Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

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

Отчет по лабораторной работе №5

По дисциплине: Спецкурс

|  |  |
| --- | --- |
|  | Преподаватель  Л.Р. Набиева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
|  | Студент гр. 20ИС-1  М.С. Пермякова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

2024

Лабораторная работа № 5

**Задания:**

**«Модуль 1. Проектирование и разработка информационных систем.**

**Задание 1.1.** Разработка пользовательских историй и сценариев использования.

* + 1. Заполните Таблицу 1, описав взаимодействие пользователей системы (по ролям) с функциональными модулями системы. Описание должно отражать действия пользователя в зависимости от его роли и реакцию системы на это действие. Описание строится на основе предметной области.
    2. Заполните Таблицу 2, описав пользовательские истории в зависимости от роли пользователя. Описание должно включать роль пользователя системы, производимые им действия и цели этих действий. У одного действия, не может быть более одной цели.

**Задание 1.2.** Постановка задач по разработке информационной системы. На основании описания предметной области, пользовательских историй и сценариев.

* + 1. Заполните Таблицу 3, описав задачи по обработке данных функциями информационной системой

**Задание 1.3.** Разработка прототипов пользовательских интерфейсов системы.

1.3.1 На основании описания предметной области, пользовательских историй, сценариев и задач по обработке данных функциями информационной системы, разработайте прототипы основных пользовательских интерфейсов системы.

**Модуль 2: Осуществление интеграции программных модулей**

**Задание 2.1.** Разработка тестовых сценариев.

2.1.1 Разработайте минимум 2 тестовых сценария для проверки функционала каждой из ролей.

**Модуль 3: Сопровождение информационных систем**

**Задание 3.1.** Разработка руководства пользователя.

3.1.1 Для информационной системы, для управления заведениями общественного питания типа – кафе, разработайте руководство пользователя. Напишите руководство пользователя для разработанной вами информационной системы. В руководстве пользователя должен быть описан функционал для каждой роли. Руководство пользователя должно включать в себя текстовое описание работы с системой и скриншоты пользовательских интерфейсов.

**Ход работы:**

**Задание 1.1.** Разработка пользовательских историй и сценариев использования.

В таблицах 1.1.1 – 1.1.2 представлены пользовательские истории и сценарии использования.

Таблица 1.1.1 пользовательские истории

|  |  |  |
| --- | --- | --- |
| Роль пользователя | Действие пользователя | Реакция системы |
| 1 | 2 | 3 |
| Администратор | Авторизация в системе | Открывается главное окно администратора со списком всех сотрудников, заказов и смен. |
| Администратор | Переход к списку сотрудников | Отображается список сотрудников с возможностью изменения статуса на "уволен" и добавления нового сотрудника. |
| Администратор | Переход к списку заказов | Отображается список всех заказов. |
| Администратор | Переход к списку смен | Отображается информация о сменах с возможностью формирования новых смен. |
| Администратор | Регистрация нового пользователя | Открывается окно с формой для ввода данных нового пользователя, после чего система добавляет пользователя в базу данных. |
| Администратор | Перевод пользователя в статус "уволен" | Система изменяет статус сотрудника на "уволен". |

Продолжение таблицы 1.1.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Повар | Авторизация в системе | Открывается главное окно повара со списком принятых заказов. |
| Повар | Просмотр заказов | Отображается список заказов с возможностью изменения их статуса (готовится, готов). |
| Повар | Изменение статуса заказа | Система обновляет статус заказа (готовится, готов). |
| Официант | Авторизация в системе | Открывается главное окно официанта со списком принятых заказов за активную смену. |
| Официант | Просмотр заказов | Отображается список заказов за активную смену. |
| Официант | Создание нового заказа | Открывается окно с формой для создания нового заказа, где указываются места (столик), количество клиентов, заказанные блюда и напитки. |
| Официант | Изменение статуса заказа | Система обновляет статус заказа (принят, оплачен). |

Таблица 1.1.2 сценарии использования

|  |  |  |
| --- | --- | --- |
| Роль пользователя | Действие пользователя | Цель действия |
| 1 | 2 | 3 |
| Администратор | Авторизация в системе | Вход в главное окно администратора для управления кафе. |
| Администратор | Просмотр списка сотрудников | Мониторинг и управление персоналом, в том числе изменение статуса сотрудника. |
| Администратор | Просмотр списка заказов | Оценка текущей загруженности и статуса заказов в кафе. |
| Администратор | Просмотр списка смен | Планирование и управление рабочими сменами в кафе. |
| Администратор | Регистрация нового пользователя | Добавление нового сотрудника для последующей работы в системе. |

Продолжение таблицы 1.1.3

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Администратор | Перевод пользователя в статус "уволен" | Управление штатом персонала, увольнение сотрудников. |
| Повар | Авторизация в системе | Вход в главное окно повара для работы с заказами. |
| Повар | Просмотр списка заказов | Оценка текущих заказов, подготовка блюд и обновление их статуса. |
| Повар | Изменение статуса заказа | Отметка о готовности заказа для официанта. |
| Официант | Авторизация в системе | Вход в главное окно официанта для принятия заказов и обслуживания клиентов. |
| Официант | Просмотр списка заказов | Оценка текущих заказов, ожидающих выполнения. |
| Официант | Создание нового заказа | Принятие заказа от клиента с указанием места, количества клиентов и заказанных блюд. |
| Официант | Изменение статуса заказа | Отметка о принятии и оплате заказа. |

**Задание 1.2.** Постановка задач по разработке информационной системы. На основании описания предметной области, пользовательских историй и сценариев.

В таблице 1.2.1 представлены задачи по обработке данных функциями информационной системой.

Таблица 1.2.1 задачи по обработке данных функциями

|  |  |  |  |
| --- | --- | --- | --- |
| Название функции | Входные параметры | Выходные параметры | Описание функции |
| 1 | 2 | 3 | 4 |
| Авторизация | Логин, пароль | Меню пользователя | Проверяет введенные логин и пароль, переходит в меню пользователя в случае успешной авторизации |

Продолжение таблицы 1.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Просмотр сотрудников | - | Список сотрудников | Возвращает список всех сотрудников кафе для отображения в интерфейсе администратора. |
| Изменение статуса сотрудника | ID сотрудника, Новый статус | - | Изменяет статус сотрудника (например, "работает" или "уволен"). |
| Просмотр заказов | - | Список заказов | Возвращает список всех заказов для отображения в интерфейсе администратора и повара. |
| Просмотр смен | - | Список смен | Возвращает информацию о текущих сменах для отображения в интерфейсе администратора. |
| Формирование смены | Дата, Время начала, Время окончания | - | Создает новую смену с указанными параметрами. |
| Регистрация нового пользователя | Данные нового пользователя | - | Добавляет нового сотрудника в базу данных. |
| Перевод в статус "уволен" | ID сотрудника | - | Изменяет статус сотрудника на "уволен". |
| Просмотр заказов поваром | - | Список заказов | Возвращает список заказов для отображения в интерфейсе повара. |
| Изменение статуса заказа поваром | ID заказа, Новый статус | - | Обновляет статус заказа (например, "готовится" или "готов"). |

Продолжение таблицы 1.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Создание заказа официантом | Места, Количество клиентов, Список блюд и напитков | ID нового заказа | Создает новый заказ с указанными параметрами и возвращает его идентификатор. |
| Изменение статуса заказа официантом | ID заказа, Новый статус | - | Обновляет статус заказа (например, "принят" или "оплачен"). |

**Задание 1.3.** Разработка прототипов пользовательских интерфейсов системы.

На рисунках 1.3.1 – 1.3.10 представлены прототипы пользовательских интерфейсов системы.

На рисунке 1.3.1 представлен прототип окна «Авторизация».

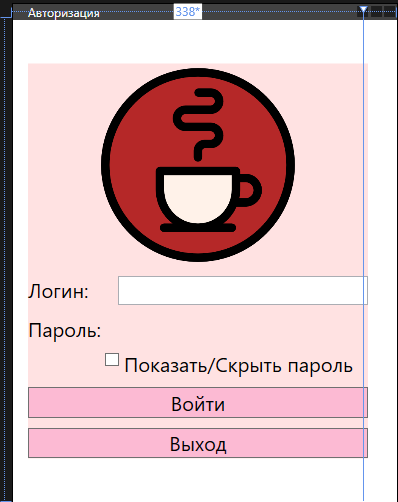


Рисунок 1.3.1 – Прототип окна «Авторизация»

На рисунке 1.3.2 представлен прототип окна «Меню администратора»

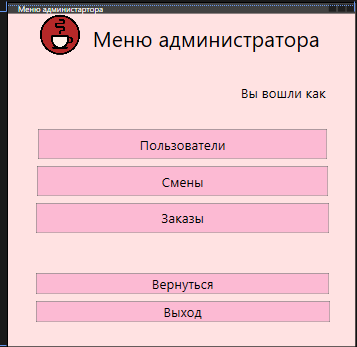


Рисунок 1.3.2 – Прототип окна «Меню администратора»

На рисунке 1.3.3 представлен прототип окна «Сотрудники».

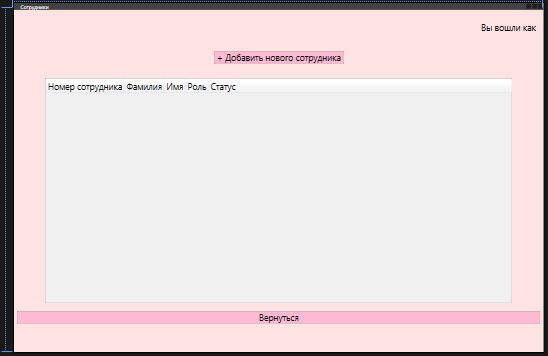


Рисунок 1.3.3 – Прототип окна «Сотрудники»

На рисунке 1.3.4 представлен прототип окна «Добавить сотрудника».

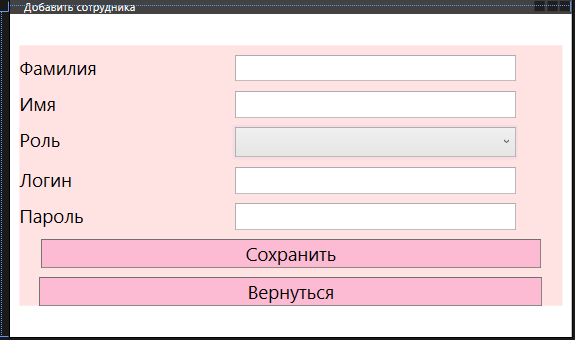


Рисунок 1.3.4 – прототип окна «Добавить сотрудника»

На рисунке 1.3.5 представлен прототип окна «Смены».

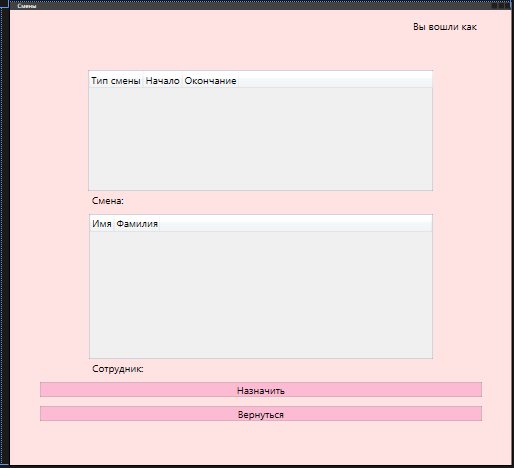


Рисунок 1.3.5 – Прототип окна «Смены»

На рисунке 1.3.6 представлен прототип окна «Меню повара».

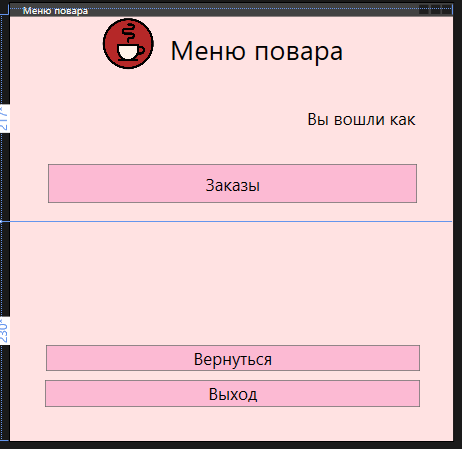


Рисунок 1.3.6 – Прототип окна «Меню повара»

На рисунке 1.3.7 представлен прототип окна «Заказы повара».

