|  |
| --- |
| 3 |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»** |

**Колледж программирования и кибербезопасности**

**ОТЧЁТ**

**по учебной практике УП.02.01**

по профессиональному модулю

**ПМ.02 Осуществление интеграции программных модулей**

Специальность 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| **Студента** | Гулевский Илья Вадимович |
| фамилия, имя, отчество | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Группа** | ЩПКО-04-20 | | | **курс** | 4 | | **отделение** | | дневное |
|  | | | | | | | | | |
| **Начало практики** | | | 27.10.2023 | | | **окончание** | | 16.11.2023 | |
|  | | | | | | | | | |
| **Объект практики** | | Колледж программирования | | | | | | | |
| и кибербезопасности | | | | | | | | | |
|  | | | | | | | | | |
| **Руководитель практики** | | | | | Е.А. Траханович | | | | |
|  | | | | | фамилия, инициалы | | | | |

Москва 2023

СОДЕРЖАНИЕ

[1. Модуль А 3](#_Toc150985672)

[1.1 Инструктаж по охране труда и технике безопасности 3](#_Toc150985673)

[1.2 Анализ технического задания 3](#_Toc150985674)

[1.3 Спецификация программного модуля 3](#_Toc150985675)

[1.4 Входные и выходные данные 5](#_Toc150985676)

[1.5 Алгоритмы 6](#_Toc150985677)

[1.6 Диаграмма активностей 8](#_Toc150985678)

[1.7 Интерфейс 8](#_Toc150985679)

[1.7.1 Дизайн программного модуля 9](#_Toc150985680)

[1.7.2 Дизайн мобильной версии 12](#_Toc150985681)

[1.8 Реализация 13](#_Toc150985682)

[1.9 Отладка модуля 24](#_Toc150985683)

[1.10 Результаты тестирования 26](#_Toc150985684)

[2. Модуль B 27](#_Toc150985685)

[2.1 ER-диаграмма 27](#_Toc150985686)

[2.2 Создание БД 27](#_Toc150985687)

[2.2.1 Запросы к базе данных 27](#_Toc150985688)

[2.2.2 Резервное копирование БД 30](#_Toc150985689)

[2.2.3 Принцип регистрации 30](#_Toc150985690)

[2.2.4 Уровни доступа 32](#_Toc150985691)

[2.3 Руководство системному программисту 33](#_Toc150985692)

[2.4 Варианты модификации программного обеспечения 34](#_Toc150985693)

[3. Модуль C 36](#_Toc150985694)

[3.1 Новый пользователь 36](#_Toc150985695)

[3.2 Роль Менеджера 36](#_Toc150985696)

[3.3 Настройка и компоненты 36](#_Toc150985697)

[3.4 Демонстрация функционала согласно ТЗ 36](#_Toc150985698)

[ЗАКЛЮЧЕНИЕ 38](#_Toc150985699)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 39](#_Toc150985700)

### ВВЕДЕНИЕ

Студент группы ПКС-44 Гулевский Илья Вадимович проходил учебную практику в организации – Колледж программирования и кибербезопасности по адресу: г.Москва, 1-й Щипковский переулок, д 23.

Целью учебной практики является **осуществление интеграции программных модулей**.

Для достижения поставленной цели были определены следующие задачи:

* разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент,
* выполнять интеграцию модулей в программное обеспечение
* выполнять отладку программного модуля с использованием специализированных программных средств,
* осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения,
* производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

### 1. Модуль А

### 1.1 Инструктаж по охране труда и технике безопасности

В ходе работы был проведен полный инструктаж по технике безопасности.

### 1.2 Анализ технического задания

Получено и проанализировано задание на учебную практику. Сформулированы основные цели и задачи, которые необходимо достигнуть.

### 1.3 Спецификация программного модуля

Краткая спецификация разрабатываемого модуля:

Модульная спецификация

**1. 1 Разделы спецификации:**

Документационные аспекты

Комплексы и их описания

Компоненты и их характеристики

Документационное описание

**2. 1 Обозначение:**

Документы, относящиеся к программе.

**2. 2 Наименование:**

Наименование и форма документов, связанных с программой; полное наименование программы, а также указание наименования и формы для заимствованных документов.

**2. 3 Примечание:**

Дополнительные сведения, касающиеся записанных в спецификации программ.

Комплексные аспекты

**3. 1 Обозначение:**

Идентификация спецификаций комплексов, входящих в данный комплекс.

**3. 2 Наименование:**

Полное наименование программы, наименование и форма документа.

**3. 3 Примечание:**

Дополнительные сведения, относящиеся к записям в спецификациях программ.

Компоненты

**6. 1 Обозначение:**

Идентификация основных программных документов компонентов.

**6. 2 Наименование:**

Полное наименование программы, наименование и форма документа.

**6. 3 Примечание:**

Дополнительные сведения, относящиеся к записям в спецификациях программ.

Структура данных

**7. 1 Переменные:**

Перечисление переменных и их предназначение.

Алгоритмы

**8. 1 Получение данных:**

Использует запрос к базе данных для извлечения информации о заявках на ремонт оборудования.

**8. 2 Обработка заявок:**

Реализует логику обработки и учета заявок на ремонт.

Ограничения и предположения

**9. 1 Ограничения:**

Необходимо наличие соединения с базой данных для работы с заявками.

**9. 2 Предположения:**

Корректное определение пользовательских ролей в системе.

Тестирование

**10. 1 Тестирование отображения данных:**

Проверка правильности отображения информации о заявках.

**10. 2 Тестирование обработки заявок:**

Проверка правильности обработки и учета заявок.

### 1.4 Входные и выходные данные

**Входные данные :**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Тип данных** | **Описание** |
| 1 | Данные заявки | Структура/Класс | Информация о заявке на ремонт оборудования. |
| 2 | Номер заявки | Целое число | Уникальный идентификатор заявки. |
| 3 | Описание проблемы | Строка | Описание проблемы, указанной клиентом. |
| 4 | Дата начала ремонта | Дата и время | Дата и время, когда начат ремонт. |
| 5 | Дата окончания ремонта | Дата и время | Планируемая дата окончания ремонта. |
| 6 | Статус заявки | Перечисление | Текущий статус заявки (в работе, выполнено, не выполнено). |
| 7 | Информация о клиенте | Строка | Данные о клиенте, подавшем заявку. |
| 8 | Цена ремонта | Десятичное число | Стоимость ремонта оборудования. |
| 9 | Оборудование | Строка | Описание оборудования, требующего ремонта. |
| 10 | Тип проблемы | Строка | Категория или тип проблемы с оборудованием. |

**Выходные данные :**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Тип данных** | **Описание** |
| 1 | Результат операции | Булево значение | Успешное или неуспешное выполнение операции. |
| 2 | Информация о заявке | Структура/Класс | Информация о заявке после обновления или добавления. |
| 3 | Список заявок | Список структур | Список заявок, соответствующих определенным критериям. |
| 4 | Статус выполнения операции | Строка | Статус выполнения операции (успешно, неуспешно). |

### 1.5 Алгоритмы

**Основной алгоритм решения:**

1. Добавление заявки:
   * Ввод данных о заявке.
   * Сохранение данных в базе данных.
2. Редактирование заявки:
   * Выбор заявки для редактирования.
   * Изменение этапа выполнения, описания проблемы, ответственного.
   * Обновление данных в базе.
3. Отслеживание статуса заявки:
   * Отображение списка заявок.
   * Получение уведомлений о смене статуса.
   * Поиск заявок по номеру или параметрам.
4. Назначение ответственных:
   * Добавление исполнителя к заявке.
   * Отслеживание состояния работы и получение уведомлений о завершении.
   * Добавление комментариев на форме заявки.
5. Расчет статистики:
   * Подсчет количества выполненных заявок.
   * Расчет среднего времени выполнения заявок.
   * Генерация статистики по типам неисправностей.

**Детализация алгоритма расчета количества выполненных заявок:**

1. Инициализация счетчика выполненных заявок в нуле.
2. Для каждой заявки в базе данных:
   * Если статус заявки равен "выполнено", увеличить счетчик на 1.
3. Вывести значение счетчика выполненных заявок.

**Детализация алгоритма расчета среднего времени выполнения заявок:**

1. Инициализация переменных: общее время выполнения, количество выполненных заявок.
2. Для каждой заявки в базе данных:
   * Если статус заявки равен "выполнено":
     + Получить дату добавления и дату выполнения заявки.
     + Рассчитать время выполнения заявки.
     + Добавить время выполнения к общему времени выполнения.
     + Увеличить счетчик выполненных заявок на 1.
3. Рассчитать среднее время выполнения как общее время выполнения, поделенное на количество выполненных заявок.
4. Вывести значение среднего времени выполнения заявок.

**Блок-схема алгоритма учета заявок на ремонт оборудования:**

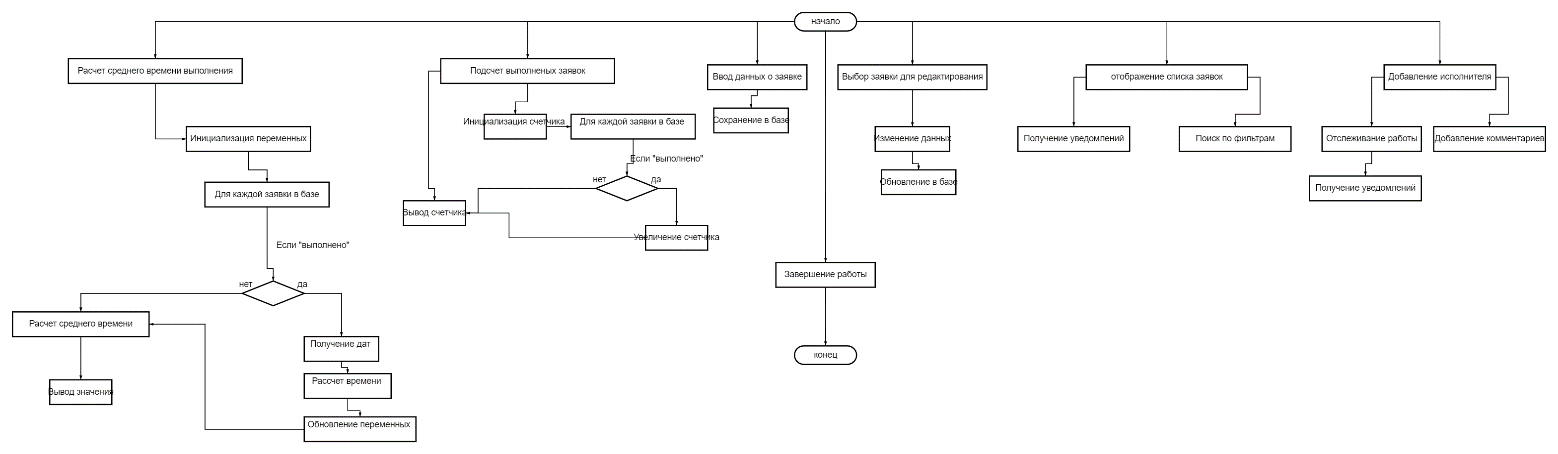


Рис 1. Блок-схема алгоритма учета заявок на ремонт оборудования

### 1.6 Диаграмма активностей

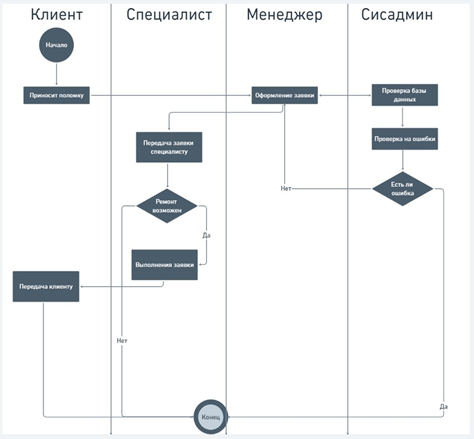


Рис 2. Диаграмма активностей

### 1.7 Интерфейс

Был разработан макет(варфрейм) меню авторизации, главного меню, списка заявок, редактирования, добавления и отслеживания заказов (разработан с помощью сайта wireframe.cc)

