko](https://iiko.ru/) [↑](#footnote-ref-2)
3. [Poster POS](https://poster.kg/) [↑](#footnote-ref-3)