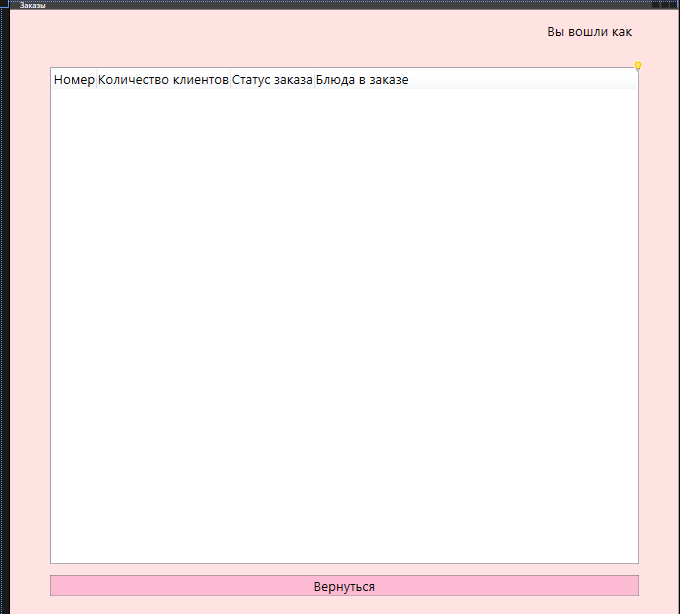


Рисунок 1.3.7 – Прототип окна «Заказы Администратора»

На рисунке 1.3.8 представлен прототип окна «Заказы повара»

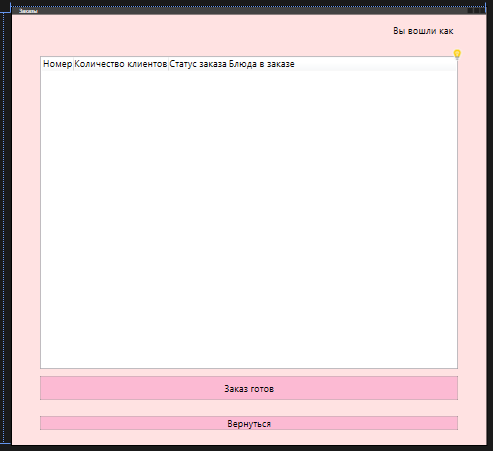


Рисунок 1.3.8 – Прототип окна «Заказы повара»

На рисунке 1.3.9 представлен прототип окна «Заказы официанта»

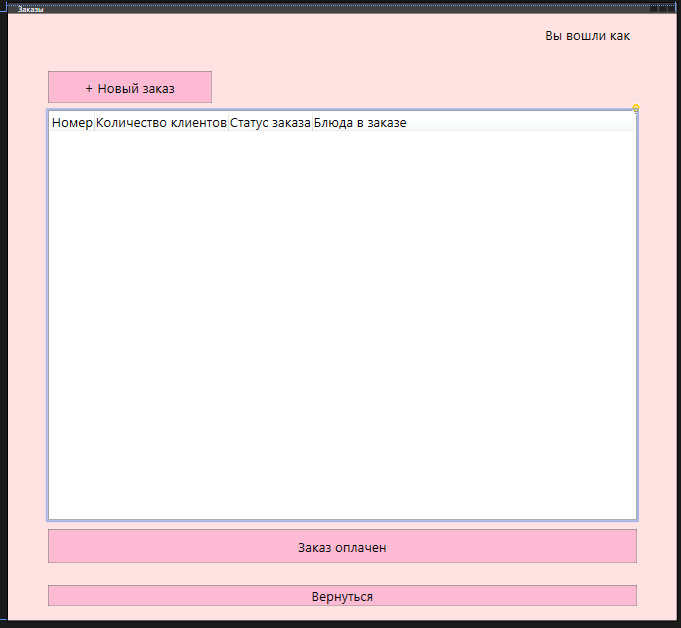


Рисунок 1.3.9 – Прототип окна «Заказы официанта»

На рисунке 1.3.10 представлен прототип окна «Создание заказа»

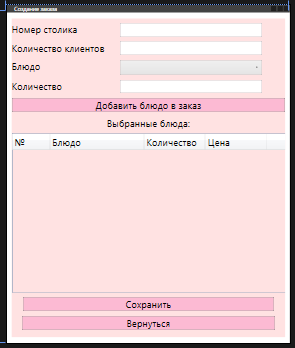


Рисунок 1.3.10 – Прототип окна «Создание заказа»

**Задание 2.1.** Разработка тестовых сценариев.

Таблица 2.1.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | CafeManagement |
| Номер версии | 1.0 |
| Имя тестера | Пермякова Мария Сергеевна |
| Даты тестирования | 26.02.2024 |

Таблица 2.1.2 – Протокол тестирования проверки назначения сотрудника на смену

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | Test\_1 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка назначения сотрудника на смену в окне ShiftsAdmin.xaml («Смены администратора») с корректным выбором данных |

Продолжение таблицы 2.1.2

|  |  |
| --- | --- |
| 1 | 2 |
| Резюме испытания | Необходимо обеспечить корректное поведение программы при назначении сотрудника на смену |
| Шаги тестирования | 1 Авторизация;  2 Переход на окно ShiftsAdmin.xaml («Смены администратора»);  3 Выбор смены;  4 Выбор сотрудника;  5 Нажатие кнопки «Назначить». |
| Данные тестирования | 1 Смена: Утренняя смена 2024-02-26 08:00:00 2024-02-26 16:00:00;  2 Сотрудник: Шакиров Марсель. |
| Ожидаемый результат | Программа должна корректно назначить сотрудника на смену на окне ShiftsAdmin.xaml («Смены администратора») и вывести уведомление |
| Фактический результат | В результате тестирования программа назначила сотрудника на смену на окне ShiftsAdmin.xaml («Смены администратора») и вывела уведомление |
| Предпосылки | Открытие страницы Authorization.xaml («Авторизация») |
| Постусловия | Система работает без сбоев и находится в рабочем состоянии |
| Статус (Pass/Fail) | Pass |

Проведём тестирование авторизации с правильным логином и паролем директора. Результаты изображены на рисунках 2.1.1 – 2.1.2.

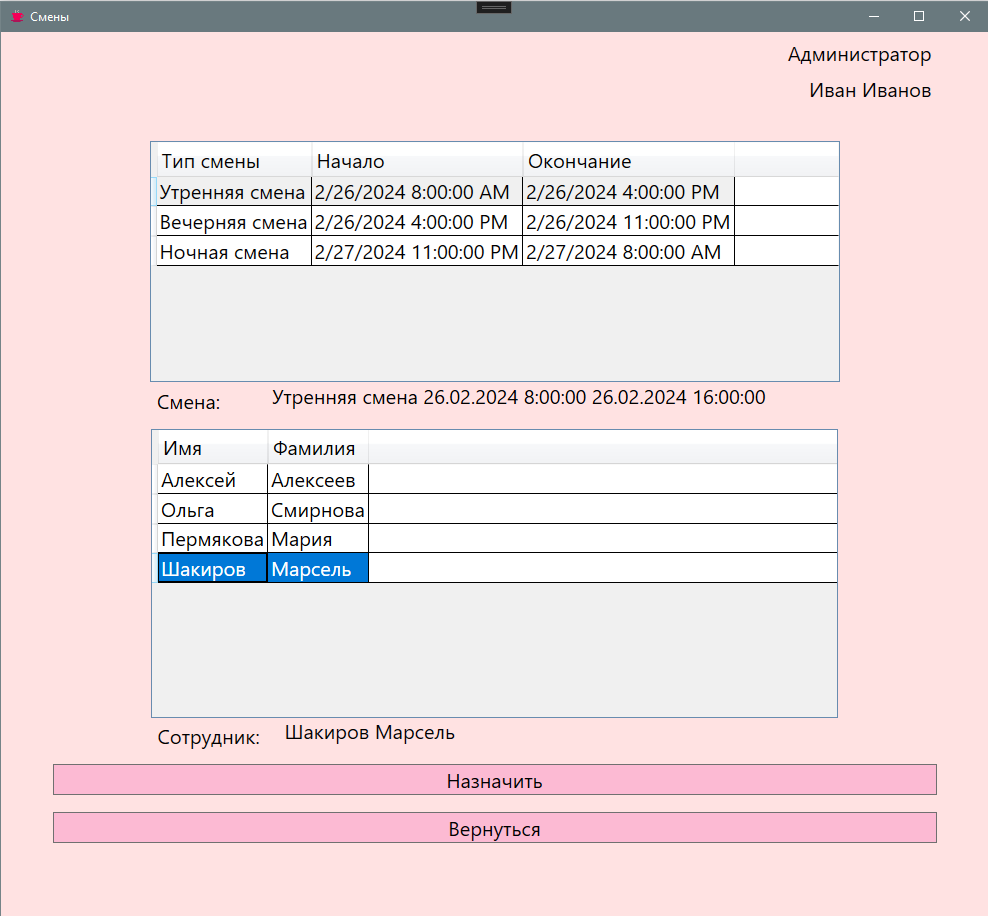


Рисунок 2.2.1 – Выбор данных для назначения на смену

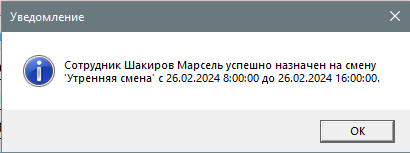


Рисунок 2.1.2 – Уведомление о назначении на смену

Таблица 2.1.3 – Протокол тестирования добавление сотрудника

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | Test\_2 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка добавление сотрудников в окне AddEmployeee.xaml («Добавление сотрудников») с корректными пользовательскими данными |
| Резюме испытания | Необходимо обеспечить корректное поведение программы при вводе верных данных и нажатия кнопки «Сохранить» |
| Шаги тестирования | 1 Авторизация;  2 Открытие окна «Добавление сотрудников»  3 Ввод корректных данных в текстовые поля;  4 Нажатие кнопки «Сохранить». |
| Данные тестирования | 1 Фамилия: Шакиров;  2 Имя: Марсель;  3 Роль: Повар;  4 Логин: Marsel;  5 Пароль: Loowe12. |
| Ожидаемый результат | Программа должна вывести сохранить данные и отобразить их в таблице на окне «UsersAdmin» («Сотрудники администратора») |
| Фактический результат | В результате тестирования программа сохранила данные и отобразила их в таблице на окне «UsersAdmin» («Сотрудники администратора») |
| Предпосылки | Открытие страницы Authorization.xaml («Авторизация») |

Продолжение таблицы 2.1.3

|  |  |
| --- | --- |
| 1 | 2 |
| Постусловия | Система работает без сбоев и находится в рабочем состоянии |
| Статус (Pass/Fail) | Pass |

Проведём тестирование добавления сотрудников с сохранением и корректным выводом. Результаты изображены на рисунках 2.1.3 – 2.1.4.