### 1.7.1 Дизайн программного модуля

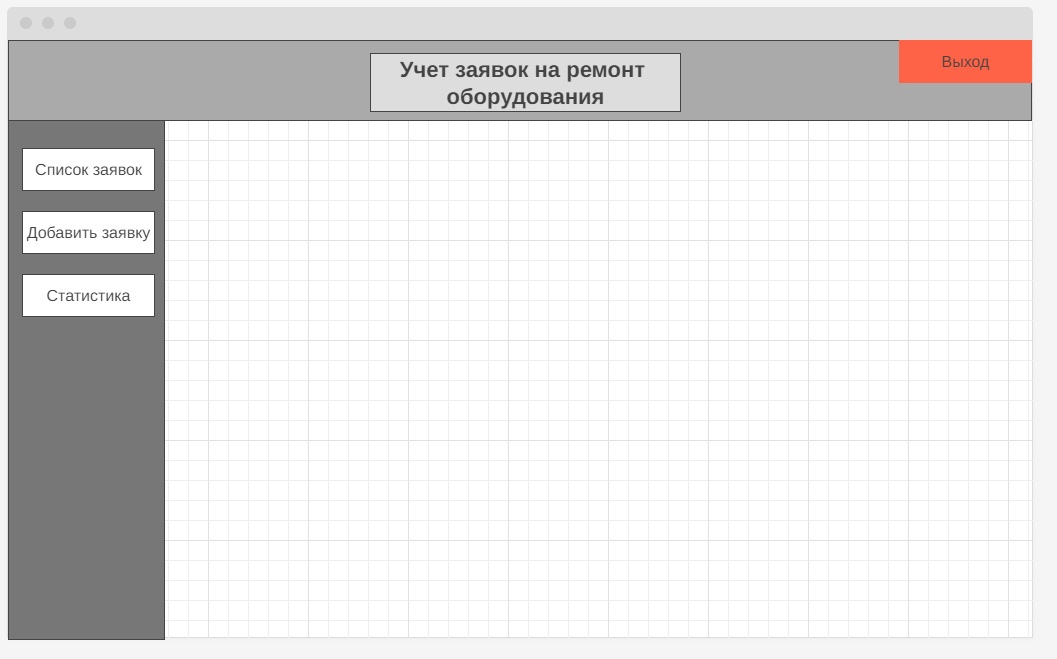


Рис 3. Меню учета заявок на ремонт оборудования

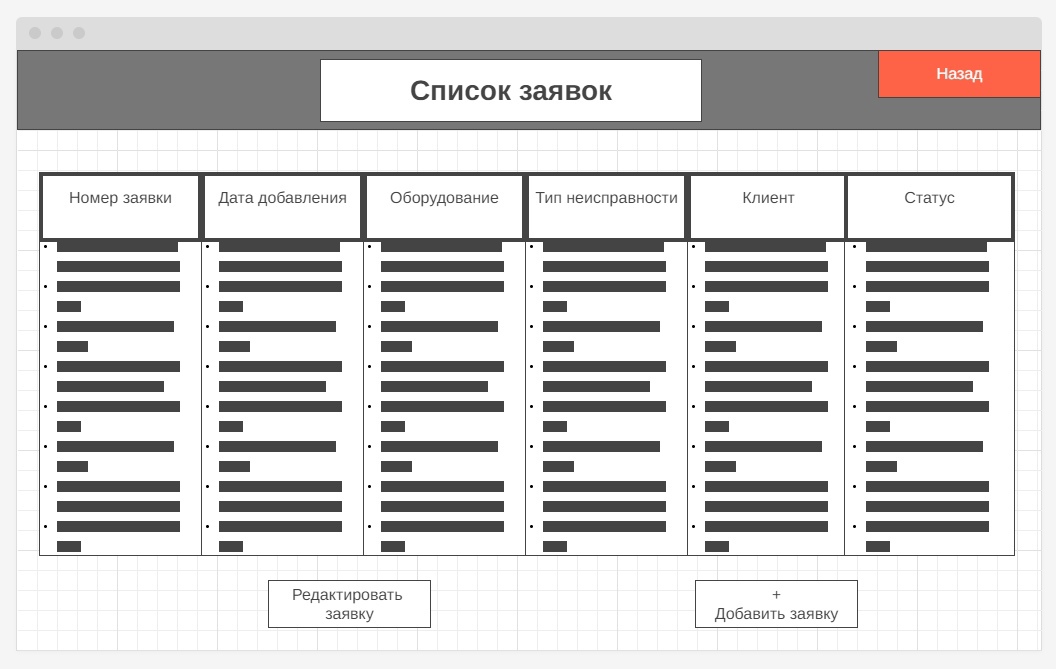


Рис 4. Меню списка заявок

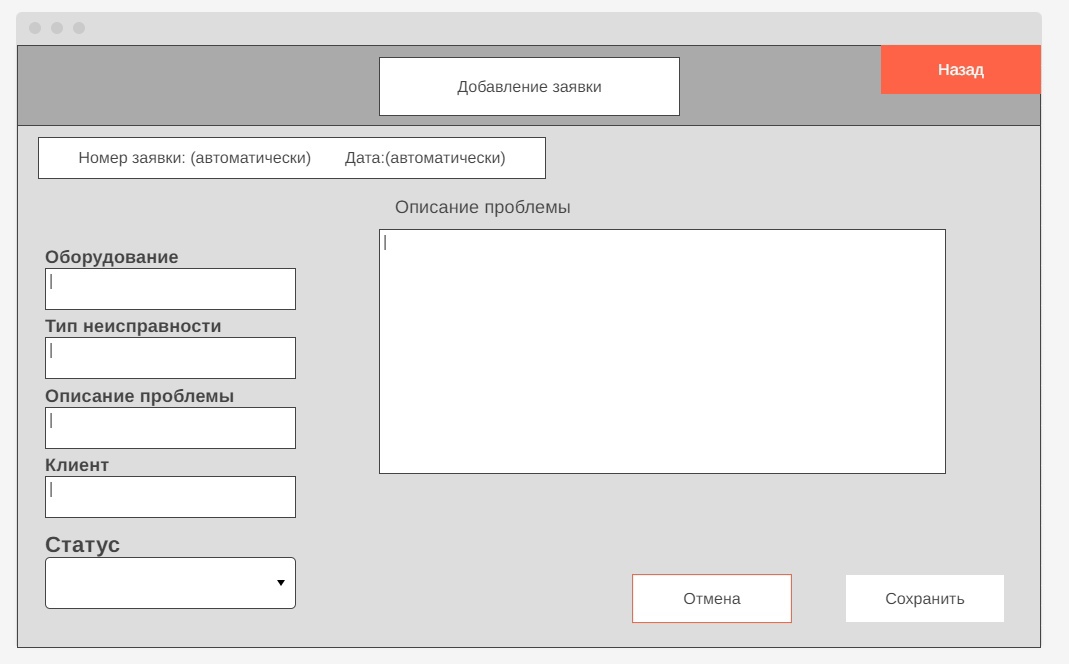


Рис 5. Меню добавления заявки

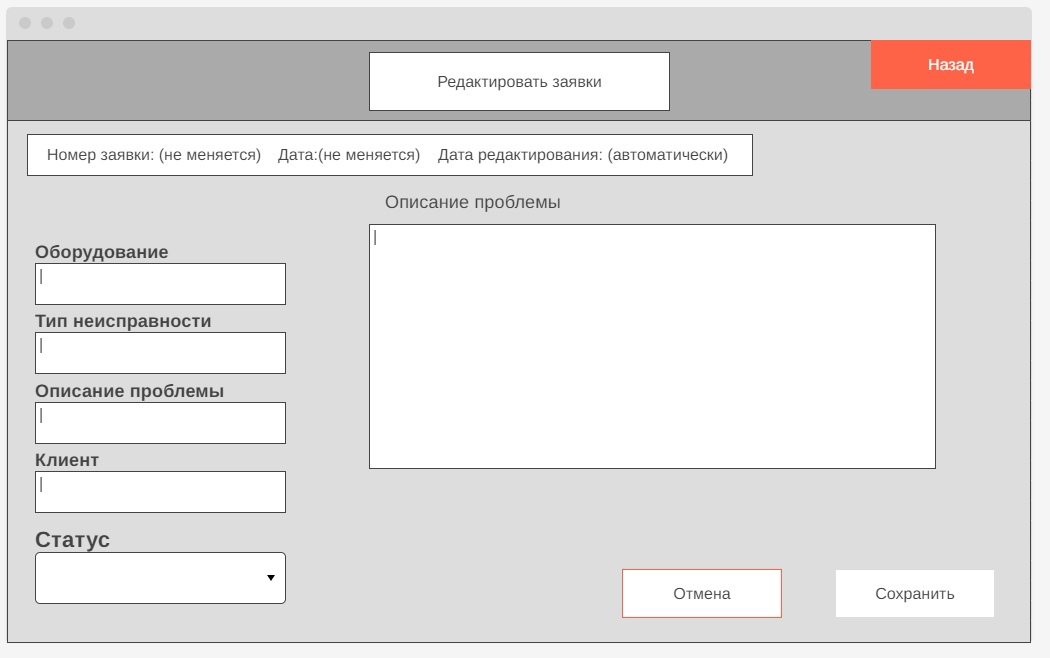


Рис 6. Меню редактирования заявки

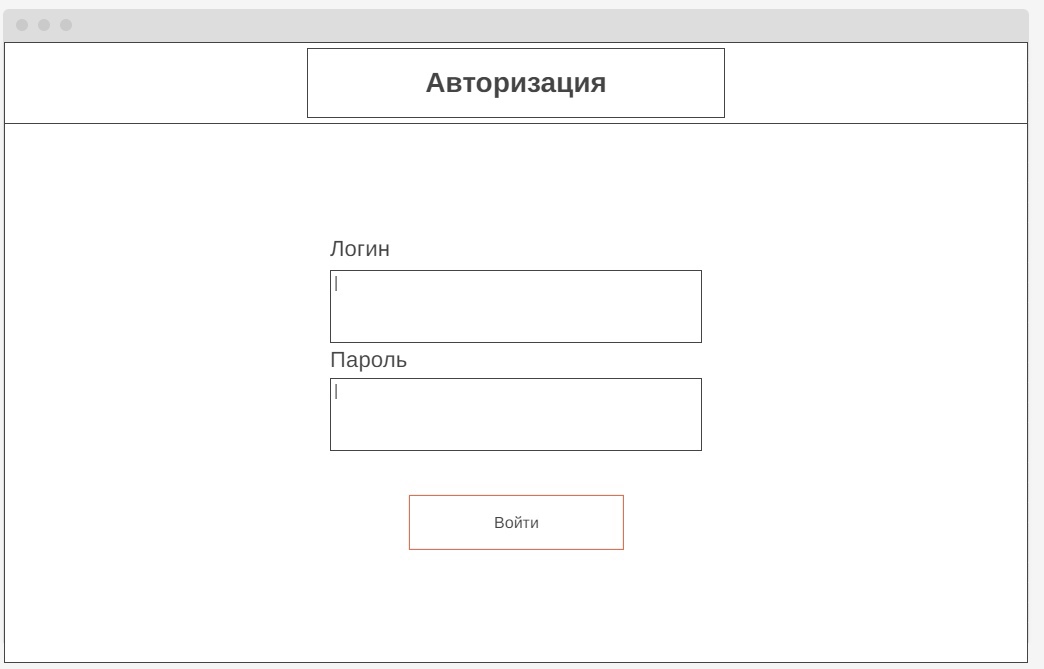


Рис 7. Меню авторизации

### 1.7.2 Дизайн мобильной версии

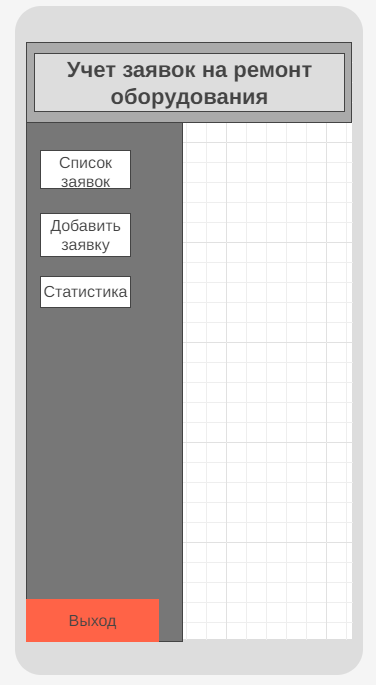


Рис 8. Меню учета заявок на ремонт оборудования для мобильной версии

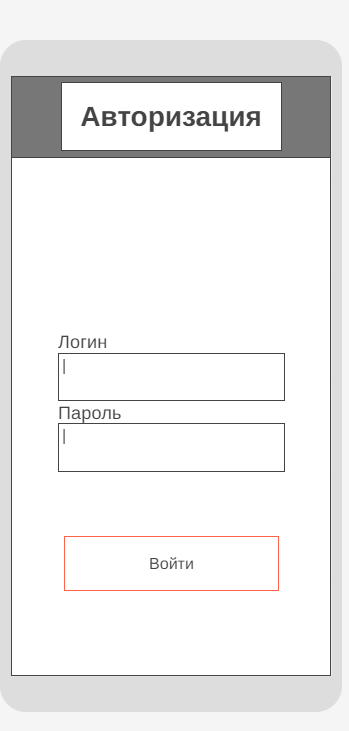


Рис 9. Меню авторизации для мобильной версии

### 1.8 Реализация

В данном докладе представлена документация о разработке программы, использующей технологию WPF. Программа включает несколько интерфейсных окон, в том числе:

1. Стартовое окно авторизации

2. Меню функций

3. Список Заявок

4. Редактирование Заявки

5. Добавление Заявки

6. Отслеживание заявок

**Окно Авторизации**

Окно авторизации предназначено для ввода логина и пароля пользователя. В случае успешной авторизации, происходит переход к главному меню.

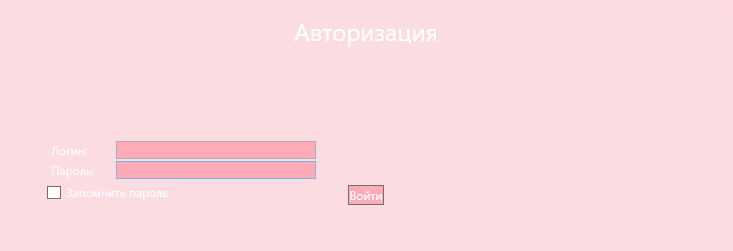


Рис 10. Стартовое окно авторизации

**Код окна авторизации**

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

// Проверяем, есть ли сохраненный логин и пароль

if (Application.Current.Properties.Contains("SavedLogin") && Application.Current.Properties.Contains("SavedPassword"))

{

string savedLogin = Application.Current.Properties["SavedLogin"].ToString();

string savedPassword = Application.Current.Properties["SavedPassword"].ToString();

// Устанавливаем сохраненные значения в соответствующие поля

LoginTextBox.Text = savedLogin;

PasswordBox.Password = savedPassword;

RememberCheckBox.IsChecked = true; // Помечаем галочку "Запомнить пароль"

}

}

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

