ФГБОУ ВО РГАТУ имени П.А. Соловьева

Авиационный колледж

УТВЕРЖДАЮ

председатель ПЦК «Программирования»

Кустова Е.А.

**ОТЧЕТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ УП.01**

Специальность ***09.02.07 Информационные системы и программирование***

**Сроки прохождения практики:** с 04.03.2024 по 23.03.2024

**Руководитель практики:**

Кустова Екатерина Андреевна

преподаватель

*(Ф.И.О., должность) (подпись)*

**Работу выполнил студент группы** ИС-3

Щукина Полина Дмитриевна

*(Ф.И.О.) (подпись)*

Оценка за практику:

Дата защиты:

Рыбинск, 2024 г.

**Аттестационный лист по учебной практике УП.01**

Ф.И.О. Щукина Полина Дмитриевна

Курс 3 группа ИС-3 .

Обучающийся(аяся) по специальности СПО

*09.02.07 Информационные системы и программирование*

*(код и наименование)*

прошел(ла) учебную практику по профессиональному модулю

*ПМ.01 Разработка модулей программного обеспечения для компьютерных систем*

в объеме *108*  часов

с 4 марта по 23 марта 2024 г.

в организации *ФГБОУ ВО РГАТУ имени П.А.Соловьева Авиационный колледж,*

*Ярославская обл., г.Рыбинск, ул. Чкалова, д.93, полигон учебных баз практик*

*(наименование организации, юридический адрес)*

Виды и качество выполнения работ:

|  |  |  |
| --- | --- | --- |
| **Виды работ, выполненные**  **обучающимся во время практики** | **Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика**  *(выполнено / не выполнено)* |  |
| Прохождение инструктажа:   1. вводного, 2. на рабочем месте, 3. по технике безопасности | *выполнено* |  |
| Разработка модулей программного обеспечения для компьютерных систем   1. создание программы по разработанному алгоритму как отдельный модуль; 2. разработка кода программного модуля на современных языках программирования; 3. отладка программы на уровне модуля; 4. разработка кода программного продукта на основе готовой спецификации на уровне модуля; 5. использование инструментальных средств на этапе отладки программного продукта; | *выполнено* |  |

У обучающегося были *сформированы/не сформированы* следующие профессиональные компетенции:

|  |  |  |
| --- | --- | --- |
| **Название**  **профессиональной (ПК) компетенции** | **Требования к умениям и практическому опыту** | **Оценка ПК**  *(освоена/*  *не освоена)* |
| ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием | разработка кода программного продукта на основе готовой спецификации на уровне модуля; | *освоена* |
| осуществлять разработку кода программного модуля на языках низкого и высокого уровней | *освоена* |
| создавать программу по разработанному алгоритму как отдельный модуль | *освоена* |
| осуществлять разработку кода программного модуля на современных языках программирования | *освоена* |
| ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств | использование инструментальных средств на этапе отладки программного продукта | *освоена* |
| выполнять отладку программы на уровне модуля | *освоена* |

Учебная практика \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(дифференцированный зачет)*

«\_\_­­\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

*(подпись руководителя практики от колледжа) (расшифровка)*

**Характеристика на обучающегося**

**по освоению общих компетенций**

**в период прохождения учебной практики УП.01**

Обучающийся(аяся)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_программу учебной практики УП.01 по ПМ.01 выполнил(а) в *полном/неполном* объеме; все задания выполнил(а)

* *самостоятельно/с некоторой помощью,*
* *качественно/недобросовестно,*
* *в соответствии с установленными сроками/не в сроки.*

За время работы проявил(а) себя как

* *ответственный/безответственный,*
* *исполнительный/неисполнительный,*
* *коммуникабельный/замкнутый,*
* *доброжелательный/наглый* сотрудник.