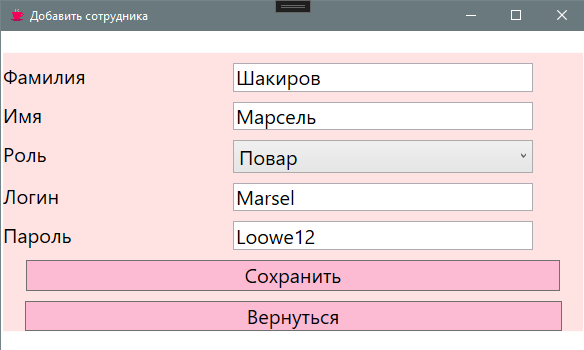


Рисунок 2.1.3 – Заполнение полей корректными данными

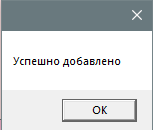


Рисунок 2.1.4 – Уведомление об успешном добавлении данных

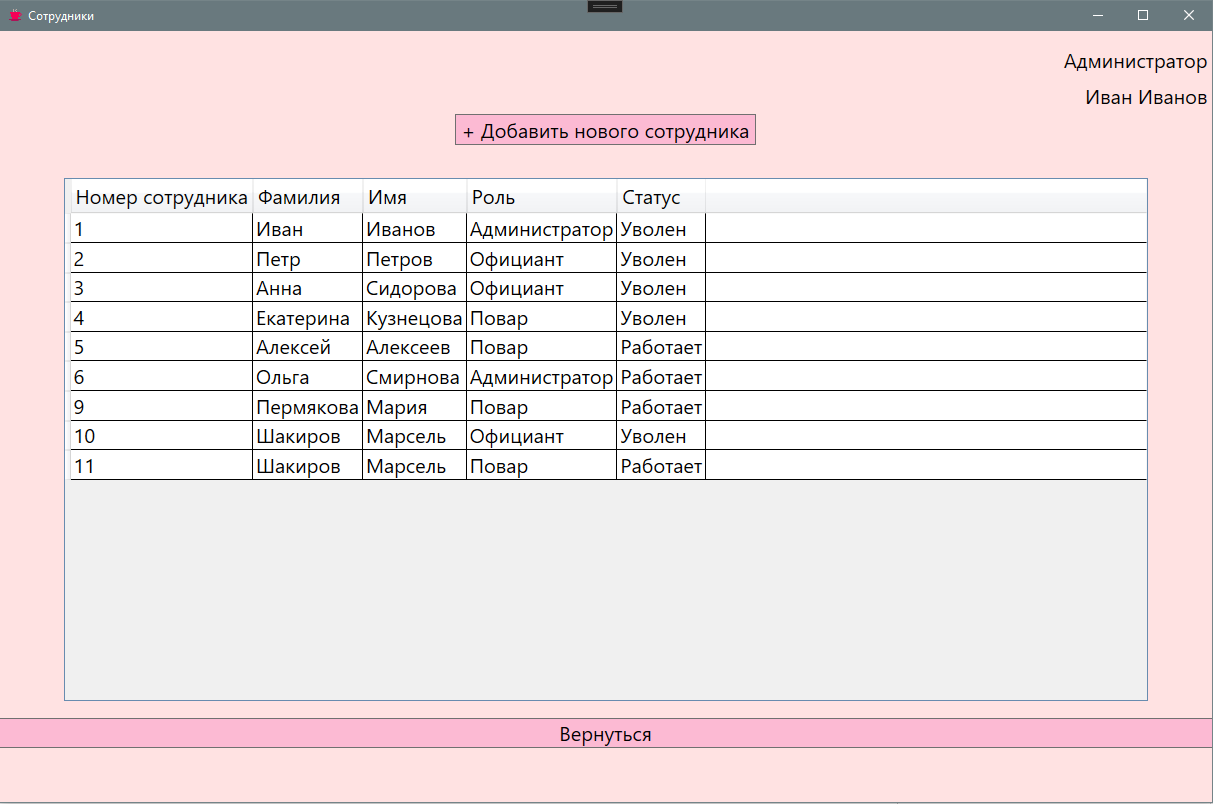


Рисунок 2.1.5 – Отображение новых данных

Таблица 2.1.4 – Протокол тестирования отображение данных на окне с заказами

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | Test\_3 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка на правильное отображение данных в таблице с заказами на окне «OrdersCook» («Заказы повара») и выбор нужного заказа путем двойного клика |
| Резюме испытания | Необходимо обеспечить корректное поведение программы при отображении данных в таблице с заказами и выбора заказа двойным нажатием |
| Шаги тестирования | 1 Авторизация;  2 Открытие окна «Заказы повара»  3 Двойной клик по нужному заказу |
| Данные тестирования | 1 Номер столика: 1;  2 Количество: 4;  3 Статус заказа: Оплачен;  4 Блюда в заказе: Борщ, Салат «Цезарь», «Лимонад». |
| Ожидаемый результат | Программа корректно отобразить данные в таблице с заказами на окне «OrdersCook» («Заказы повара») и правильно выбрать заказ путем двойного клика |
| Фактический результат | В результате тестирования программа отобразила данные в таблице с заказами на окне «OrdersCook» («Заказы повара») и правильно выбрала заказ путем двойного клика |
| Предпосылки | Открытие страницы Authorization.xaml («Авторизация») |
| Постусловия | Система работает без сбоев и находится в рабочем состоянии |
| Статус (Pass/Fail) | Pass |

Проведём тестирование отображения данных и выбора заказа. Результаты изображены на рисунках 2.1.6 – 2.1.7.



Рисунок 2.1.6 – Вывод данные о заказе

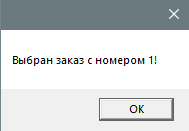


Рисунок 2.1.7 – Уведомление о выборе заказа

Таблица 2.1.5 – Протокол тестирования на смену статуса заказа

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | Test\_4 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка на смену статуса заказа на «Готов» |
| Резюме испытания | Необходимо обеспечить корректное поведение программы при смене статуса заказа на «Готов» |

Продолжение таблицы 2.1.5

|  |  |
| --- | --- |
| 1 | 2 |
| Шаги тестирования | 1 Авторизация;  2 Открытие окна «Заказы повара»;  3 Двойной клик по нужному заказу;  4 Нажатие кнопки «Заказ готов». |
| Данные тестирования | 1 Номер столика: 1;  2 Количество: 4;  3 Статус заказа: Оплачен;  4 Блюда в заказе: Борщ, Салат «Цезарь», «Лимонад». |
| Ожидаемый результат | Программа должна корректно сменить статус заказа на «Готов» в таблице с заказами на окне «OrdersCook» («Заказы повара») |
| Фактический результат | В результате тестирования программа успешно сменила статус на «Готов» в таблице с заказами на окне «OrdersCook» («Заказы повара») |
| Предпосылки | Открытие окна Authorization.xaml («Авторизация») |
| Постусловия | Система работает без сбоев и находится в рабочем состоянии |
| Статус (Pass/Fail) | Pass |

Проведём тестирование добавления сотрудников с сохранением и корректным выводом. Результаты изображены на рисунках 2.1.8 – 2.1.10.



Рисунок 2.1.8 – Выбор заказа для смены статуса

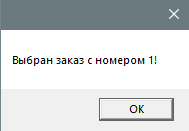


Рисунок 2.1.9 – Уведомление о выборе заказа

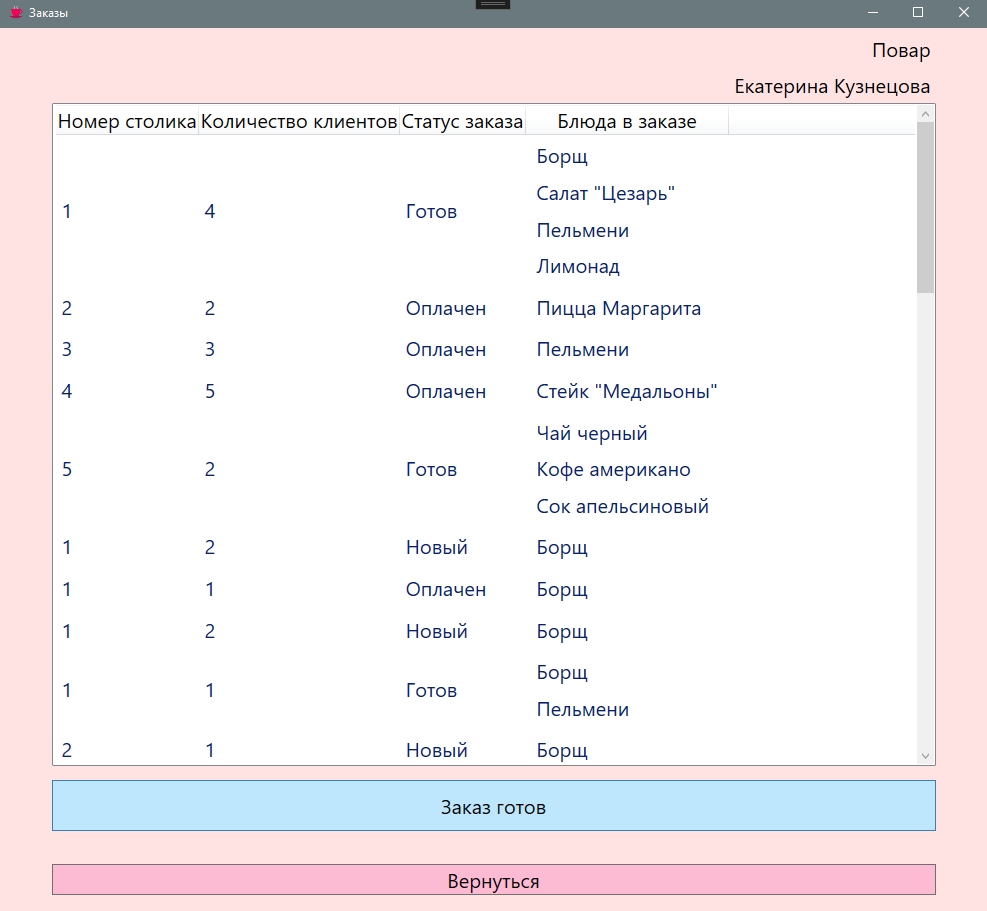


Рисунок 2.1.10 – Смена статуса на «Готов»

Таблица 2.1.6 – Протокол тестирования авторизации на ввод корректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | Test\_5 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка авторизации в окне Authorization.xaml («Авторизация») с корректными пользовательскими данными |
| Резюме испытания | Необходимо обеспечить корректное поведение программы при вводе верных данных |

Продолжение таблицы 2.1.6

|  |  |
| --- | --- |
| 1 | 2 |
| Шаги тестирования | 1 Ввод верных данных в текстовые поля;  2 Нажатие кнопки «Войти». |
| Данные тестирования | 1 Логин: 2;  2 Пароль: 2. |
| Ожидаемый результат | Программа должна вывести уведомление о приветствии и отобразить на окно «MenuWaiter» («Меню официанта») |
| Фактический результат | В результате тестирования программа вывела уведомление о приветствии и отобразила окно «MenuWaiter» («Меню официанта») |
| Предпосылки | Вывод уведомления об успешном входе и переход на окно «MenuWaiter» («Меню официанта») |
| Постусловия | Система работает без сбоев и находится в рабочем состоянии |
| Статус (Pass/Fail) | Pass |

Проведём тестирование авторизации с правильным логином и паролем директора. Результаты изображены на рисунках 2.1.11 – 2.1.13.

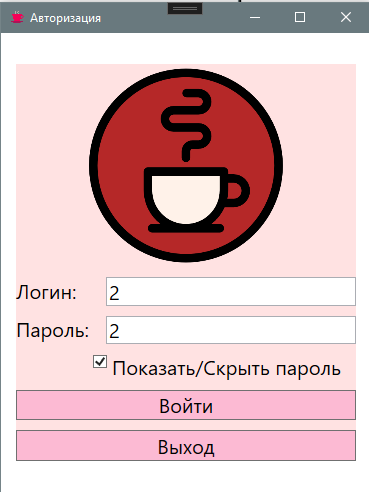


Рисунок 2.1.11 – Авторизация с вводом верных данных

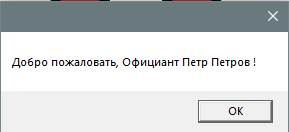


Рисунок 2.1.12 – Уведомление об успешном входе

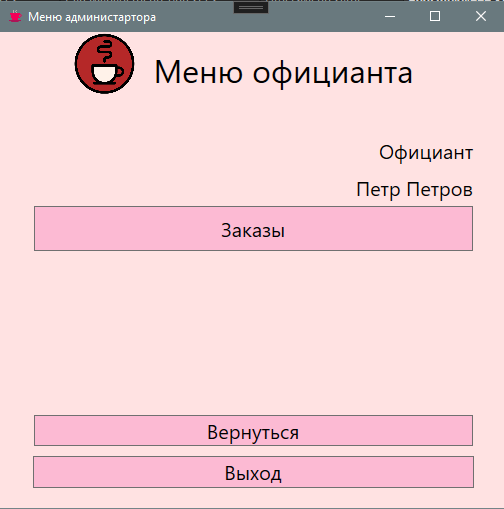


Рисунок 2.1.13 – Переход на окно «Меню официанта»

Таблица 2.1.7 – Протокол тестирования на создание нового заказа

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | Test\_6 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка создания нового заказа на окне «AddOrder» («Добавление заказа») |
| Резюме испытания | Необходимо обеспечить корректное поведение программы при создании нового заказа и заполнения его данными |

Продолжение таблицы 2.1.7

|  |  |
| --- | --- |
| 1 | 2 |
| Шаги тестирования | 1 Авторизация;  2 Переход на окно «Добавление заказа»;  3 Заполнение полей корректными данными;  4 Нажатие кнопки «Добавить блюдо»;  5 Нажатие кнопки «Сохранить». |
| Данные тестирования | 1 Номер столика: 1;  2 Количество клиентов: 1;  3 Блюдо: Борщ, Чай черный;  4 Количество: 1, 1. |
| Ожидаемый результат | Программа должна создать новый заказ на окне «AddOrder» («Добавление заказа») и отобразить все новые данные заказа на окне «OrdersWaiter» («Заказ официанта») |
| Фактический результат | В результате тестирования программа создала новый заказа на окне «AddOrder» («Добавление заказа») и отобразила все новые данные заказа на окне «OrdersWaiter» («Заказ официанта») |
| Предпосылки | Открытие окна Authorization.xaml («Авторизация») |
| Постусловия | Система работает без сбоев и находится в рабочем состоянии |
| Статус (Pass/Fail) | Pass |

Проведём тестирование авторизации с правильным логином и паролем директора. Результаты изображены на рисунках 2.1.14 – 2.1.16.