{

string login = LoginTextBox.Text;

string password = PasswordBox.Password;

bool rememberPassword = RememberCheckBox.IsChecked ?? false;

if (login == "123" && password == "321")

{

if (rememberPassword)

{

// Сохраняем логин и пароль в Properties

Application.Current.Properties["SavedLogin"] = login;

Application.Current.Properties["SavedPassword"] = password;

}

MainMenu mainMenu = new MainMenu();

mainMenu.Show();

this.Close();

}

else

{

MessageBox.Show("Неверный логин или пароль. Попробуйте снова.");

}

}

}

}

**Главное Меню**

Главное меню предоставляет пользователю выбор различных действий, таких как добавление заявки, редактирование заявки, отслеживание статуса заказа и просмотр списка заявок.

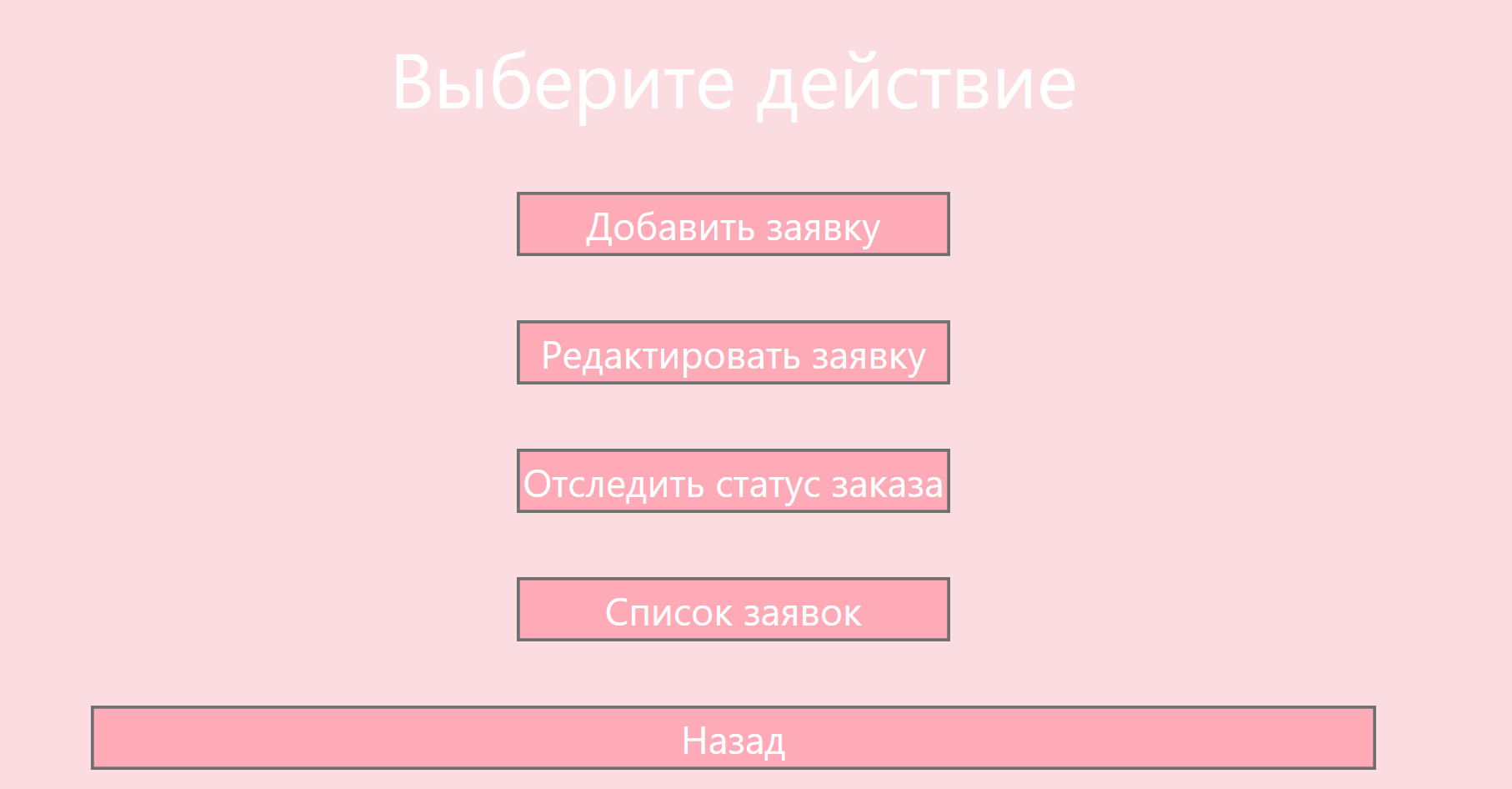


Рис 11. Меню функций

**Код для Меню функций:**

public partial class MainMenu : Window

{

public MainMenu()

{

InitializeComponent();

}

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

{

AddRequest addRequest = new AddRequest(); addRequest.Show(); this.Close();

}

private void EditRequestButton\_Click(object sender, RoutedEventArgs e) {

EditRequest editRequest = new EditRequest(); editRequest.Show(); this.Close();

}

private void TrackOrderStatusButton\_Click(object sender, RoutedEventArgs e) {

TrackOrderStatus trackOrderStatus = new TrackOrderStatus(); trackOrderStatus.Show(); this.Close();

}

private void RequestListButton\_Click(object sender, RoutedEventArgs e) {

RequestList requestList = new RequestList(); requestList.Show(); this.Close();

}

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

{

MainWindow mainWindow = new MainWindow(); mainWindow.Show(); this.Close();

}

}

**Список Заявок**

Экран списка заявок отображает информацию о всех существующих заявках в виде таблицы. Пользователь имеет возможность сортировать заявки по различным параметрам.