У обучающегося были *сформированы/не сформированы* следующие общие компетенции:

|  |  |  |
| --- | --- | --- |
| **Название**  **общей (ОК) компетенции** | **Требования к умениям** | **Оценка ОК**  *(освоена/*  *не освоена)* |
| ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам | осуществлять разработку кода программного модуля на языках низкого и высокого уровней | *освоена* |
| ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности | определять необходимые источники информации | *освоена* |
| ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие. | осуществлять разработку кода программного модуля на современных языках программирования | *освоена* |
| ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами | создавать программу по разработанному алгоритму как отдельный модуль | *освоена* |
| ОК 9. Использовать информационные технологии в профессиональной деятельности. | выполнять отладку и тестирование программы на уровне модуля | *освоена* |

Дата «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

Руководитель практики от колледжа

*(подпись)*

**Дневник учебной практики УП.01**

**Задание 1.** Выбор темы индивидуального задания для разработки программного продукта. Изучение предметной области (см. приложение 1)

|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 04.03.2024 | Название разработки | 1 |
| Описание предметной области | 1 |
| Краткая спецификация разрабатываемого модуля | 2 |
| Входные и выходные данные | 1 |

**Описание предметной области**

Интернет магазин

Интернет-магазин – это электронная платформа, предоставляющая возможность пользователям заказывать различные товары через приложение. Оно позволяет потребителям просматривать каталог товаров, выбирать из него нужные товары, осуществлять оплату и оформлять доставку товаров.

Предметная область интернет-магазина для покупателей включают возможность покупки и продажи товаров.

В данной области включены следующие основные составляющие:

1. Платформа: это основное средство взаимодействия между продавцами и клиентами. Платформа должна быть удобной для навигации, иметь привлекательный дизайн и интуитивно понятный интерфейс.
2. Каталог товаров: содержит товары из бд, содержащей информацию о каждом товаре, который предлагается в магазине. Включает в себя название, описание, изображения и цены.

Входные данные интернет-магазина:

1. Запросы клиентов: это могут быть поисковые запросы.
2. Личные данные клиентов: такие как имя и номер телефона, которые клиенты предоставляют при создании учетной записи или оформлении заказа.

Выходные данные интернет-магазина:

1. Результаты поиска: когда клиент выполняет поисковый запрос, интернет-магазин должен предоставить соответствующие результаты поиска, которые включают список товаров, соответствующих запросу клиента.
2. Подтверждение заказа: после оформления заказа, интернет-магазин должен предоставить клиенту подтверждение заказа, которое может включать информацию о заказе, выбранных товарах, их названии, описании и цене

**Задание 2.** Написание технического задания для реализации проекта (см. приложение 2)

|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 05.03.2024 – 06.03.2024 | Общие сведения (наименование проекта, заказчик, исполнитель) | 1 |
| Функциональные требования (требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.д) | 2 |
| Нефункциональные требования (безопасность, производительность, удобство использования и т.д.) | 1 |
| Требования к реализации (язык программирования, СУБД) | 1 |
| Требования к документации | 1 |

1. Общие сведения
   1. Наименование проекта: Разработка программного модуля для интернет магазина
   2. Заказчик: ООО "Кэтрин".
   3. Исполнитель: Компания "KПK".
2. Функциональные требования
   1. Возможность добавления заказов в каталог с указанием следующих параметров:

-Название товара;

-Описание товара;

-Стоимость товара;

-Изображение товара;

2.2. Возможность выбора товара из каталога:  
-Отображение списка товаров с их основными характеристиками (название, цена);

-Полное описание товара;

-Поиск товаров по запросу.

* 1. Возможность оформления заказа.

2.4. Отображение списка заказов.

3. Нефункциональные требования:

3.1. Кроссплатформенность:

- Поддержка работы на ОС семейства Windows.

3.2. Безопасность:

- Логин и пароль для доступа к аккаунту;

3.3. Удобство использования:

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.

3.4. Производительность:

- Приложение должно иметь быстрый доступ к данным;

- Минимальное время отклика на запросы пользователя.4. Требования к реализации

4.1. Язык программирования: C#

4.2. СУБД: Sql Server.

5. Требования к документации

5.1. Техническое задание на разработку программного модуля.

5.2. Руководство системному программисту.

**Задание 3.** Разработка алгоритма поставленной задачи и его реализация средствами автоматизированного проектирования.

|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 07.03.2024 – 09.03.2024 | Сформировать основной алгоритм в виде блок-схемы в соответствии с техническим заданием | 3 |
| Детализировать в виде алгоритма одну из функций | 3 |
| Алгоритмы представить одним из способов:  − Алгоритм в виде блок-схемы выполнить по правилам, установленным ГОСТ 19.701.  − Алгоритм в виде таблиц выполнить по правилам, установленным ГОСТ 2.105.  − Алгоритм в виде текстового описания выполнить по правилам, установленным ГОСТ 24.301. | 3  1  1 |