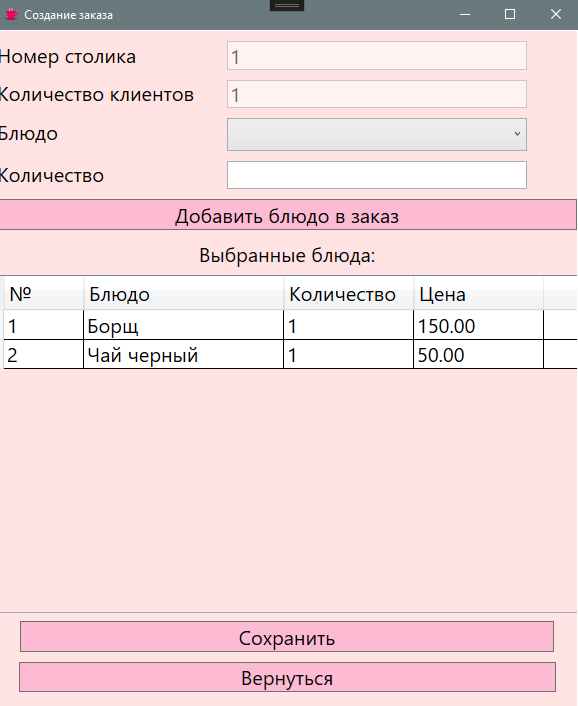


Рисунок 2.1.14 – Ввод корректных данных

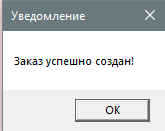


Рисунок 2.1.15 – Уведомление о успешном создании нового заказа



Рисунок 2.1.16 – Окно «Заказы официанта» с новыми данными

**Задание 3.1.** Разработка руководства пользователя.

Руководство администратора

При запуске программы будет показано окно «Авторизация», где пользователю потребуется ввести логин и пароль для входа.

На рисунке 3.1.1 показано окно «Авторизация».

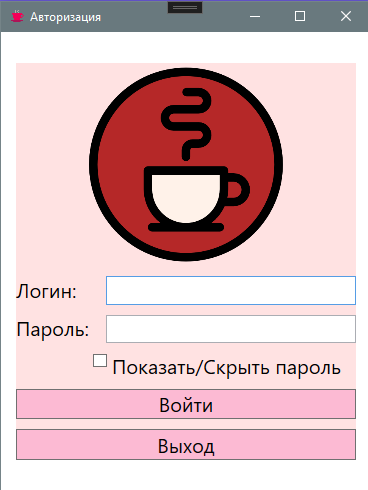


Рисунок 3.1.1 – Окно «Авторизация»

При вводе логина и пароля администратора, будет выполнен вход в программу и будет запущено окно «Меню администратора». На этом окне отображается фамилия и имя администратора, а также его должность, кнопки для перехода на другие окна, такие как: «Сотрудники», «Смены», «Заказы». Так же на окне будет отображаться кнопка «Вернуться», при нажатии на нее, пользователь выходит из системы и открывается окно «Авторизация». И кнопка «Выход», при нажатии на которую программа полностью закрывается Окно «Меню администратора» изображено на рисунке 3.1.2.

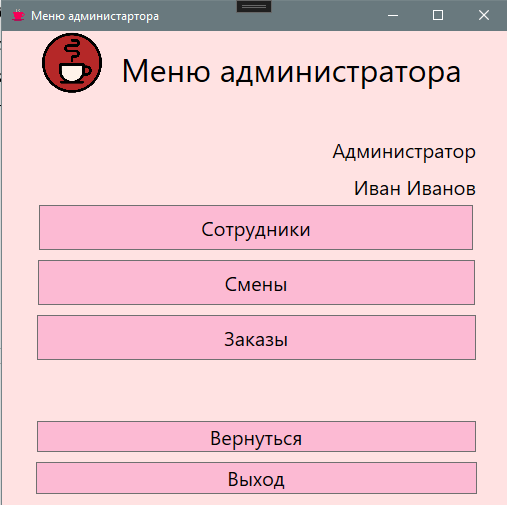


Рисунок 3.1.2 – Окно «Меню администратора»

При нажатии на кнопку «Сотрудники» открывается окно с информацией о сотрудниках.

На рисунке 3.1.3 представлено окно «Сотрудники».

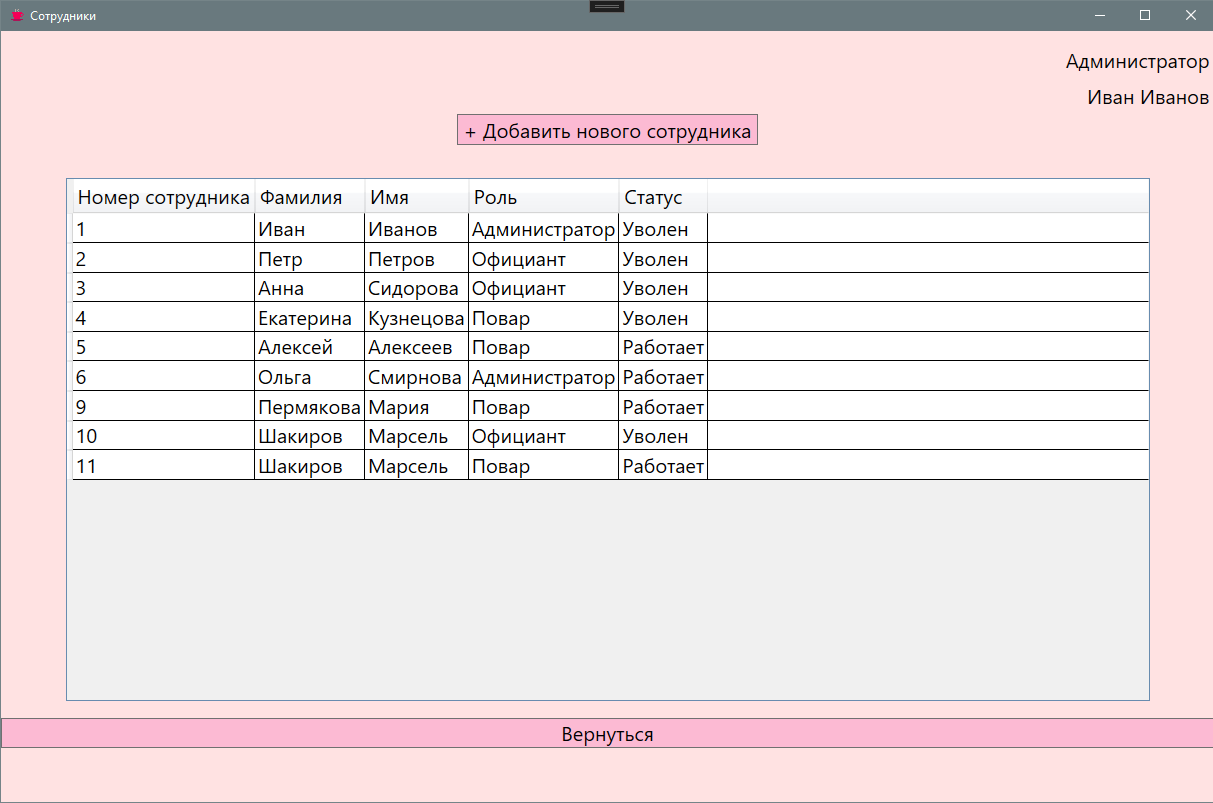


Рисунок 3.1.3 – Окно «Сотрудники»

При двойном щелчке по данным определенного сотрудника открывается уведомление для подтверждения смены статуса на «Уволен». Кнопка «Вернуться» позволяет вернуться на предыдущее окно и работает во всей программе одинаково. Уведомление представлено на рисунке 3.1.4.

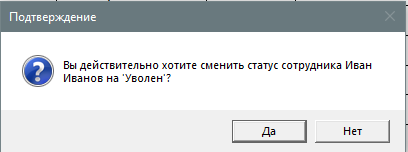


Рисунок 3.1.4– Уведомление о подтверждении смены статуса

В этом окне также представлена функциональная кнопка «Добавить нового сотрудника». При нажатии на которую открывается окно «Добавление сотрудника». На рисунке 3.1.5 представлено окно «Добавление сотрудника».

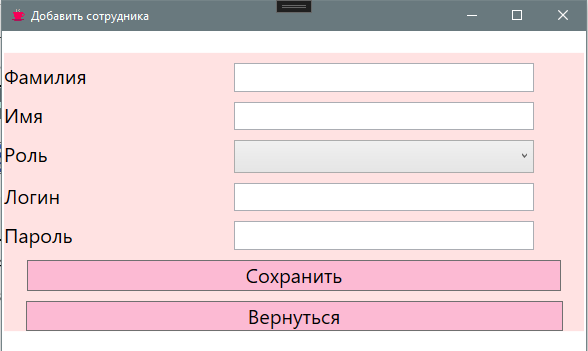


Рисунок 3.1.5 – Окно «Добавление сотрудника»

Для добавления сотрудника необходимо заполнить поля корректными данными и нажать на кнопку «Сохранить». После этого данные сохранятся и откроется окно «Сотрудники» уже с новыми данными.

При нажатии в меню на кнопку «Смены» откроется окно «Смены» с возможностью выбрать смену и назначить на неё сотрудника. На рисунке 3.1.6 представлено окно «Смены».

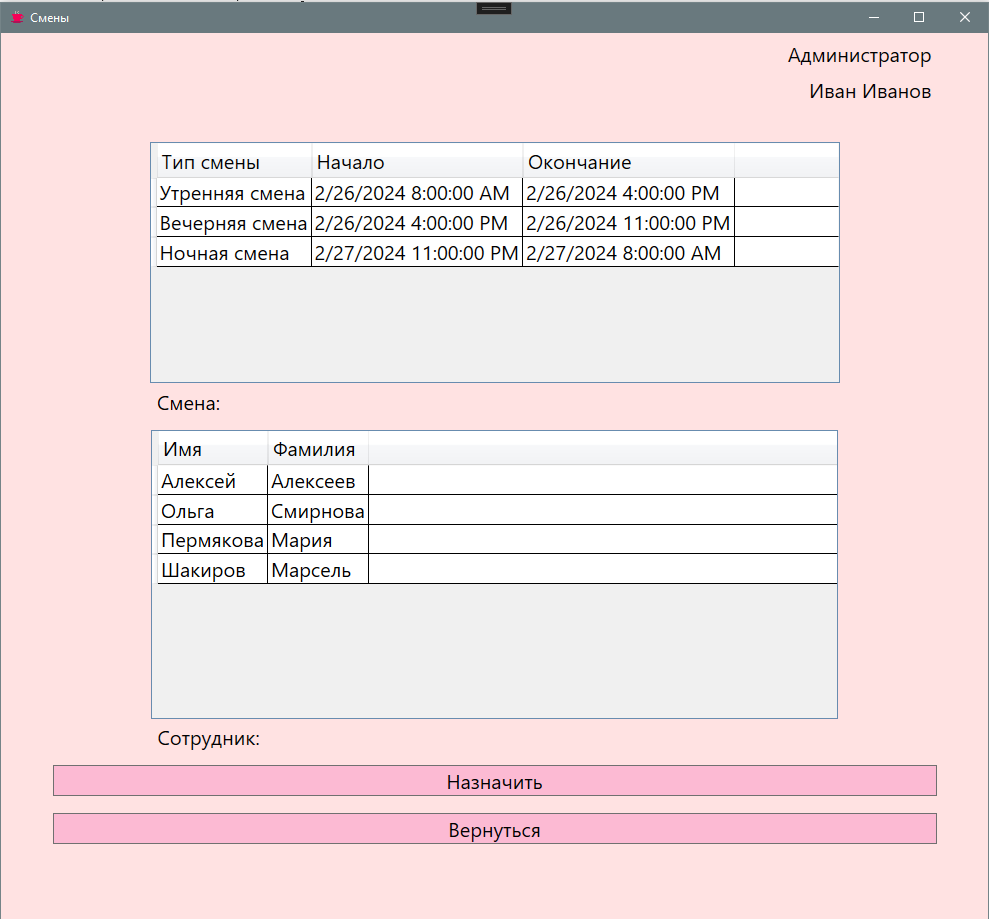


Рисунок 3.1.6 – Окно «Смены»

Для назначения сотрудника на смену необходимо двойным щелчком мыши выбрать данные из таблиц со сменами и сотрудниками. После того как данные выбраны необходимо нажать на кнопку «Назначить». При успешном назначении появится уведомление, которое представлено на рисунке 3.1.7.