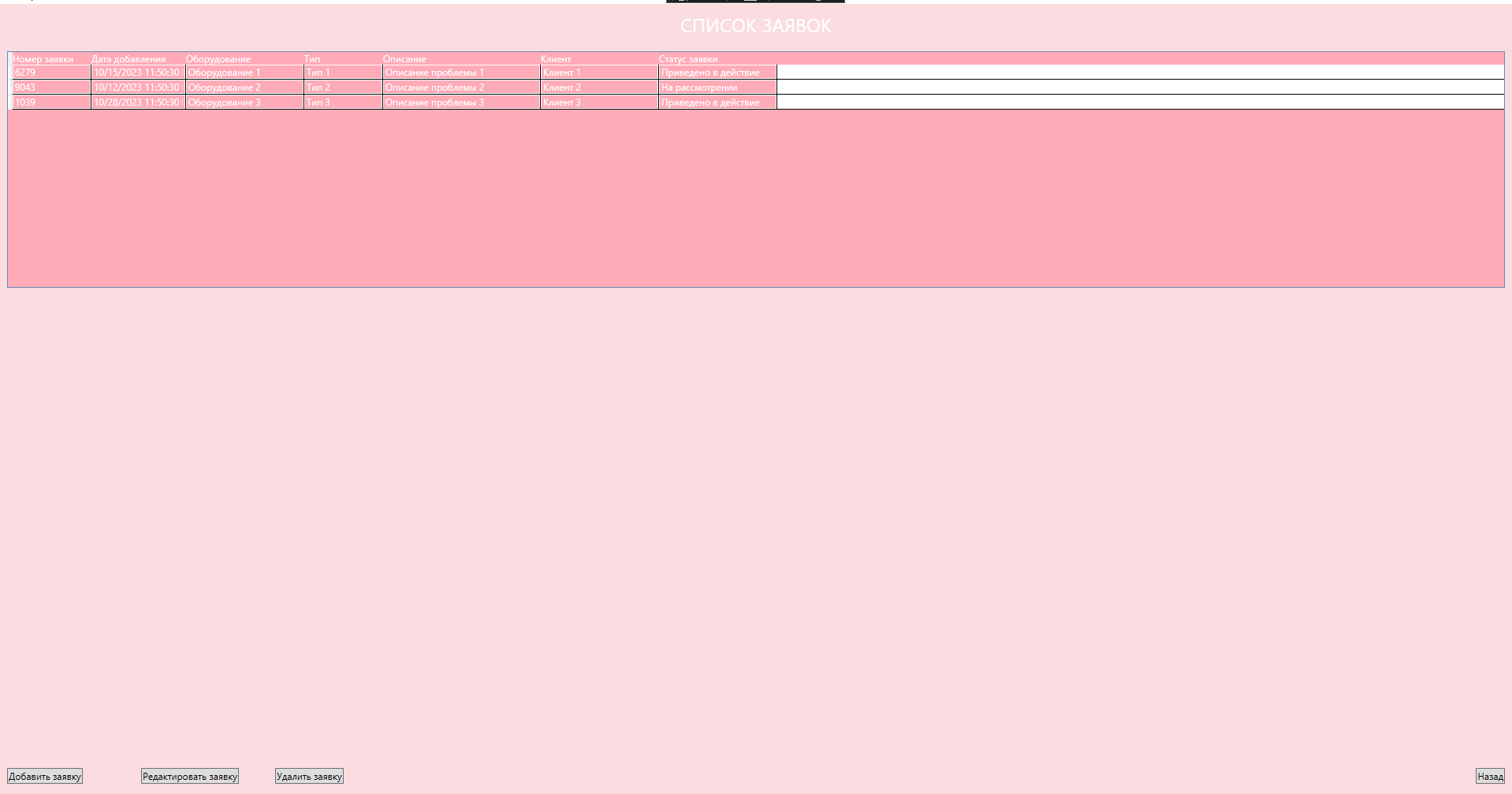


Рис 12. Список Заявок

Код для Списка заявок:

public partial class RequestList : Window

{

public RequestList()

{

InitializeComponent();

PopulateDataGrid();

}

public class Request

{

public int Number { get; set; } public DateTime Date { get; set; } public string Equipment { get; set; } public string Type { get; set; } public string Description { get; set; } public string Client { get; set; } public string Status { get; set; }

}

private void AddRequestButton\_Click(object sender, RoutedEventArgs e) {

AddRequest addRequest = new AddRequest(); addRequest.Show(); this.Close();

}

private void EditRequestButton\_Click(object sender, RoutedEventArgs e) {

EditRequest editRequest = new EditRequest(); editRequest.Show(); this.Close();

}

private void DeleteRequestButton\_Click(object sender, RoutedEventArgs e) {

// Код для удаления заявки

}

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

{

MainMenu mainMenu = new MainMenu(); mainMenu.Show(); this.Close();

}

private void PopulateDataGrid()

{

Random random = new Random();

for (int i = 0; i < 3; i++)

{

Request request = new Request

{

Number = random.Next(1000, 9999),

Date = DateTime.Now.AddDays(-random.Next(1, 30)),

Equipment = "Оборудование " + (i + 1),

Type = "Тип " + (i + 1),

Description = "Описание проблемы " + (i + 1),

Client = "Клиент " + (i + 1),

Status = random.Next(2) == 0 ? "На рассмотрении" : "Приведено в действие"

};

dataGrid.Items.Add(request);

}

}

}

**Редактирование Заявки**

Окно редактирования заявки предоставляет пользователю возможность изменить данные о заявке, такие как оборудование, тип неисправности, описание проблемы и статус.

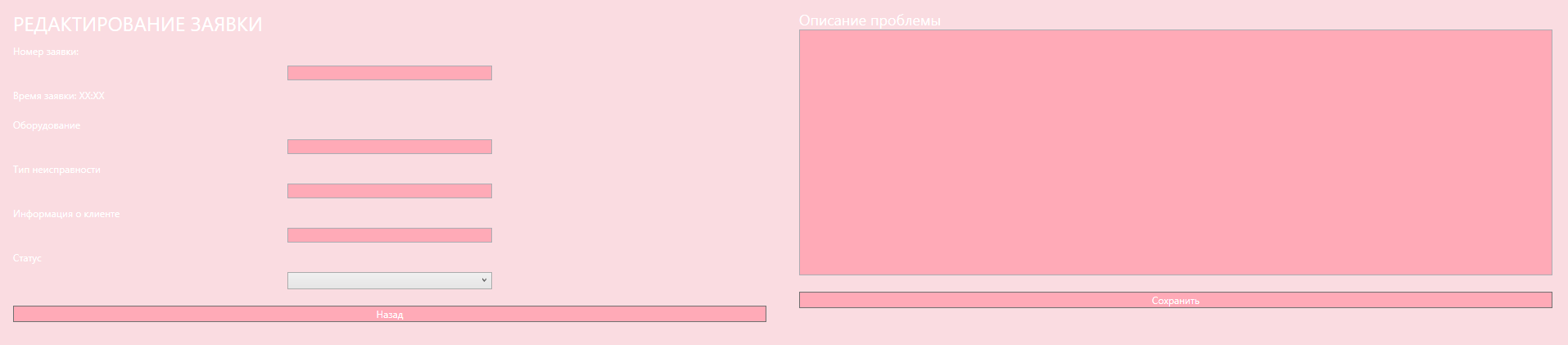


Рис 13. Окно Редактирования Заявок Временный код для окна Редактирования заявок:

public EditRequest()

{

InitializeComponent();

}

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

{

MainMenu mainMenu = new MainMenu(); mainMenu.Show(); this.Close();

}

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

{

string requestNumber = RequestNumberTextBox.Text; // есть номера заявки, которое можно использовать. }

**Добавление Заявки**

Экран добавления заявки предоставляет пользователю возможность создать новую заявку, указав необходимые данные.

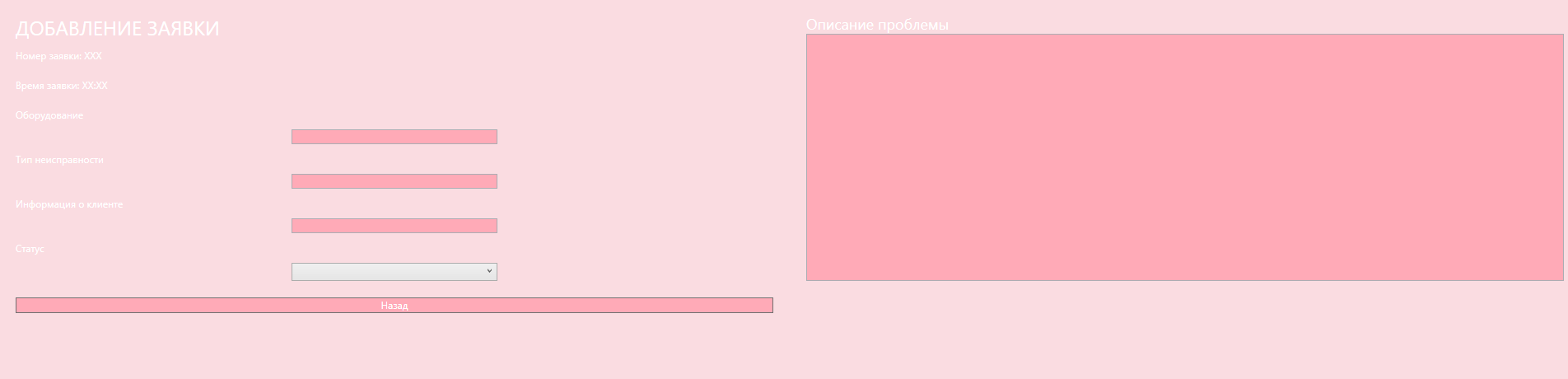


Рис 14. Окно Добавления Заявок Временный код для окна Добавления заявок:

public AddRequest()

{

InitializeComponent();

}

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

{

MainMenu mainMenu = new MainMenu(); mainMenu.Show(); this.Close();

}

private void ComboBox\_SelectionChanged(object sender, SelectionChangedEventArgs e) {

}

**Отслеживание Заказов**

Окно отслеживания заказов позволяет пользователю следить за статусами размещенных заказов.