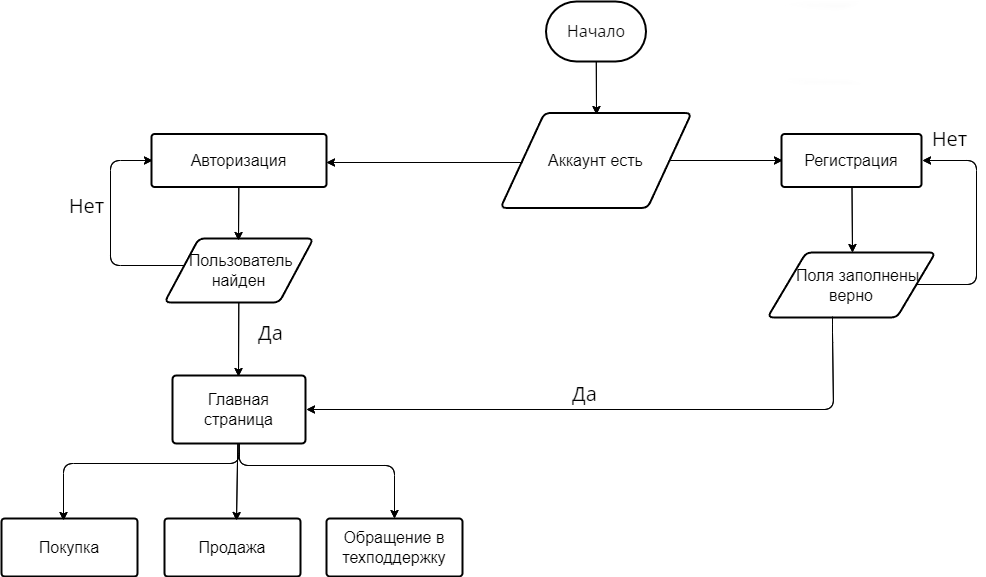


Рисунок 1 – Основной алгоритм в виде блок схемы



Рисунок 2 - Детализация функции «Описание авторизации пользователя»

**Задание 4.** Разработка интерфейса программного модуля по составленному алгоритму в среде разработки в соответствии технического задания (см. приложение 3).

|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 11.03.2024 – 13.03.2024 | Разработать руководство по стилю. Все компоненты системы должны иметь единый согласованный внешний вид | 2 |
| Для работы с разными сущностями используйте разные формы, где это уместно. | 2 |
| Необходимо использовать последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад») | 1 |
| Необходимо использовать соответствующий заголовок на каждом окне приложения | 1 |

Руководство по стилю

Визуальные компоненты должны соответствовать руководству по стилю, предоставленному в качестве ресурсов к заданию в соответствующем файле. Обеспечьте соблюдение требований всех компонентов в следующих областях:

1. цветовая схема,
2. размещение логотипа,
3. использование шрифтов,
4. установка иконки приложения.

Макет и технические характеристики

Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

1. разметка и дизайн (предпочтение отдается масштабируемой компоновке; должно присутствовать ограничение на минимальный размер окна; должна присутствовать возможность изменения размеров окна, где это необходимо; увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
2. группировка элементов (в логические категории);
3. использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);
4. расположение и выравнивание элементов (метки, поля для ввода и т.д.);
5. последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
6. общая компоновка логична, понятна и проста в использовании;
7. последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);
8. соответствующий заголовок на каждом окне приложения.

Руководство по стилю

**Использование логотипа**

Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом (в ресурсах). Не искажайте логотип (не изменяйте изображение, его пропорции, цвет). Также для приложений должна быть установлена иконка.

****

**Шрифт**

Используйте шрифт Open Sans.



Рисунок 4 – иконка приложения.

**Цветовая схема**

В качестве основного фона используется фиолетовый цвет; в качестве дополнительного белый.

Для акцентирования внимания пользователя на целевое действие интерфейса используйте серый цвет.

