

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**«Система управления ресторанным меню»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнила студентка группы 307ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **И.А. Исаева** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **О.А. Калашникова** |

**Москва 2024**



**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**По дисциплине: МДК 01.01 Разработка программных модулей**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 307ИС-22 Интизар Исаева**

**ТЕМА: Разработка программного модуля «Система управления ресторанным меню»**

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

ВВЕДЕНИЕ

В условиях быстро меняющихся запросов клиентов и потребности в оптимизации процессов управления рестораном, автоматизация учёта и изменений меню становится важной задачей. Современные программные решения позволяют значительно упростить этот процесс, обеспечивая не только удобство для сотрудников, но и повышая качество обслуживания гостей.

Основной целью данной курсовой работы является разработка программного модуля для управления ресторанным меню. Разработанная система должна позволить управлять ассортиментом блюд, их категориями, ценами, а также учитывать изменения в доступности блюд. 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.

В рамках работы было создано десктопное приложение с графическим интерфейсом, использующее базу данных для хранения информации об ингредиентах, блюдах, категориях и других элементах меню. Реализованные функции включают добавление, удаление и редактирование блюд, вывод текущего меню в табличном виде, а также формирование отчётов о продажах. Система поддерживает разграничение доступа, предоставляя гостям упрощённый интерфейс для выбора блюд, а менеджерам — расширенный функционал для управления ресторанным процессом.

Программа была разработана с использованием языка программирования Python, реляционной базы данных SQLite и библиотеки PyQt6 для создания удобного и интуитивно понятного пользовательского интерфейса. Уделено внимание не только функциональной части, но и эстетике приложения, что особенно важно в условиях высоких требований к современным программным продуктам.

Работа направлена на создание эффективного инструмента для ресторанов, который поможет автоматизировать процесс управления меню и улучшить взаимодействие с клиентами. Это способстует снижению количества ошибок и более качественному предоставлению услуг.

Важно, чтобы программа оптимизировала рабочие процессы, сокращая время на внесение изменений в меню, обеспечивая актуальность данных и возможность быстро получать отчёты о состоянии меню. Также система должна обеспечивать прозрачность процессов для гостей ресторана, что способствует повышению их лояльности.

Объектом 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.

Предметом исследования стали методы и средства разработки программных модулей, позволяющих реализовать автоматизированные системы управления в условиях ресторанного бизнеса, включая работу с базой данных, графическим интерфейсом и обработкой данных о блюдах.

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 отчётности в виде чека в формате Excel. Система проектировалась с учётом масштабируемости и возможности её дальнейшего расширения.

1 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.

Информационное обеспечение системы базируется на данных, описывающих блюда, их категории, ингредиенты, а также на сведениях о весе блюд.

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

* + Блюда: название, цена, вес, категория.
  + Категории блюд: название категории блюда (например, напитки, десерты, горячие блюда).
  + Ингредиенты: название ингредиента, единица измерения.
  + Меню: название типа меню (например, завтрак, обед, ужин).

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.
  + Гость ресторана, который 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 описания, удаление блюд, которые больше не входят в меню, и редактирование данных о блюдах, включая изменения цены, веса, описания и доступности.
* Управление категориями блюд: создание категорий для структурирования меню, таких как "Супы", "Основные блюда", "Напитки", с возможностью их удаления и редактирования.
* Работа с ингредиентами: ведение базы данных ингредиентов, необходимых для приготовления блюд, включая добавление, удаление и редактирование этих данных.
* Формирование меню: создание меню на основе категорий и блюд с разделением по времени дня (завтрак, обед, ужин), а также отображение текущего меню для гостей и сотрудников ресторана.

1.2 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. Особенность системы — глубокая интеграция с POS-терминалами и другой периферией, что делает её удобной для крупных ресторанных сетей.
* Iiko автоматизирует ресторанные процессы, включая управление меню, учет складских остатков, анализ продаж и поддержку онлайн-заказов. Важной особенностью является гибкость настроек акций и скидок, а также интеграция с CRM для лояльности клиентов.
* Poster POS фокусируется на простоте использования для небольших заведений. Функционал включает управление меню, учет заказов и генерацию отчетов. Дополнительно поддерживается удаленный доступ к данным через облако.