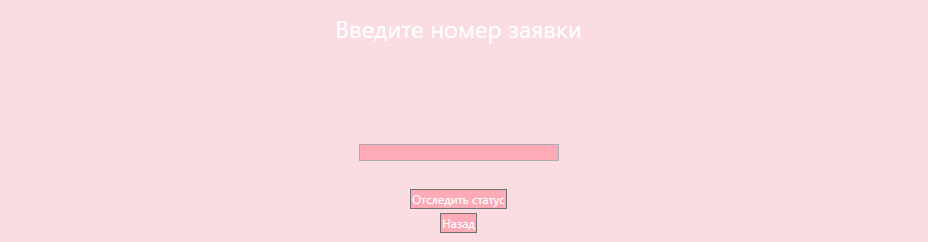


Рис 15. Окно Отслеживание заявок

Временный код для отслеживания статуса заказа:

public TrackOrderStatus()

{

InitializeComponent();

}

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

{

// Здесь будет код для отслеживания заявок

}

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

{

MainMenu mainMenu = new MainMenu(); mainMenu.Show(); this.Close(); }

Разработка новых функций, таких как:

1. Уведомление о неверном вводе данных
2. Запрос подтверждения на совершение безвозвратного удаления
3. Запрос подтверждения на неотвратимую операцию

**Уведомление о неверном вводе данных**

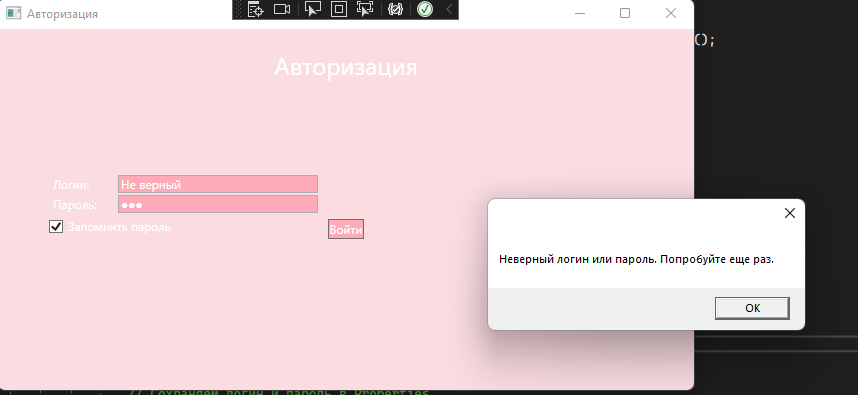


Рис 16. Окно с предупреждением об ошибке

Временно был поставлен единственный верный пароль, “Верный”, так что при вводе “Неверный” и последующем нажатии кнопки “Войти”, появляется окно с предупреждением об ошибке **Изменения в коде:**

{

MessageBox.Show("Неверный логин или пароль. Попробуйте еще раз.");

}

**Запрос подтверждения на совершение безвозвратного удаления**

При выделении заявки из списка, и последующем нажатии на кнопку “Удалить заявку”, появляется окно с запросом на подтверждения действия.

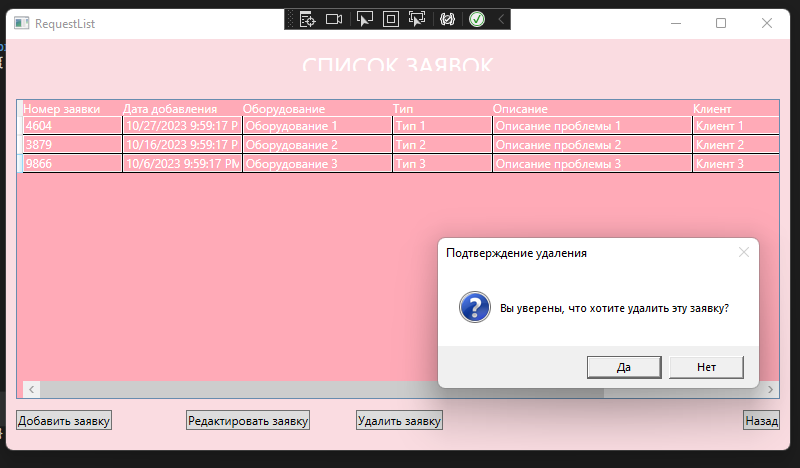


Рис 17. Запрос на подтверждение действия

**Изменения в коде:**

// Проверка наличия выделенной строки

if (dataGrid.SelectedItem != null)

{

// Запрос подтверждения удаления

MessageBoxResult result = MessageBox.Show("Вы уверены, что хотите удалить эту заявку?", "Подтверждение удаления", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

// Удаление выбранной заявки из списка

var selectedRequest = (Request)dataGrid.SelectedItem;

dataGrid.Items.Remove(selectedRequest);

}

}

else

{

MessageBox.Show("Пожалуйста, выберите заявку для удаления.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

**Запрос подтверждения на неотвратимую операцию**

При нажатии на кнопку назад, в окне заполнения новой заявки, программа проверяет, есть ли заполненные поля. Если таких полей нет – кнопка работает как обычно и окно меняется на главное меню, если есть заполненные поля – то появляется запрос на подтверждения.

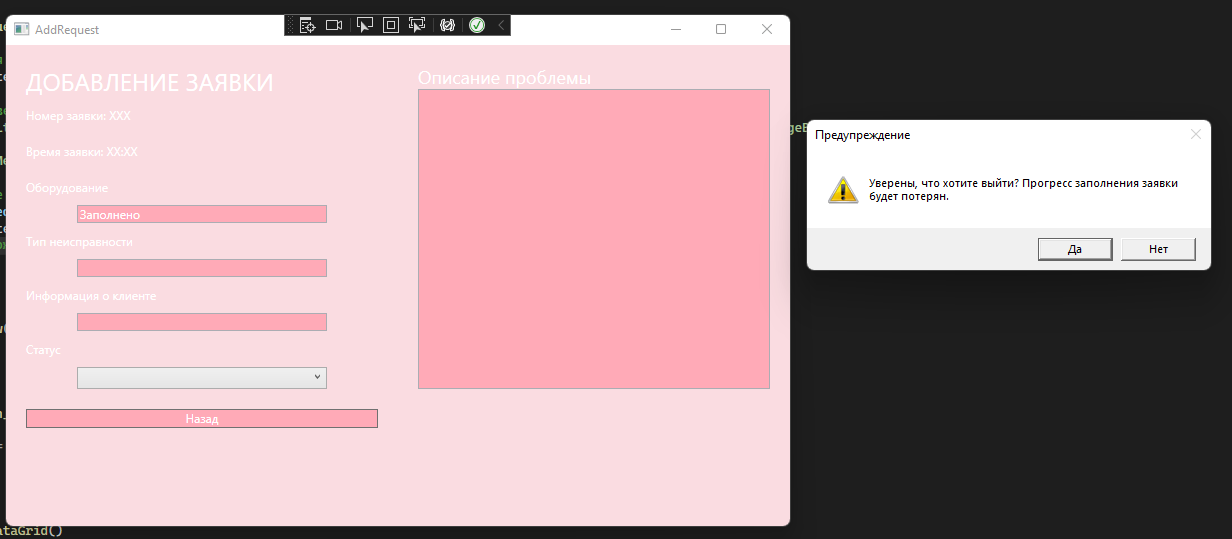


Рис 18. Запрос на подтверждение операции

**Изменения в коде:**

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

{

if (unsavedData)

{

MessageBoxResult result = MessageBox.Show("Уверены, что хотите выйти? Прогресс заполнения заявки будет потерян.", "Предупреждение", MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes)

{

MainMenu mainMenu = new MainMenu();

mainMenu.Show();

this.Close();

}

}

else

{

MainMenu mainMenu = new MainMenu();

mainMenu.Show();

this.Close();

}

}

### 1.9 Отладка модуля

Была выполнена отладкапрограммного обеспечения с использованием инструментальных средств

Тестируется функция проверки пароля:

Вводим верный пароль и нажимаем на “Готов”