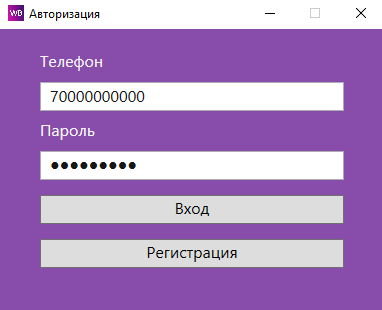


Рисунок 5 – Форма авторизации

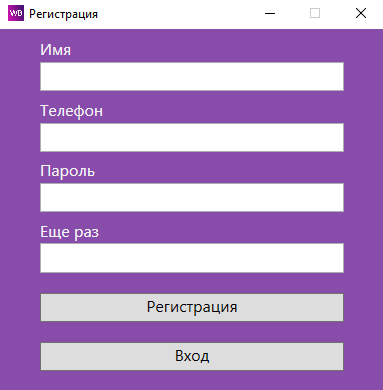
****

Рисунок 6 – Форма регистрации.

|  |  |  |
| --- | --- | --- |
| Основной фон | Дополнительный фон 2 | Акцентирование внимания |
| #884cab | #f5f5f5 | #d9d9d9 |
|  |  |  |

**Задание 5.** Разработка базы данных.

|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 14.03.2024 – 16.03.2024 | Спроектировать ER-диаграмму. Обязательна 3 нормальная форма с обеспечением ссылочной целостности. При разработке диаграммы обратите внимание на согласованную осмысленную схему именования, создайте необходимые первичные и внешние ключи, определите ограничения внешних ключей, отражающие характер предметной области. |  |
| ER - диаграмма должна быть представлена в формате удобном для просмотра и содержать таблицы, связи между ними, атрибуты и ключи (типами данных на данном этапе можно пренебречь) | 2 |
| Создайте все необходимые сущности, определите отношения, создайте ограничения на связи между сущностями (при наличии всех связей), приведите базу данных к 3НФ (при наличии всех сущностей и связей). | 1 |
| Заполните базу тестовым набором данных | 1 |

База данных

Была использована бд Sql Server, язык программирования: T-SQL. ERD-диаграмма на Рисунке 5. Было создано 3 таблицы: пользователи, вещи и сообщения.

Заполнение данными

В качестве вещей были добавлены 2 записи с информацией.

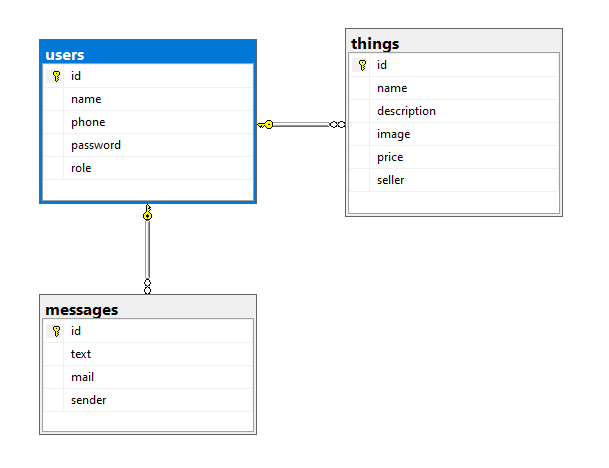


Рисунок 5 – Схема бд

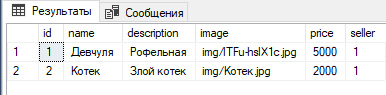


Рисунок 6 – Тестовые данные

**Задание 6.** Реализация алгоритма программного модуля.

|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 18.03.2024 – 21.03.2024 | В качестве источников данных для реализации алгоритмов используйте динамические списки или массивы в вашем коде, если не реализовывается БД | 2 |
| Выполнить исходный код модуля в соответствии гайдлайну: идентификаторы должны соответствовать соглашению об именовании – стилю CamelCase для C# | 1 |
| Допустимо использование не более одной команды в строке | 1 |
| Необходимо использовать комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода. Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения. | 1 |
| Реализовать программные обработки исключительных ситуаций в приложении. Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. | 3 |
| Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных. | 2 |

Подключение к БД

Был использован ЭнтитиФреймворк для подключения к БД.