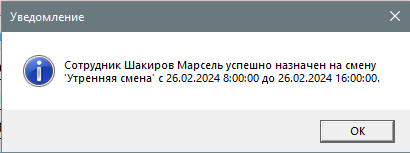


Рисунок 3.1.7 – Уведомление о назначении

Если сотрудник уже записан на выбранную смену система выдаст предупреждение о том, что сотрудник уже назначен. Предупреждение о том, что сотрудник уже назначен представлен на рисунке 3.1.8.

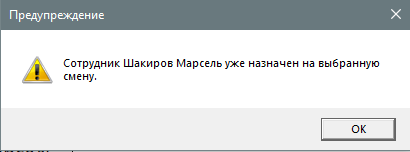


Рисунок 3.1.8 – Предупреждение о том, что сотрудник уже назначен

При нажатии в меню на кнопку «Заказы» откроется окно с заказами для просмотра. В таблице с заказами отображены такие данные как: номер столика, количество, статус заказа и блюда в заказе. На рисунке 3.1.9 представлено окно «Заказы».

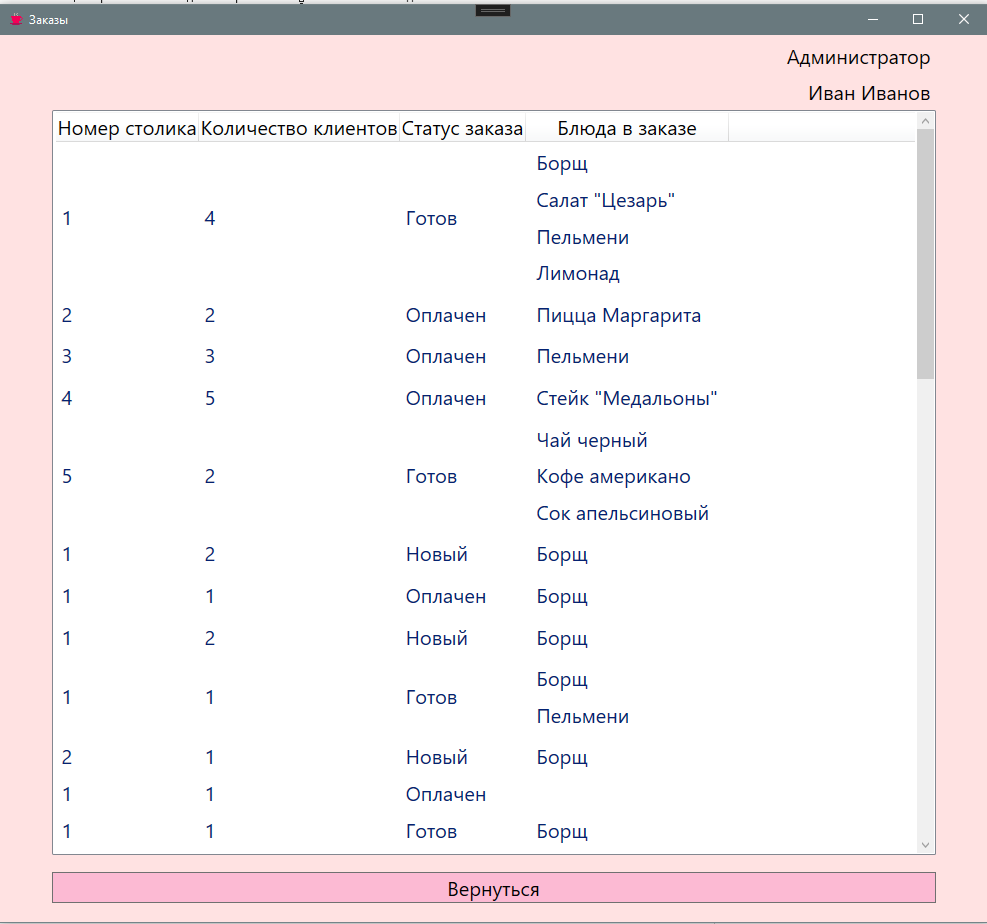


Рисунок 3.1.9 – Окно «Заказы»

Руководство повара

При запуске программы будет показано окно «Авторизация», где пользователю потребуется ввести логин и пароль для входа.

На рисунке 3.1.10 показано окно «Авторизация».

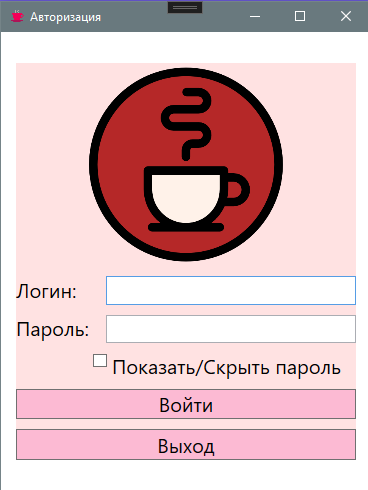


Рисунок 3.1.10 – Окно «Авторизация»

При вводе логина и пароля повара, будет выполнен вход в программу и будет запущено окно «Меню повара». На этом окне отображается фамилия и имя повара, а также его должность и кнопка «Заказы» для перехода на другое окно. Так же на окне будет отображаться кнопка «Вернуться», при нажатии на нее, пользователь выходит из системы и открывается окно «Авторизация». И кнопка «Выход», при нажатии на которую программа полностью закрывается Окно «Меню повара» изображено на рисунке 3.1.11.

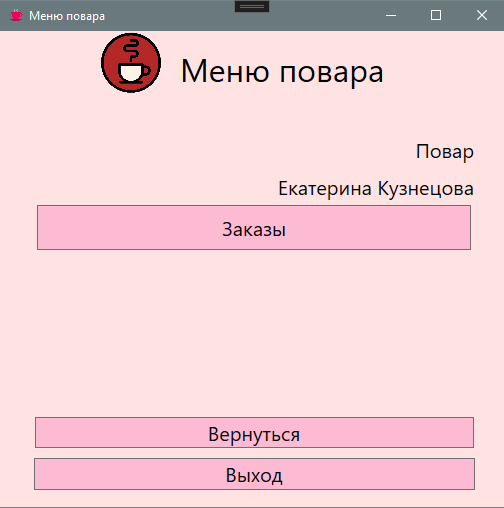


Рисунок 3.1.11 – Окно «Меню повара»

При нажатии на кнопку «Заказы» открывается окно с информацией о заказах.

На рисунке 3.1.12 представлено окно «Заказы».



Рисунок 3.1.12 – Окно «Заказы»

При двойном щелчке по данным определенного заказа открывается уведомление о выборе заказа. При нажатии на кнопку «Заказ готов», статус заказа меняется на «Готов». Кнопка «Вернуться» позволяет вернуться на предыдущее окно. Уведомление представлено на рисунке 3.1.13.

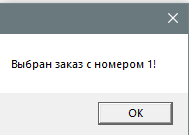


Рисунок 3.1.13– Уведомление о выборе заказа

Руководство официанта

При запуске программы будет показано окно «Авторизация», где пользователю потребуется ввести логин и пароль для входа.

На рисунке 3.1.14 показано окно «Авторизация».

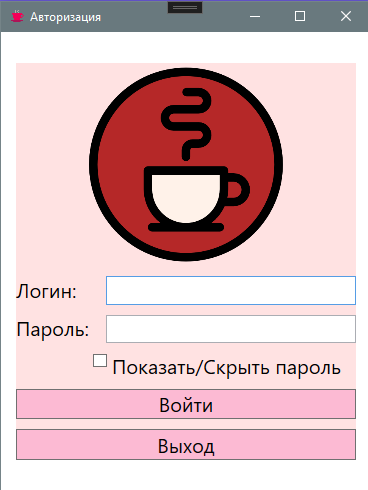


Рисунок 3.1.14 – Окно «Авторизация»

При вводе логина и пароля официанта, будет выполнен вход в программу и будет запущено окно «Меню официанта». На этом окне отображается фамилия и имя официанта, а также его должность и кнопка «Заказы» для перехода на другое окно. Так же на окне будет отображаться кнопка «Вернуться», при нажатии на нее, пользователь выходит из системы и открывается окно «Авторизация». И кнопка «Выход», при нажатии на которую программа полностью закрывается Окно «Меню официанта» показано на рисунке 3.1.15.

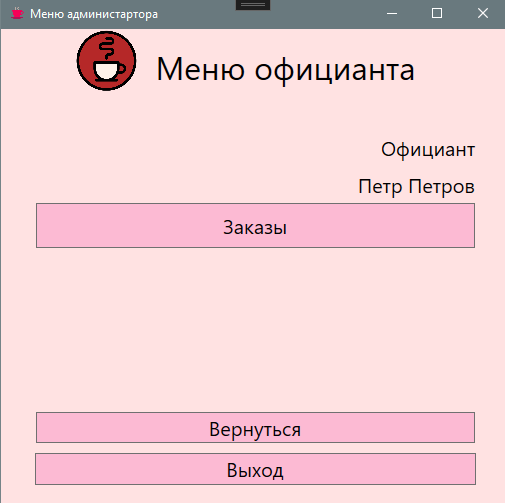


Рисунок 3.1.15 – Окно «Меню официанта»

При нажатии на кнопку «Заказы» открывается окно с информацией о заказах.

На рисунке 3.1.16 представлено окно «Заказы».



Рисунок 3.1.16 – Окно «Заказы»

При двойном щелчке по данным определенного заказа открывается уведомление о выборе заказа. При нажатии на кнопку «Заказ оплачен», статус заказа меняется на «Оплачен». Кнопка «Вернуться» позволяет вернуться на предыдущее окно. Уведомление представлено на рисунке 3.1.17.

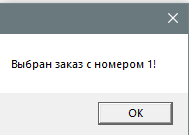


Рисунок 3.1.17– Уведомление о выборе заказа

При нажатии на кнопку «Новый заказ» открывается окно «Создание заказа», которое представлено на рисунке 3.1.18.

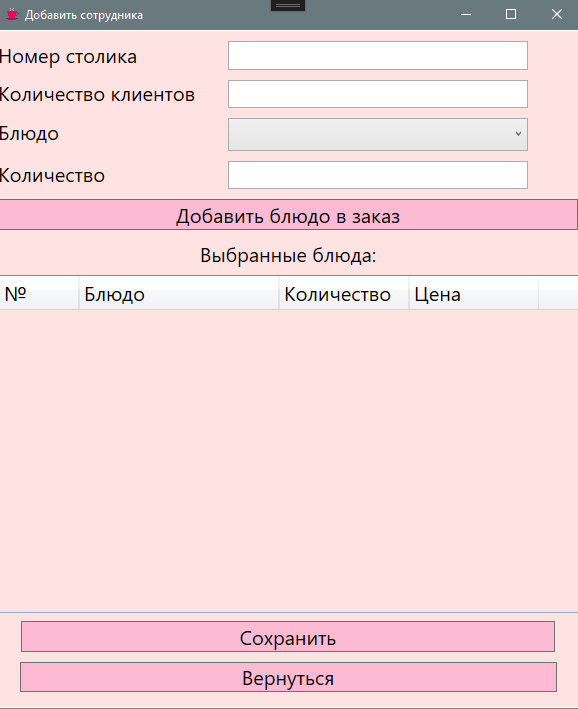


Рисунок 3.1.18 – Окно «Создание заказа»

Здесь необходимо заполнить все поля необходимыми корректными данными. После того как поля заполнены необходимо нажать на кнопку «Добавить блюдо в заказ». После этого изменить данные номера столика, количества клиентов будет невозможно. Добавляем все необходимые блюда в заказ и нажимаем кнопку «Сохранить». Заказ сохраняется со статусом «Новый».  
  