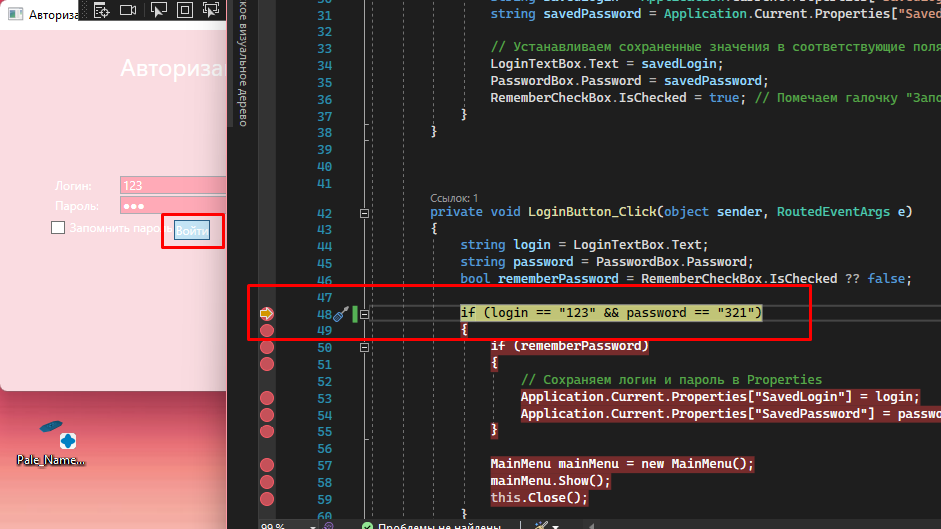


Рис 19. Проверка верности логина и пароля

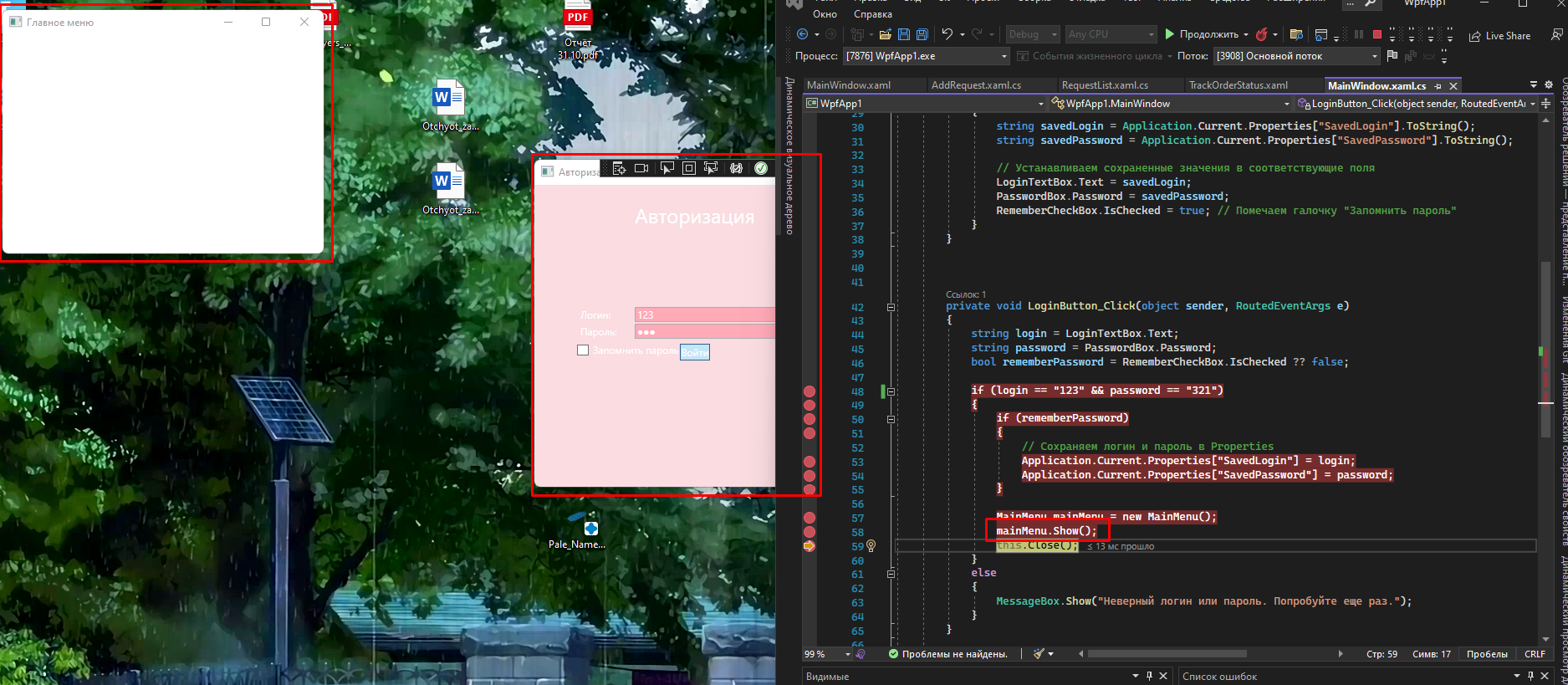


Рис 20. Появление нового окна

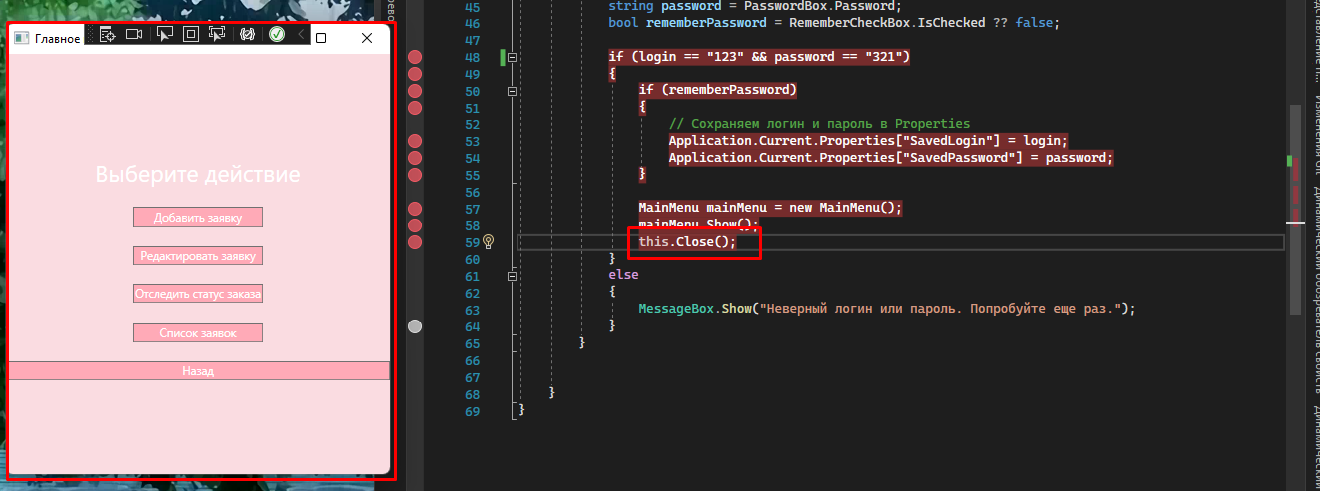


Рис 21. Закрытие прошлого окна

Теперь вводим неверный пароль:

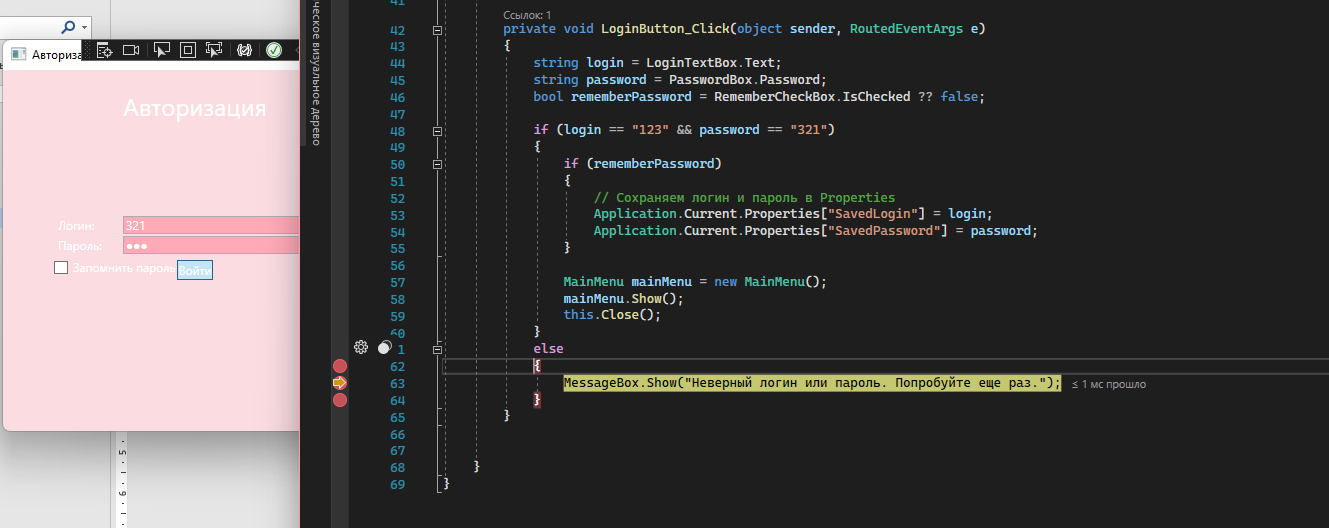


Рис 22. Проверка пароля, с неверно введенными данными

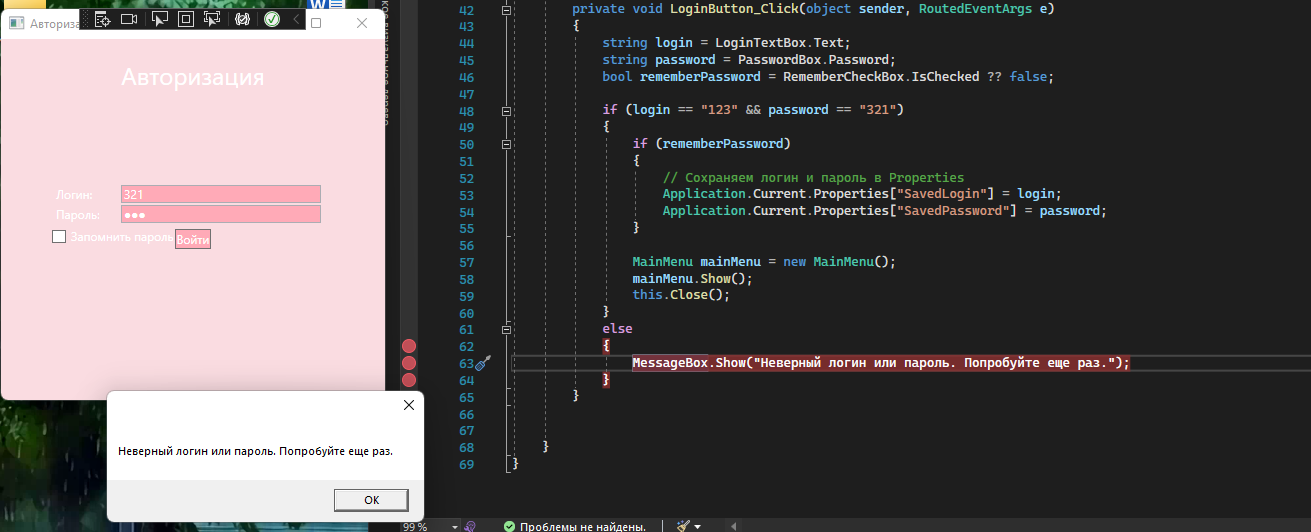


Рис 23. Появление окна ошибки

Все работает корректно.

### 1.10 Результаты тестирования

**Тест:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название теста | Ожидаемый результат | Фактический результат |
| 1 | Ввод верного пароля и нажатие кнопки "Войти" | Успешная аутентификация, переход на следующую страницу | Успешная аутентификация, переход на следующую страницу |
| 2 | Ввод неверного пароля и нажатие кнопки "Войти" | Отображение сообщения об ошибке и неудачная попытка аутентификации | Отображение сообщения об ошибке и неудачная попытка аутентификации |
| 3 | Нажатие кнопки для выбора страницы "Список заявок" | Переход на страницу "Список заявок" | Переход на страницу "Список заявок" |
| 4 | Нажатие кнопки для выбора страницы "Добавление/редактирование заявки" | Переход на страницу "Добавление/редактирование заявки" | Переход на страницу "Добавление/редактирование заявки" |
| 5 | Нажатие кнопки для выбора страницы "Отслеживание заявки" | Переход на страницу "Отслеживание заявки" | Переход на страницу "Отслеживание заявки" |
| 6 | При добавлении заявки нажать на кнопку "Назад" с заполненными полями | Появится подтверждение о потере данных, действие отменится | Появится подтверждение о потере данных, действие отменится |
| 7 | В окне списка заявок выделить одну заявку и нажать кнопку "Удалить" | Появится подтверждение удаления заявки | Появится подтверждение удаления заявки |
| 8 | Подтвердить удаление заявки | Заявка будет удалена из списка | Заявка будет удалена из списка |
| 9 | Нажать на кнопку "Добавить заявку" и ввести данные | Новая заявка будет добавлена с введенными данными | Новая заявка будет добавлена с введенными данными |
| 10 | Нажать на кнопку "Редактировать заявку" и ввести данные | Заявка будет изменена в соответствии с введенными данными | Заявка будет изменена в соответствии с введенными данными |

### 2. Модуль B

### 2.1 ER-диаграмма

Была разработана ER-диаграмма для учета заявок на ремонт оборудования.