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.

Программный модуль для управления ресторанным меню, ориентированный на небольшие и средние рестораны, должен предоставлять ключевые функции — управление меню, учет заказов, создание отчетов и взаимодействие с клиентами. Такое решение не только упрощает ежедневные операции, но и позволяет поддерживать актуальность данных, адаптироваться к изменяющимся условиям и обеспечивать конкурентоспособность заведения.

1.3 Постановка задачи. Структура входной и выходной информации

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.

* 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.

2 ПРОЕКТИРОВАНИЕ

2.1 Блок-схема работы и диаграмма классов

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

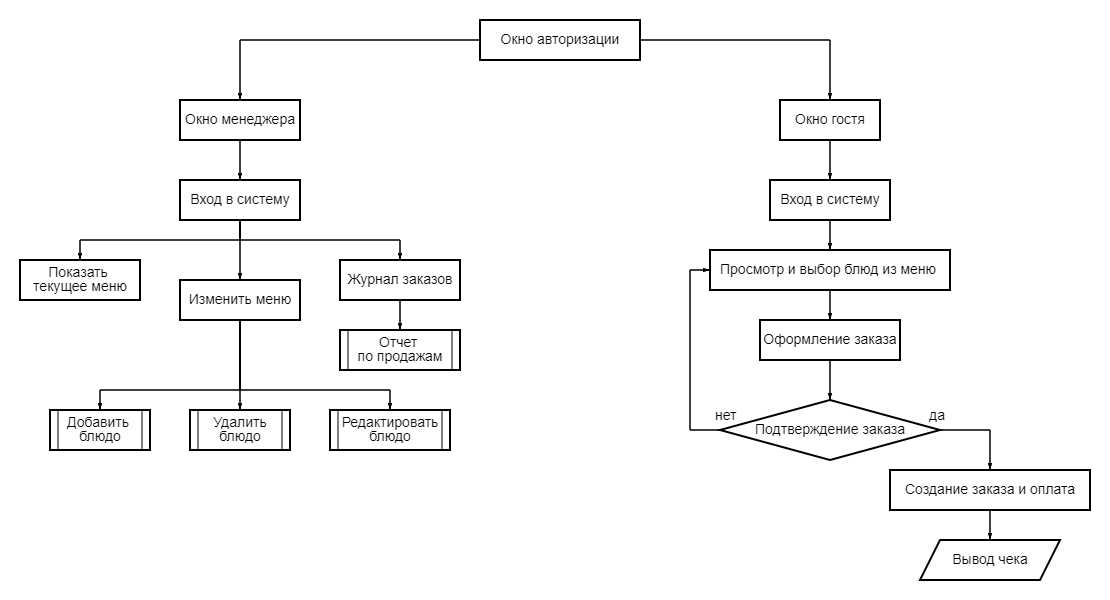


Рисунок 1 – Блок-схема работы программы

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 (не реализовано):

Нажимает кнопку "Журнал заказов".

В следующем окне нажимает кнопку "Отчет по продажам".

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

Отчет сохраняется на диск, 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 "ОК", чтобы 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.

2.2 Диаграмма прецедентов

Авторизация

Гость

Менеджер

При авторизации

в роли менеджера

При авторизации

в роли гостя

Рисунок 2 – Диаграмма прецедентов

Диаграмма прецедентов необходима для визуализации взаимодействия пользователей с системой.

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

* Менеджер

Просмотр текущего меню в виде таблицы.

Добавление, удаление и редактирование блюд.

Работа с журналом заказов: просмотр информации о заказах, 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.

2.3 Описание базы данных  
2.4 Разработка программного кода и интерфейса

2.4.1 Реализация базы данных

2.4.2 Реализация интерфейса

2.4.3 Реализация программного кода

3 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

3.1 Тестирование программного модуля

3.2 Отладка программного продукта