  
  
  
скрипт бд

CREATE DATABASE CafeManagement;

USE CafeManagement;

CREATE TABLE Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Username VARCHAR(50) NOT NULL,

Password VARCHAR(50) NOT NULL,

Role VARCHAR(20) NOT NULL

);

select \* from Users;

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Status VARCHAR(20) NOT NULL,

UserID INT,

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

CREATE TABLE Shifts (

ShiftID INT PRIMARY KEY AUTO\_INCREMENT,

ShiftType VARCHAR(50) NOT NULL,

StartTime DATETIME NOT NULL,

EndTime DATETIME NOT NULL

);

select\*from Employees;

CREATE TABLE ShiftEmployees (

ShiftEmployeeID INT PRIMARY KEY AUTO\_INCREMENT,

ShiftID INT,

EmployeeID INT,

FOREIGN KEY (ShiftID) REFERENCES Shifts(ShiftID),

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

TableNumber INT NOT NULL,

CustomerCount INT NOT NULL,

OrderStatus VARCHAR(20) NOT NULL,

ShiftID INT,

FOREIGN KEY (ShiftID) REFERENCES Shifts(ShiftID)

);

select\*from Dishes;

CREATE TABLE Dishes (

DishID INT PRIMARY KEY AUTO\_INCREMENT,

DishName VARCHAR(50) NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

Photo longblob

);

CREATE TABLE OrderDishes (

OrderDishID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

DishID INT,

Quantity INT NOT NULL,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (DishID) REFERENCES Dishes(DishID)

);

-- Заполнение таблицы Users

INSERT INTO Users (Username, Password, Role) VALUES

('admin', 'admin123', 'Администратор'),

('waiter1', 'waiterpass', 'Официант'),

('waiter2', 'waiterpass', 'Официант'),

('cook1', 'cookpass', 'Повар'),

('cook2', 'cookpass', 'Повар'),

('manager', 'managerpass', 'Администратор');

-- Заполнение таблицы Employees

INSERT INTO Employees (FirstName, LastName, Status, UserID) VALUES

('Иван', 'Иванов', 'Работает', 1),

('Петр', 'Петров', 'Работает', 2),

('Анна', 'Сидорова', 'Работает', 3),

('Екатерина', 'Кузнецова', 'Работает', 4),

('Алексей', 'Алексеев', 'Работает', 5),

('Ольга', 'Смирнова', 'Уволен', 6);

-- Заполнение таблицы Shifts

INSERT INTO Shifts (ShiftType, StartTime, EndTime) VALUES

('Утренняя смена', '2024-02-26 08:00:00', '2024-02-26 16:00:00'),

('Вечерняя смена', '2024-02-26 16:00:00', '2024-02-26 23:00:00'),

('Ночная смена', '2024-02-27 23:00:00', '2024-02-27 08:00:00');

-- Заполнение таблицы ShiftEmployees

INSERT INTO ShiftEmployees (ShiftID, EmployeeID) VALUES

(1, 1),

(1, 2),

(2, 3),

(2, 4),

(3, 5),

(3, 6);

-- Заполнение таблицы Orders

INSERT INTO Orders (TableNumber, CustomerCount, OrderStatus, ShiftID) VALUES

(1, 4, 'В обработке', 1),

(2, 2, 'Оплачен', 1),

(3, 3, 'В обработке', 2),

(4, 5, 'Ожидание оплаты', 2),

(5, 2, 'Готов', 3);

-- Заполнение таблицы Dishes

INSERT INTO Dishes (DishName, Price) VALUES

('Борщ', 150.00),

('Пельмени', 120.00),

('Салат "Цезарь"', 180.00),

('Пицца Маргарита', 220.00),

('Стейк "Медальоны"', 320.00),

('Чай черный', 50.00),

('Кофе американо', 80.00),

('Сок апельсиновый', 70.00),

('Десерт "Тирамису"', 150.00),

('Лимонад', 60.00);

-- Заполнение таблицы OrderDishes

INSERT INTO OrderDishes (OrderID, DishID, Quantity) VALUES

(1, 1, 2),

(1, 3, 1),

(2, 4, 1),

(3, 2, 3),

(4, 5, 1),

(5, 6, 2),

(5, 7, 1),

(5, 8, 3);

Подключение

Scaffold-DbContext "Server=localhost;User=root;Password=Ivan2004;Database=cafemanagement" "Pomelo.EntityFrameworkCore.MySql" -OutputDir "Model"

Создаём папку Classes

Класс AllData

public static int ID;

==========================================================

В моделс

CafeManagementContext.cs

public static CafeManagementContext \_context;

public CafeManagementContext()

{

}

public CafeManagementContext(DbContextOptions<CafeManagementContext> options)

: base(options)

{

}

public static CafeManagementContext GetContext()

{

if (\_context == null) \_context = new CafeManagementContext();

return \_context;

}

==============================================================================

В моделс

User

public string FIO

{

get

{

Employee emp = App.context.Employees.ToList().Find(u => u.UserId == UserId);

return $"{emp.FirstName} {emp.LastName}";

}

}

public string Id

{

get

{

Employee emp = App.context.Employees.ToList().Find(u => u.UserId == UserId);

return $"{emp.EmployeeId}";

}

}

public string dolgnost

{

get

{

return $"{Role}";

}

}

====================================================================================

В App.xam.cs

public static CafeManagementContext context { get; } = new CafeManagementContext();

=================================================

Создаём папку виндовс

Окно AddEmployeee.xaml

XAML

Title="Добавить сотрудника" Height="361" Width="600" FontSize="20">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Background="#FFE2E2" Width="580">

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Фамилия</TextBlock>

<TextBox x:Name="Surname" Width="300" Text="{Binding FirstName}" />

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Имя</TextBlock>

<TextBox x:Name="Name" Width="300" Text="{Binding LastName}"></TextBox>

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Роль</TextBlock>

<ComboBox x:Name="Role" Width="300" Text="{Binding User.Role}">

<ComboBoxItem>Администратор</ComboBoxItem>

<ComboBoxItem>Повар</ComboBoxItem>

<ComboBoxItem>Официант</ComboBoxItem>

</ComboBox>

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Логин</TextBlock>

<TextBox x:Name="Username" Width="300" Text="{Binding User.Username}"></TextBox>

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Пароль</TextBlock>

<TextBox x:Name="Password" Width="300" Text="{Binding User.Password}"></TextBox>

</StackPanel>

<Label/>

<Button x:Name="Add" Click="Add\_Click" Background="#FCBAD3" Width="534">Сохранить</Button>

<Label/>

<Button x:Name="Back" Click="Back\_Click" Background="#FCBAD3" Width="537">Вернуться</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class AddEmployeee : System.Windows.Window

{

public AddEmployeee()

{

InitializeComponent();

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы уверены, что хотите выйти? Новые данные не сохранятся.", "Подтверждение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

UsersAdmin usersAdmin = new UsersAdmin();

usersAdmin.Show();

this.Close();

}

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

User newUser = new User

{

Role = Role.Text,

Username = Username.Text,

Password = Password.Text,

};

CafeManagementContext context = CafeManagementContext.GetContext();

context.Users.Add(newUser);

context.SaveChanges();

Employee newEmployee = new Employee

{

FirstName = Surname.Text,

LastName = Name.Text,

Status = "Работает",

UserId = newUser.UserId,

};

context.Employees.Add(newEmployee);

context.SaveChanges();

MessageBox.Show("Успешно добавлено");

UsersAdmin usersAdmin = new UsersAdmin();

usersAdmin.Show();

this.Close();

}

}

}

========================================================================================================

Окно AddOrder.xaml

XAML

Title="Создание заказа" Height="717" Width="600" FontSize="20">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Background="#FFE2E2" Width="580" Height="676">

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Номер столика</TextBlock>

<TextBox x:Name="Number" Width="300" Text="{Binding TableNumber}" />

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Количество клиентов</TextBlock>

<TextBox x:Name="QuantityClient" Width="300" Text="{Binding CustomerCount}"></TextBox>

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230">Блюдо</TextBlock>

<ComboBox x:Name="Diches" Width="300" SelectedValuePath="DishId">

<ComboBox.ItemTemplate>

<DataTemplate>

<TextBlock>

<Run Text="{Binding DishName}" />

<Run Text=" " />

<Run Text="{Binding Price}" />

</TextBlock>

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox>

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="230" Text="{Binding Quantity}">Количество</TextBlock>

<TextBox x:Name="Quantity" Width="300">

</TextBox>

</StackPanel>

<Label/>

<Button x:Name="AddDish" Background="#FCBAD3" Click="AddDish\_Click\_1" >Добавить блюдо в заказ</Button>

<Label/>

<StackPanel Orientation="Vertical" Height="381">

<TextBlock Width="574" TextAlignment="Center">Выбранные блюда:</TextBlock>

<Label x:Name="EmployeeId" Height="8"/>

<DataGrid x:Name="ordersListView" Height="338" Width="588" Background="{x:Null}" AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="№" Binding="{Binding Номер}" Width="80"/>

<DataGridTextColumn Header="Блюдо" Binding="{Binding Блюдо}" Width="200"/>

<DataGridTextColumn Header="Количество" Binding="{Binding Количество}" Width="130"/>

<DataGridTextColumn Header="Цена" Binding="{Binding Цена}" Width="130"/>

</DataGrid.Columns>

</DataGrid>

</StackPanel>

<Button x:Name="Add" Click="Add\_Click" Background="#FCBAD3" Width="534">Сохранить</Button>

<Label/>

<Button x:Name="Back" Click="Back\_Click" Background="#FCBAD3" Width="537">Вернуться</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class AddOrder : System.Windows.Window

{

private int dishCounter = 1;

private int? orderNumber;

public AddOrder()

{

InitializeComponent();

List<Dish> participantsList = App.context.Dishes.ToList();

Diches.ItemsSource = participantsList;

orderNumber = null;

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

OrdersWaiter ordersWaiter = new OrdersWaiter();

ordersWaiter.Show();

this.Close();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

MessageBox.Show("Заказ успешно создан!", "Уведомление", MessageBoxButton.OK);

OrdersWaiter ordersWaiter = new OrdersWaiter();

ordersWaiter.Show();

this.Close();

}

private void AddDish\_Click\_1(object sender, RoutedEventArgs e)

{

if (string.IsNullOrWhiteSpace(Diches.Text) || string.IsNullOrWhiteSpace(Quantity.Text) || string.IsNullOrWhiteSpace(Number.Text) || string.IsNullOrWhiteSpace(QuantityClient.Text))

{

MessageBox.Show("Пожалуйста, заполните все поля перед сохранением.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

if (orderNumber == null)

{

Order newOrder = new()

{

OrderStatus = "Новый",

ShiftId = 3,

TableNumber = int.Parse(QuantityClient.Text),

CustomerCount = int.Parse(QuantityClient.Text)

};

App.context.Orders.Add(newOrder);

App.context.SaveChanges();

orderNumber = newOrder.OrderId;

QuantityClient.IsEnabled = false;

Number.IsEnabled = false;

}

Dish selectedDish = Diches.SelectedItem as Dish;

if (selectedDish != null)

{

if (App.context.Orderdishes.Any(od => od.OrderId == orderNumber && od.DishId == selectedDish.DishId))

{

MessageBox.Show("Это блюдо уже добавлено в заказ.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

Orderdish orderDish = new()

{

OrderId = orderNumber,

DishId = selectedDish.DishId,

Quantity = int.Parse(Quantity.Text)

};

App.context.Orderdishes.Add(orderDish);

App.context.SaveChanges();

ordersListView.Items.Add(new

{

Номер = dishCounter,

Блюдо = selectedDish.DishName,

Количество = orderDish.Quantity,

Цена = selectedDish.Price

});

dishCounter++;

Diches.SelectedIndex = -1;

Quantity.Clear();

}

}

}

}

=================================================================================================================

Окно Authorization.xaml

XAML

Title="Авторизация" Height="498" Width="384" FontSize="20" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="338\*"/>

<ColumnDefinition Width="31\*"/>

</Grid.ColumnDefinitions>

<StackPanel HorizontalAlignment="Left" VerticalAlignment="Center" Background="#FFE2E2" Grid.ColumnSpan="2" Margin="15,0,0,0">

<Image Height="203" Width="194" Source="/Images/free-icon-cafe-1223041.png" />

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="80" >Логин:</TextBlock>

<Label/>

<TextBox x:Name="Login" Width="250" />

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="80">Пароль:</TextBlock>

<Label/>

<PasswordBox x:Name="Password" Visibility="Collapsed" Width="250" />

<TextBox x:Name="Password2" Visibility="Collapsed" Width="250" TextChanged="Password2\_TextChanged"/>