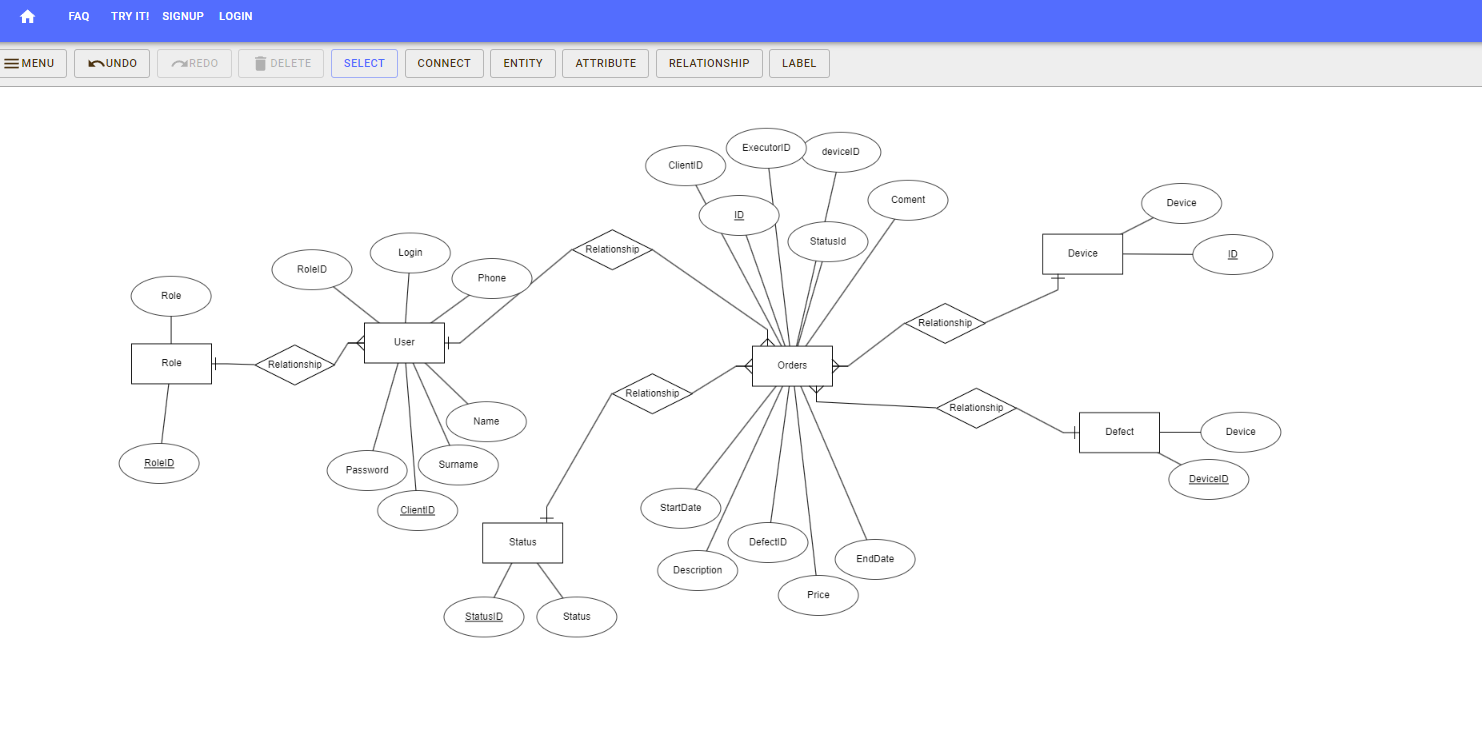


Рис 24. ER-диаграмма учёта заявок

**Приведение к 3НФ**

В результате проделанной работы была создана ER-диаграмма, описывающая

структуру базы данных для учета заявок на ремонт оборудования. База данных соответствует третьей нормальной форме и обеспечивает ссылочную

целостность.

### 2.2 Создание БД

**Создание базы данных MSSQL:**

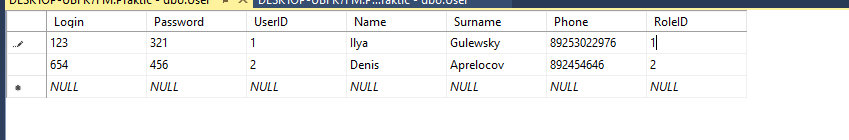
Была разработана и создана база данных в СУБД MSSQL, названная "Praktic". База данных была развернута с учетом предварительной разработки ER-диаграммы.

**Создание таблиц и определение отношений:**

В рамках создания базы данных были разработаны и созданы необходимые таблицы. Отношения между таблицами были настроены с учетом предполагаемых связей и зависимостей.

### 2.2.1 Запросы к базе данных

**Подключение таблицы "User" к приложению WPF:**

В разрабатываемом приложении WPF была реализована возможность работы с пользователями, включая регистрацию и аутентификацию. Таблица "User" была успешно подключена к приложению, позволяя взаимодействовать с данными пользователей, включая логины и пароли.  
  
 Рис 25. Данные в таблице User

Код, для работы окна авторизации с базой данных:

string connectionString = "Data Source=DESKTOP-UBFK7FM;Initial Catalog=Praktic;Integrated Security=True";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = "SELECT Password FROM [User] WHERE Login = @Login";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.Add(new SqlParameter("@Login", SqlDbType.NVarChar, 50));

command.Parameters["@Login"].Value = login;

string storedPassword = command.ExecuteScalar() as string;

if (storedPassword != null && storedPassword == password)

{

if (rememberLogin)

{

Application.Current.Properties["SavedLogin"] = login;

}

MainMenu mainMenu = new MainMenu();

mainMenu.Show();

this.Close();

}

else

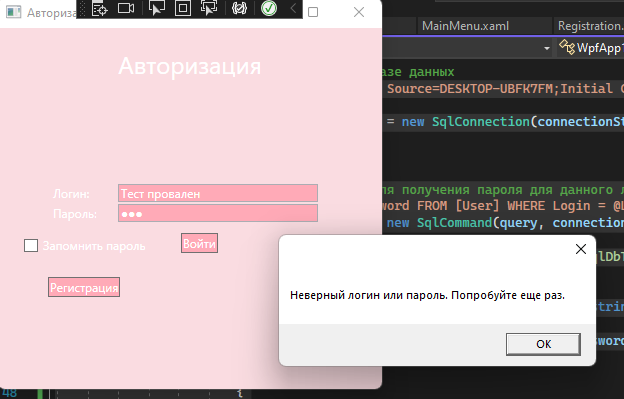
{

MessageBox.Show("Неверный логин или пароль. Попробуйте еще раз.");

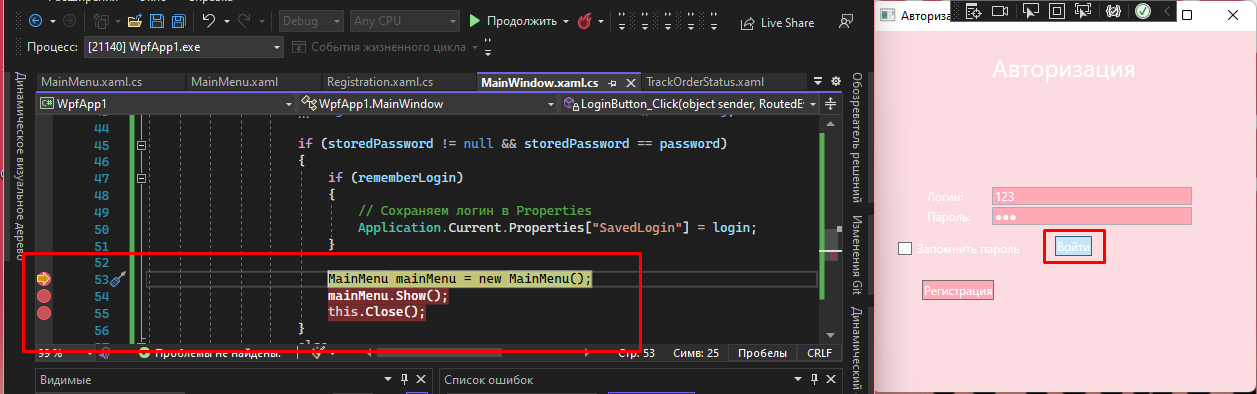
}

Проверка работоспособности:

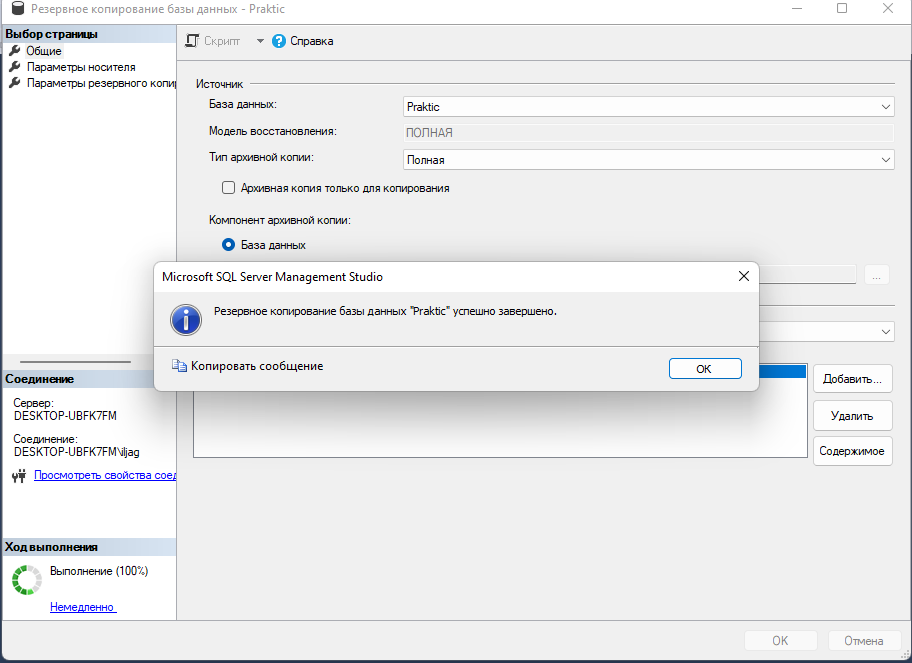
У пользователя Ilya логин: 123, следовательно, при другом логине программа не даст авторизоваться под именем Ilya

Рис 26. Не верный логин

При вводе верного пароля, срабатывает функция перемещения на главное меню, это показано с помощью режима отладки:

 Рис 27. Верный логин

### 2.2.2 Резервное копирование БД

 Рис 28. Создание резервной копии

### 2.2.3 Принцип регистрации

Создано окно регистрации с возможностью выбора группы пользователя, либо специалист, либо Менеджер, либо Системный программист.