</StackPanel>

<Label/>

<StackPanel Orientation="Horizontal">

<Label Width="76"/>

<CheckBox x:Name="Safety" Checked="Safety\_Checked" Unchecked="Safety\_Unchecked">Показать/Скрыть пароль</CheckBox>

</StackPanel>

<Label/>

<Button x:Name="Enter" Background="#FCBAD3" Click="Enter\_Click">Войти</Button>

<Label/>

<Button x:Name="Exit" Background="#FCBAD3" Click="Exit\_Click">Выход</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class Authorization : System.Windows.Window

{

public Authorization()

{

InitializeComponent();

Password.PasswordChanged += Password\_PasswordChanged;

Password.Visibility = Visibility.Visible;

Password.Password = Password2.Text;

}

private void Password\_PasswordChanged(object sender, RoutedEventArgs e)

{

Password2.Text = Password.Password;

}

private void GuestEnter\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void Password2\_TextChanged(object sender, TextChangedEventArgs e)

{

Password.Password = Password2.Text;

}

private void Safety\_Checked(object sender, RoutedEventArgs e)

{

Password2.Visibility = Visibility.Visible;

Password.Visibility = Visibility.Collapsed;

Password2.Text = Password.Password;

}

private void Safety\_Unchecked(object sender, RoutedEventArgs e)

{

Password2.Visibility = Visibility.Collapsed;

Password.Visibility = Visibility.Visible;

Password.Password = Password2.Text;

}

private void Enter\_Click(object sender, RoutedEventArgs e)

{

if (Login.Text.Length != 0 && Password.Password.Length != 0 && Password2.Text.Length != 0)

{

User user = App.context.Users.ToList().Find(u => u.Username == Login.Text && u.Password == Password.Password && u.Password == Password2.Text);

if (user != null)

{

if (user.Role == "Администратор")

{

AllData.ID = user.UserId;

MessageBox.Show($"Добро пожаловать, {user.dolgnost} {user.FIO} !");

MenuAdmin adminMenu = new MenuAdmin();

adminMenu.Show();

this.Close();

return;

}

else if (user.Role == "Повар")

{

AllData.ID = user.UserId;

MessageBox.Show($"Добро пожаловать, {user.dolgnost} {user.FIO} !");

MenuCook cookMenu = new MenuCook();

cookMenu.Show();

this.Close();

return;

}

else if (user.Role == "Официант")

{

AllData.ID = user.UserId;

MessageBox.Show($"Добро пожаловать, {user.dolgnost} {user.FIO} !");

MenuWaiter waiterMenu = new MenuWaiter();

waiterMenu.Show();

this.Close();

return;

}

}

else

{

MessageBox.Show("Вы ввели неверные данные!");

}

}

else

{

MessageBox.Show("Вы должны заполнить все поля!");

}

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

===========================================================================================================================

Окно MenuAdmin.xaml

XAML

Title="Меню администартора" Height="515" Width="521" FontSize="20" WindowStartupLocation="CenterScreen" Background="#FFE2E2">

<Grid>

<StackPanel Margin="15,0,12,9" Grid.RowSpan="2">

<StackPanel Orientation="Horizontal" Width="448" >

<Label/>

<Image Source="/images/free-icon-cafe-1223041.png" Height="60" Width="60"/>

<Label Content="Меню администратора" Margin="14,11,0,0" VerticalAlignment="Top" FontSize="32" Width="365"/>

</StackPanel>

<Label Height="37"/>

<StackPanel Width="450" Height="368" >

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<Button x:Name="Users" Background="#FCBAD3" Click="Users\_Click" Height="45" Width="434">Сотрудники</Button>

<Label/>

<Button x:Name="Shifts" Background="#FCBAD3" Click="Shifts\_Click" Height="45" Width="437">Смены</Button>

<Label/>

<Button x:Name="Orders" Background="#FCBAD3" Click="Orders\_Click" Height="45" Width="439">Заказы</Button>

<Label Height="61"/>

<Button x:Name="Back" Background="#FCBAD3" Click="Back\_Click" Height="31" Width="439">Вернуться</Button>

<Label/>

<Button x:Name="Exit" Background="#FCBAD3" Click="Exit\_Click" Height="32" Width="441">Выход</Button>

<Label/>

</StackPanel>

</StackPanel>

</Grid>

</Window>

КОД

public partial class MenuAdmin : System.Windows.Window

{

public MenuAdmin()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

}

private void Users\_Click(object sender, RoutedEventArgs e)

{

UsersAdmin usersAdmin = new UsersAdmin();

usersAdmin.Show();

this.Close();

}

private void Shifts\_Click(object sender, RoutedEventArgs e)

{

ShiftsAdmin shiftsAdmin = new ShiftsAdmin();

shiftsAdmin.Show();

this.Close();

}

private void Orders\_Click(object sender, RoutedEventArgs e)

{

OrdersMenu ordersMenu = new OrdersMenu();

ordersMenu.Show();

this.Close();

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

Authorization authorization = new Authorization();

authorization.Show();

this.Close();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

=========================================================================================================================================================

Окно MenuCook.xaml

XAML

Title="Меню повара" Height="515" Width="521" FontSize="20" WindowStartupLocation="CenterScreen" Background="#FFE2E2">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="217\*"/>

<RowDefinition Height="230\*"/>

</Grid.RowDefinitions>

<StackPanel Margin="15,0,12,9" Grid.RowSpan="2">

<StackPanel Orientation="Horizontal" Width="326" >

<Label/>

<Image Source="/images/free-icon-cafe-1223041.png" Height="60" Width="60"/>

<Label Content="Меню повара" Margin="14,11,0,0" VerticalAlignment="Top" FontSize="32" Width="365"/>

</StackPanel>

<Label Height="37"/>

<StackPanel Width="450" Height="368" >

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<Button x:Name="Orders" Background="#FCBAD3" Click="Orders\_Click" Height="45" Width="434">Заказы</Button>

<Label Height="167"/>

<Button x:Name="Back" Background="#FCBAD3" Click="Back\_Click" Height="31" Width="439">Вернуться</Button>

<Label/>

<Button x:Name="Exit" Background="#FCBAD3" Click="Exit\_Click" Height="32" Width="441">Выход</Button>

<Label/>

</StackPanel>

</StackPanel>

</Grid>

</Window>

КОД

public partial class MenuCook : System.Windows.Window

{

public MenuCook()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

Authorization authorization = new Authorization();

authorization.Show();

this.Close();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void Orders\_Click(object sender, RoutedEventArgs e)

{

OrdersCook ordersCook = new OrdersCook();

ordersCook.Show();

this.Close();

}

}

}

=====================================================================================================================================================

Окно MenuWaiter.xaml

XAML

Title="Меню администартора" Height="515" Width="521" FontSize="20" WindowStartupLocation="CenterScreen" Background="#FFE2E2">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="217\*"/>

<RowDefinition Height="230\*"/>

</Grid.RowDefinitions>

<StackPanel Margin="15,0,12,9" Grid.RowSpan="2">

<StackPanel Orientation="Horizontal" Width="377" >

<Label/>

<Image Source="/images/free-icon-cafe-1223041.png" Height="60" Width="60"/>

<Label Content="Меню официанта" Margin="14,11,0,0" VerticalAlignment="Top" FontSize="32" Width="365"/>

</StackPanel>

<Label Height="37"/>

<StackPanel Width="450" Height="368" >

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<Button x:Name="Orders" Background="#FCBAD3" Click="Orders\_Click" Height="45" Width="439">Заказы</Button>

<Label Height="164"/>

<Button x:Name="Back" Background="#FCBAD3" Click="Back\_Click" Height="31" Width="439">Вернуться</Button>

<Label/>

<Button x:Name="Exit" Background="#FCBAD3" Click="Exit\_Click" Height="32" Width="441">Выход</Button>

<Label/>

</StackPanel>

</StackPanel>

</Grid>

</Window>

КОД

public partial class MenuWaiter : System.Windows.Window

{

public MenuWaiter()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

Authorization authorization = new Authorization();

authorization.Show();

this.Close();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void Orders\_Click(object sender, RoutedEventArgs e)

{

OrdersWaiter ordersWaiter = new OrdersWaiter();

ordersWaiter.Show();

this.Close();

}

}

}

=================================================================================================================================================================

Окно OrdersCook.xaml

XAML

Title="Заказы" MinHeight="300" MinWidth="700" FontSize="20" Background="#FFE2E2" WindowStartupLocation="CenterScreen" Height="926" Width="1003" DataContext="{Binding RelativeSource={RelativeSource Self}}">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Height="883" Width="884">

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<ListView x:Name="ordersListView" Height="663" AlternationCount="2" MouseUp="ordersListView\_MouseUp">

<ListView.View>

<GridView>

<GridViewColumn Header="Номер столика" DisplayMemberBinding="{Binding TableNumber}" />

<GridViewColumn Header="Количество клиентов" DisplayMemberBinding="{Binding CustomerCount}" />

<GridViewColumn Header="Статус заказа" DisplayMemberBinding="{Binding OrderStatus}" />

<GridViewColumn Header="Блюда в заказе">

<GridViewColumn.CellTemplate>

<DataTemplate>

<ItemsControl ItemsSource="{Binding DishList}">

<ItemsControl.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding DishName}" Margin="5"/>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

<Label Height="14"/>

<Button Content="Заказ готов" Background="#FCBAD3" Height="51" Click="MarkOrderAsReady\_Click"/>

<Label Height="33"/>

<Button x:Name="back" Background="#FCBAD3" Click="back\_Click" >Вернуться</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class OrdersCook : System.Windows.Window

{

public OrdersCook()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

LoadOrders();

}

private void LoadOrders()

{

using (var dbContext = new CafeManagementContext())

{

var orders = dbContext.Orders

.Include(o => o.Orderdishes)

.ThenInclude(od => od.Dish)

.ToList();

var orderData = orders.Select(order => new

{

order.OrderId,

order.TableNumber,

order.CustomerCount,

order.OrderStatus,

DishList = order.Orderdishes.Select(od => new

{

DishName = od.Dish?.DishName

}).ToList()

}).ToList();

ordersListView.ItemsSource = orderData;

}

}

private void MarkOrderAsReady\_Click(object sender, RoutedEventArgs e)

{

if (ordersListView.SelectedItem != null)

{

var selectedOrder = (dynamic)ordersListView.SelectedItem;

int orderId = selectedOrder.OrderId; // Используем OrderId вместо TableNumber

UpdateOrderStatus(orderId, "Готов");

LoadOrders();

}

}

private void UpdateOrderStatus(int tableNumber, string newStatus)

{

using (var dbContext = new CafeManagementContext())

{

var orderToUpdate = dbContext.Orders.FirstOrDefault(o => o.OrderId == tableNumber);

if (orderToUpdate != null)

{

orderToUpdate.OrderStatus = newStatus;

dbContext.SaveChanges();

}

}

}

private void ordersListView\_MouseUp(object sender, MouseButtonEventArgs e)

{

if (ordersListView.SelectedItem != null)

{

var selectedOrder = (dynamic)ordersListView.SelectedItem;

int orderId = selectedOrder.OrderId; // Используем OrderId вместо TableNumber

MessageBox.Show($"Выбран заказ с номером {orderId}!");

}

}

private void back\_Click(object sender, RoutedEventArgs e)

{

MenuCook menuCook = new MenuCook();

menuCook.Show();

this.Close();

}

}

}

======================================================================================================================

Окно OrdersMenu.xaml

XAML

Title="Заказы" MinHeight="300" MinWidth="700" FontSize="20" Background="#FFE2E2" WindowStartupLocation="CenterScreen" Height="926" Width="1003">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Height="883" Width="884">

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<ListView x:Name="ordersListView" Height="745" AlternationCount="2">

<ListView.View>

<GridView>

<GridViewColumn Header="Номер столика" DisplayMemberBinding="{Binding TableNumber}" />

<GridViewColumn Header="Количество клиентов" DisplayMemberBinding="{Binding CustomerCount}" />

<GridViewColumn Header="Статус заказа" DisplayMemberBinding="{Binding OrderStatus}" />

<GridViewColumn Header="Блюда в заказе">

<GridViewColumn.CellTemplate>

<DataTemplate>

<ItemsControl ItemsSource="{Binding DishList}">

<ItemsControl.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding DishName}" Margin="5"/>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

<Label Height="17"/>

<Button x:Name="back" Background="#FCBAD3" Click="back\_Click">Вернуться</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class OrdersMenu : System.Windows.Window

{

public OrdersMenu()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

LoadOrders();

}

private void LoadOrders()

{

using (var dbContext = new CafeManagementContext())

{

var orders = dbContext.Orders

.Include(o => o.Orderdishes)

.ThenInclude(od => od.Dish)

.ToList();

var orderData = orders.Select(order => new

{

order.TableNumber,

order.CustomerCount,

order.OrderStatus,

DishList = order.Orderdishes.Select(od => new

{

DishName = od.Dish?.DishName

}).ToList()

}).ToList();

ordersListView.ItemsSource = orderData;

}

}

private void back\_Click(object sender, RoutedEventArgs e)

{

MenuAdmin menuAdmin = new MenuAdmin();

menuAdmin.Show();

this.Close();

}

}

}

=====================================================================================================================

Окно OrdersWaiter.xaml

XAML

Title="Заказы" MinHeight="300" MinWidth="700" FontSize="20" Background="#FFE2E2" WindowStartupLocation="CenterScreen" Height="926" Width="1003" DataContext="{Binding RelativeSource={RelativeSource Self}}">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Height="883" Width="884">

<StackPanel>

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<Button x:Name="AddOrder" Height="48" Width="246" HorizontalAlignment="Left" Background="#FCBAD3" Click="AddOrder\_Click">+ Новый заказ</Button>

</StackPanel>

<Label/>

<ListView x:Name="ordersListView" Height="615" AlternationCount="2" MouseUp="ordersListView\_MouseUp">

<ListView.View>

<GridView>

<GridViewColumn Header="Номер столика" DisplayMemberBinding="{Binding TableNumber}" />

<GridViewColumn Header="Количество клиентов" DisplayMemberBinding="{Binding CustomerCount}" />

<GridViewColumn Header="Статус заказа" DisplayMemberBinding="{Binding OrderStatus}" />

<GridViewColumn Header="Блюда в заказе">

<GridViewColumn.CellTemplate>

<DataTemplate>

<ItemsControl ItemsSource="{Binding DishList}">

<ItemsControl.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding DishName}" Margin="5"/>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

<Label Height="14"/>

<Button Content="Заказ оплачен" Background="#FCBAD3" Height="51" Click="MarkOrderAsReady\_Click"/>

<Label Height="33"/>

<Button x:Name="back" Background="#FCBAD3" Click="back\_Click" >Вернуться</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class OrdersWaiter : System.Windows.Window

{

public OrdersWaiter()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

LoadOrders();

}

private void LoadOrders()

{

using (var dbContext = new CafeManagementContext())

{

var orders = dbContext.Orders

.Include(o => o.Orderdishes)

.ThenInclude(od => od.Dish)

.ToList();

var orderData = orders.Select(order => new

{

order.OrderId,

order.TableNumber,

order.CustomerCount,

order.OrderStatus,

DishList = order.Orderdishes.Select(od => new

{

DishName = od.Dish?.DishName

}).ToList()

}).ToList();

ordersListView.ItemsSource = orderData;

}

}

private void MarkOrderAsReady\_Click(object sender, RoutedEventArgs e)

{

if (ordersListView.SelectedItem != null)

{

var selectedOrder = (dynamic)ordersListView.SelectedItem;

int orderId = selectedOrder.OrderId; // Используем OrderId вместо TableNumber

UpdateOrderStatus(orderId, "Оплачен");

LoadOrders();

}

}

private void UpdateOrderStatus(int tableNumber, string newStatus)

{

using (var dbContext = new CafeManagementContext())

{

var orderToUpdate = dbContext.Orders.FirstOrDefault(o => o.OrderId == tableNumber);

if (orderToUpdate != null)

{

orderToUpdate.OrderStatus = newStatus;

dbContext.SaveChanges();

}

}

}

private void ordersListView\_MouseUp(object sender, MouseButtonEventArgs e)

{

if (ordersListView.SelectedItem != null)

{

var selectedOrder = (dynamic)ordersListView.SelectedItem;

int orderId = selectedOrder.OrderId; // Используем OrderId вместо TableNumber

MessageBox.Show($"Выбран заказ с номером {orderId}!");

}

}

private void back\_Click(object sender, RoutedEventArgs e)

{

MenuCook menuCook = new MenuCook();

menuCook.Show();

this.Close();

}

private void AddOrder\_Click(object sender, RoutedEventArgs e)

{

int newOrderNumber = GetNewOrderNumber();

AddOrder addorder = new AddOrder();

addorder.Show();

this.Close();

}

private int GetNewOrderNumber()

{

return 1;

}

}

}

=============================================================================================================

Окно ShiftsAdmin.xaml

XAML

Title="Cмены" MinHeight="300" MinWidth="700" FontSize="20" Background="#FFE2E2" WindowStartupLocation="CenterScreen" Height="926" Width="1003">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Height="883" Width="884">

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<StackPanel/>

<Label Height="34"/>

<DataGrid x:Name="shiftsDataGrid" CanUserAddRows="False" AutoGenerateColumns="False" SelectionMode="Single" Height="241" Width="690" SelectionChanged="shiftsDataGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Тип смены" Binding="{Binding ShiftType}" />

<DataGridTextColumn Header="Начало" Binding="{Binding StartTime}" />

<DataGridTextColumn Header="Окончание" Binding="{Binding EndTime}" />

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Horizontal" Width="685">

<Label Width="120">Смена:</Label>

<TextBlock x:Name="Shifts1" Width="563"/>

</StackPanel>

<Label/>

<DataGrid x:Name="employeesDataGrid" CanUserAddRows="False" AutoGenerateColumns="False" SelectionMode="Single" Height="289" SelectionChanged="employeesDataGrid\_SelectionChanged" Width="687">

<DataGrid.Columns>

<DataGridTextColumn Header="Имя" Binding="{Binding FirstName}" />

<DataGridTextColumn Header="Фамилия" Binding="{Binding LastName}" />

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Horizontal" Width="684">

<Label Width="132">Сотрудник:</Label>

<TextBlock x:Name="Emp" Width="214"/>

</StackPanel>

<Label/>

<Button Content="Назначить" Background="#FCBAD3" Click="AssignButton\_Click"/>

<Label Height="17"/>

<Button x:Name="back" Background="#FCBAD3" Click="back\_Click">Вернуться</Button>

</StackPanel>

</Grid>

</Window>

КОД

public partial class ShiftsAdmin : System.Windows.Window

{

// Используйте List вместо ObservableCollection для простоты примера

public List<Shift> Shifts { get; set; } = new List<Shift>();

public List<Employee> Employees { get; set; } = new List<Employee>();

// Выбранные элементы

private Shift selectedShift;

private Employee selectedEmployee;

public ShiftsAdmin()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

shiftsDataGrid.ItemsSource = Shifts;

employeesDataGrid.ItemsSource = Employees;

LoadShifts();

LoadEmployees();

}

private void AssignButton\_Click(object sender, RoutedEventArgs e)

{

if (selectedShift == null)

{

MessageBox.Show("Выберите смену для назначения.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

if (selectedEmployee == null)

{

MessageBox.Show("Выберите сотрудника для назначения.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

// Проверка, что сотрудник не назначен на выбранную смену

if (IsEmployeeAssigned(selectedShift.ShiftId, selectedEmployee.EmployeeId))

{

MessageBox.Show($"Сотрудник {selectedEmployee.FirstName} {selectedEmployee.LastName} уже назначен на выбранную смену.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

Shiftemployee newAssignment = new Shiftemployee

{

ShiftId = selectedShift.ShiftId,

EmployeeId = selectedEmployee.EmployeeId

};

SaveAssignment(newAssignment);

LoadShifts();

LoadEmployees();

MessageBox.Show($"Сотрудник {selectedEmployee.FirstName} {selectedEmployee.LastName} успешно назначен на смену '{selectedShift.ShiftType}' с {selectedShift.StartTime} до {selectedShift.EndTime}.", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

}

private bool IsEmployeeAssigned(int shiftId, int employeeId)

{

using (var dbContext = new CafeManagementContext())

{

return dbContext.Shiftemployees.Any(se => se.ShiftId == shiftId && se.EmployeeId == employeeId);

}

}

private void LoadShifts()

{

using (var dbContext = new CafeManagementContext())

{

Shifts.Clear();

var shifts = dbContext.Shifts.ToList();

foreach (var shift in shifts)

{

Shifts.Add(shift);

}

}

}

private void LoadEmployees()

{

using (var dbContext = new CafeManagementContext())

{

var workingEmployees = dbContext.Employees.Where(employee => employee.Status == "Работает").ToList();

Employees.Clear();

foreach (var employee in workingEmployees)

{

Employees.Add(employee);

}

}

}

private void shiftsDataGrid\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

selectedShift = shiftsDataGrid.SelectedItem as Shift;

Shifts1.Text = selectedShift != null ? $"{selectedShift.ShiftType} {selectedShift.StartTime} {selectedShift.EndTime}" : "";

}

private void employeesDataGrid\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

selectedEmployee = employeesDataGrid.SelectedItem as Employee;

Emp.Text = selectedEmployee != null ? $"{selectedEmployee.FirstName} {selectedEmployee.LastName}" : "";

}

private void SaveAssignment(Shiftemployee assignment)

{

using (var dbContext = new CafeManagementContext())

{

dbContext.Shiftemployees.Add(assignment);

dbContext.SaveChanges();

}

}

private void back\_Click(object sender, RoutedEventArgs e)

{

MenuAdmin menuAdmin = new MenuAdmin();

menuAdmin.Show();

this.Close();

}

}

}

========================================================================================================================================================

Окно UsersAdmin.xaml

XAML

Title="Сотрудники" MinHeight="300" MinWidth="700" FontSize="20" Background="#FFE2E2" WindowStartupLocation="CenterScreen" Height="810" Width="1229">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Height="751" Width="1214">

<Label x:Name="dolj" Content="Вы вошли как " HorizontalAlignment="Right"/>

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<StackPanel>

<Button x:Name="Add" Background="#FCBAD3" Click="Add\_Click" Width="301" >+ Добавить нового сотрудника</Button>

</StackPanel>

<StackPanel Orientation="Horizontal" Height="23">

<Label x:Name="RecordsCountLabel"/>

<Label x:Name="FilteredRecordsCountLabel"/>

</StackPanel>

<Label/>

<DataGrid x:Name="usersView" AutoGenerateColumns="False" Width="1084" Height="523" CanUserAddRows="False" MouseDoubleClick="usersView\_MouseDoubleClick">

<DataGrid.Columns>

<DataGridTextColumn Header="Номер сотрудника" Binding="{Binding EmployeeId}" />

<DataGridTextColumn Header="Фамилия" Binding="{Binding FirstName}" />

<DataGridTextColumn Header="Имя" Binding="{Binding LastName}" />

<DataGridTextColumn Header="Роль" Binding="{Binding User.Role}" />

<DataGridTextColumn Header="Статус" Binding="{Binding Status}" />

</DataGrid.Columns>

</DataGrid>

<Label Height="17"/>

<Button x:Name="back" Background="#FCBAD3" Click="back\_Click">Вернуться</Button>

</StackPanel>

</Grid>

</Window>

public partial class UsersAdmin : System.Windows.Window

{

private CafeManagementContext dbContext = new CafeManagementContext();

public List<Employee> Employees { get; set; } = new List<Employee>();

public UsersAdmin()

{

InitializeComponent();

if (AllData.ID != 0)

{

User user = App.context.Users.ToList().Find(u => u.UserId == AllData.ID);

name.Content = user.FIO;

dolj.Content = user.dolgnost;

}

Loadata();

}

public void Loadata()

{

List<Employee> tabularsections = dbContext.Employees.Include(t => t.User).ToList();

usersView.ItemsSource = tabularsections;

}

private void back\_Click(object sender, RoutedEventArgs e)

{

MenuAdmin menuAdmin = new MenuAdmin();

menuAdmin.Show();

this.Close();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

AddEmployeee addEmployeee = new AddEmployeee();

addEmployeee.Show();

this.Close();

}

private void usersView\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

if (usersView.SelectedItem != null)

{

Employee selectedEmployee = (Employee)usersView.SelectedItem;

// Отображение диалогового окна с вопросом

MessageBoxResult result = MessageBox.Show($"Вы действительно хотите сменить статус сотрудника {selectedEmployee.FirstName} {selectedEmployee.LastName} на 'Уволен'?", "Подтверждение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

// Если пользователь выбрал 'Да', меняем статус

selectedEmployee.Status = "Уволен";

dbContext.SaveChanges();

Loadata();

MessageBox.Show($"Статус сотрудника {selectedEmployee.FirstName} {selectedEmployee.LastName} изменен на 'Уволен'.", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

}

}