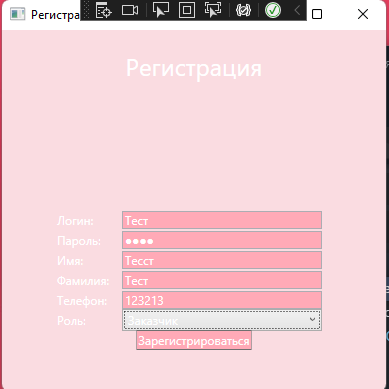
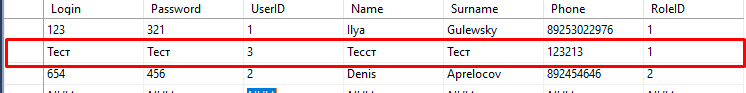


Рис 29. Тест регистрации

 Рис 27. Появившиеся данные

Мы выбрали роль Специалист, следовательно, в RoleID указана единица.

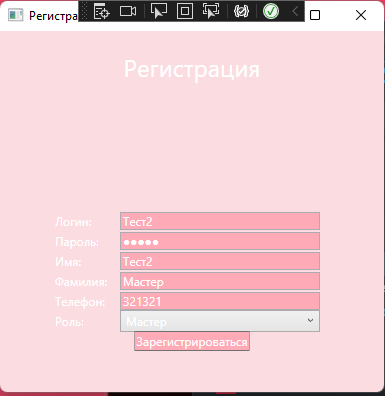
Теперь укажем роль Менеджера  


Рис 30. Тест регистрации Менеджер

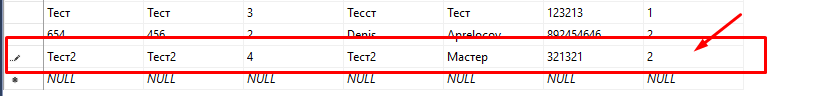


Рис 21. Появившиеся данные

### 2.2.4 Уровни доступа

При указании роли Менеджер, во время регистрации, аккаунту присваивается RoleID 2.

### 2.3 Руководство системному программисту

Введение

Данное руководство предназначено для системных программистов, работающих над разработкой и сопровождением приложения для учета заявок на ремонт оборудования.

1. Обзор приложения

1.1. Назначение

Приложение предназначено для учета и обработки заявок на ремонт оборудования. Оно обеспечивает следующие основные функции:

- Регистрация новых заявок.

- Отслеживание статуса заявок.

- Генерация отчетов по заявкам.

- Управление данными об оборудовании и клиентах.

1.2. Технологии

Приложение разрабатывается с использованием следующих технологий:

- Язык программирования: [указать язык]

- База данных: [указать СУБД]

- Интерфейс пользователя: [указать фреймворк или библиотеку]

2. Структура проекта

2.1. Каталоги и файлы

Описание основных каталогов и файлов в проекте:

- \*\*/src\*\*: Исходные коды приложения.

- \*\*/docs\*\*: Документация проекта.

- \*\*/db\*\*: Скрипты создания и обновления базы данных.

3. Разработка

3.1. Соглашения по кодированию

- Комментарии к коду обязательны для всех функций и сложных участков кода.

3.2. Работа с версиями

- Используется система контроля версий.

3.3. Тестирование

- Все изменения должны сопровождаться соответствующими тестами.

- Перед каждым релизом проводится тестирование функционала.

4. Документация

4.1. Техническая документация

- Техническое описание архитектуры приложения.

- Схема базы данных.

- Описание API (если применимо).

4.2. Руководство пользователя

- Создание и поддержка документации для конечных пользователей.

5. Сопровождение

5.1. Регулярные обновления

- План регулярных обновлений и поддержки.

5.2. Отчеты об ошибках

- Процедура подачи и отслеживания отчетов об ошибках.

Заключение

Данное Руководство системному программисту предназначено для обеспечения эффективной разработки, тестирования и сопровождения приложения. В случае вопросов обращайтесь к контактной информации.

### 2.4 Варианты модификации программного обеспечения

1. Добавление функционала уведомлений

Рекомендуется добавить механизм уведомлений для пользователей о статусе и изменениях в заявках. Это может быть реализовано через электронные письма, SMS или панель уведомлений в приложении.

2. Расширение функционала отчетности

Предлагается расширить возможности отчетности, добавив новые типы отчетов, такие как статистика по времени выполнения заявок, наиболее часто встречающиеся проблемы и др.

3. Интеграция с внешними системами

Рассмотрите возможность интеграции приложения с другими системами, такими как системы учета клиентов или системы мониторинга оборудования. Это может улучшить обмен данными и сократить дублирование информации.

4. Добавление поддержки множественных ролей пользователей

Предлагается внести изменения в систему управления ролями для поддержки нескольких уровней доступа, что обеспечит более гибкую настройку прав пользователей.

5. Реализация мобильного приложения

Рекомендуется разработать мобильное приложение для обеспечения удобного доступа к системе независимо от места нахождения пользователя.

### 3. Модуль C

### 3.1 Новый пользователь

Приложение предоставляет возможность создания новых пользователей через процесс регистрации. Пользователь вводит необходимую информацию, такую как логин, пароль, имя, фамилия, телефон и роль (Системный программист, Менеджер или Специалист). После успешной регистрации данные пользователя сохраняются в базе данных, а сам пользователь получает доступ к функционалу приложения.

### 3.2 Роль Менеджера

Роль Менеджера в приложении предоставляет возможность управления заявками, просмотр списка заявок, их редактирование и удаление. Менеджер может эффективно координировать рабочие процессы, просматривать статусы заявок и взаимодействовать с другими пользователями при необходимости.

### 3.3 Настройка и компоненты

Приложение построено на технологии WPF (Windows Presentation Foundation) с использованием языка программирования C#. Взаимодействие с базой данных осуществляется с использованием технологии ADO.NET и языка SQL для выполнения запросов. Для лучшего визуального восприятия применены стили и темы оформления, что делает пользовательский интерфейс интуитивно понятным.

### 3.4 Демонстрация функционала согласно ТЗ

1. Регистрация и Авторизация

Регистрация нового пользователя: Приложение предоставляет возможность новым пользователям зарегистрироваться, указав логин, пароль, имя, фамилию, телефон и роль (Системный программист, Менеджер или Специалист). После успешной регистрации данные пользователя сохраняются в базе данных.

Авторизация пользователя: Зарегистрированные пользователи могут войти в систему, вводя свой логин и пароль. После успешной авторизации система определяет роль пользователя, что влияет на доступные ему функции.

1. Создание/Редактирование заявок

Создание новой заявки: Пользователь, авторизованный как Менеджер или Специалист, может добавить новую заявку, указав оборудование, тип неисправности, информацию о клиенте и описание проблемы. Дополнительно можно выбрать статус заявки из выпадающего списка.

Редактирование существующих заявок: Менеджер имеет возможность редактировать атрибуты заявок, такие как статус, оборудование, тип неисправности, информацию о клиенте и описание проблемы.

1. Уровни доступа

**Роль Менеджера:** Менеджер имеет полный доступ к управлению заявками, включая просмотр, создание, редактирование и удаление.

**Роль Специалиста:** Специалист может просматривать заявки, менять их статусы, а также создавать и редактировать.

**Системный администратор:** Системный администратор обладает полными правами доступа и контроля над всеми аспектами приложения, включая управление пользователями, заявками и настройками.

1. Просмотр имеющихся заявок

Отображение списка заявок: Пользователи могут просматривать список имеющихся заявок, где видны основные атрибуты каждой заявки, такие как ID, DeviceID, ExecutorID, StartDate, Comment и Description.

1. Отслеживание заявки

Отслеживание статуса заявки: Пользователи могут отслеживать статус своих заявок, а Менеджер может отслеживать статус всех заявок в системе.

1. Работа с базой данных

Взаимодействие с базой данных: Приложение взаимодействует с базой данных для хранения информации о пользователях, их ролях и заявках. Применяется технология ADO.NET и язык SQL для выполнения запросов к базе данных.

**3.5 Качественные характеристики**

**Удобство использования**

Интерфейс приложения создан с учетом простоты и понятности для пользователей. Он обеспечивает удобное взаимодействие с функционалом, что снижает время обучения новых пользователей.

**Надежность**

Приложение использует базу данных для хранения информации о пользователях и заявках, что обеспечивает надежность хранения данных и их целостность.

**Эффективность**

Роль Менеджера позволяет эффективно управлять заявками, повышая оперативность принятия решений и координации рабочих процессов.

**Безопасность**

Для обеспечения безопасности пользовательских данных используется параметризованный SQL-запрос при взаимодействии с базой данных, что защищает от SQL-инъекций.

### **ЗАКЛЮЧЕНИЕ**

За время прохождения учебной практики выполнены все задания, предусмотренные программой.

В ходе разработки приложения были успешно реализованы основные функциональные требования, представленные в Техническом Задании. Приложение предоставляет удобный и интуитивно понятный интерфейс. Разработанное приложение предоставляет надежное и эффективное средство управления заявками с учетом ролей пользователей. Его гибкая структура позволяет внедрять дополнительные функции и расширять возможности в соответствии с будущими потребностями бизнеса.

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем: учебное пособие / Л. Г. Гагарина. — Москва: ФОРУМ: ИНФРА-М, 2021. — 384 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0735-1. – Текст: электронный. - URL:

<https://znanium.com/catalog/product/1214882>

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова- Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М,

2021. — 400 с. — (Среднее профессиональное образование). - ISBN 978-

5-8199-0812-9. - Текст: электронный. - URL:

<https://znanium.com/catalog/product/1189951>

1. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство

Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN

978-5-534-05047-9. — Текст: электронный // Образовательная платформа Юрайт

1. Федорова Г.Н Осуществление интеграции программных модулей, учебник для студентов учреждений среднего профессионального образования, 4-е изд., перераб. Издательство: Академия. , 2021. — 272 с - URL:<https://www.academia-library.ru/catalogue/4891/345766/>
2. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем [Электронный ресурс]:учебник для вузов. - Москва: Юрайт, 2022. - 432 с – Режим доступа: https://urait.ru/bcode/491029
3. Рудаков А. В. Технология разработки программных продуктов. Учебное пособие Издательство: Академия [https://academia- library.ru/reader/?id=401005&demo=Y](https://academia-library.ru/reader/?id=401005&demo=Y)