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)  
{  
 optionsBuilder.UseSqlServer("Server=localhost;Trusted\_Connection=True;TrustServerCertificate=True;Database=polina;");  
}

Классы данных:

Вещь

public class thing  
{  
 [Key] public int id { get; set; }  
 [MaxLength(50)] public string name { get; set; } = null;  
 [MaxLength(50)] public string description { get; set; } = null;  
 [MaxLength(255)] public string image { get; set; } = null;  
 [ForeignKey(name: "seller")] public user user { get; set; } = null;  
 public double price { get; set; }  
 public int seller { get; set; }  
}

Пользователь

public class user  
{  
 [Key] public int id { get; set; }  
 [MaxLength(20)] public string name { get; set; } = null;  
 [StringLength(11)] public string phone { get; set; } = null;  
 [MaxLength(60)] public string password { get; set; } = null;  
 [StringLength(1)] public string role { get; set; } = null;   
}

Сообщение

public class message  
{  
 [Key] public int id { get; set; }  
 [MaxLength(100)] public string text { get; set; } = null;  
 [MaxLength(100)] public string mail { get; set; } = "sample@mail.ru";  
 public int sender { get; set; }  
 [ForeignKey(name: "sender")] public user user { get; set; } = null;  
}

Класс navigator для навигации по окнам

public static class navigator  
{  
 private static Dictionary<Type, Window> windows { get; set; } = [];  
  
 public static void show<T>() where T : Window, new()  
 {  
 var window = new T();  
 windows[typeof(T)] = window;  
 window.Closed += (\_, \_) => windows.Remove(typeof(T));  
 window.Show();  
 }  
  
 public static void close<T>() where T : Window  
 {  
 if (!windows.TryGetValue(typeof(T), out var window)) return;  
 window.Close();  
 windows.Remove(typeof(T));  
 }  
}

Класс h для отладки

public static class h  
{  
 public static void showEmpty(string message, string caption = "Сообщение")  
 {  
 System.Windows.MessageBox.Show(message, caption);  
 }  
  
 public static void showError(string message, string caption = "Ошибка")  
 {  
 System.Windows.MessageBox.Show(message, caption, System.Windows.MessageBoxButton.**OK**, System.Windows.MessageBoxImage.**Error**);  
 }  
   
 public static System.Windows.MessageBoxResult showConfirm(string message, string caption = "Подтверждение")  
 {  
 return System.Windows.MessageBox.Show(message, caption,  
 System.Windows.MessageBoxButton.**YesNo**,  
 System.Windows.MessageBoxImage.**Question**);  
 }  
  
 public static void debug(object value)  
 {  
 System.Diagnostics.Debug.WriteLine(value.ToString());  
 }  
  
 public static void consoleLog(object value)  
 {  
 Console.WriteLine(value.ToString());  
 }  
}

Класс App был доработан

public static user User { get; set; } = null;  
public static AppDbContext AppDbContext { get; } = new();  
  
protected override void OnStartup(System.Windows.StartupEventArgs e)  
{  
 navigator.show<InWindow>();  
}  
  
protected override void OnExit(System.Windows.ExitEventArgs e)  
{  
 AppDbContext.SaveChanges();  
 AppDbContext.Dispose();  
}

Авторизация происходит с помощью попытки получить первый попавшийся элемент из набора пользователей

try  
{  
 App.User = App.AppDbContext.users.First(u => u.phone == phone && u.password == password);  
 navigator.show<MainWindow>();  
 navigator.close<InWindow>();  
}  
catch (Exception ex)  
{  
 h.showError("Неверный логин или пароль",  
 "Ошибка авторизации");  
 h.consoleLog(ex);  
 h.debug(ex);  
}

Регистрация происходит через добавление пользователя в набор

try  
{  
 var user = new user  
 {  
 name = name, phone = phone, password = password, role = "u"  
 };  
 App.AppDbContext.users.Add(user);  
 App.AppDbContext.SaveChanges();  
   
 h.showEmpty("Вы зарегистрировались!", "Успех");  
 App.User = App.AppDbContext.users.First(u => u.phone == phone);  
 navigator.show<MainWindow>();  
 Close();  
}  
catch (Exception ex)  
{  
 h.showError(ex.Message);  
 h.consoleLog(ex);  
 h.debug(ex);  
}

Поиск товара

try  
{  
 clothes = [.. App.AppDbContext.things.Where(  
 c => EF.Functions.Like(string.Concat(c.name, c.description), $"%{query}%")  
 )];  
 setListView();  
}  
catch (Exception ex)  
{  
 h.showError($"Товары по запросу '{query}' не найдены",  
 "Ошибка поиска");  
 h.debug(ex);  
 h.consoleLog(ex);  
}

Покупка товара

try  
{  
 if (h.showConfirm("Уверен?") == MessageBoxResult.**Yes**)  
 {  
 Microsoft.Win32.SaveFileDialog sfd = new()  
 {  
 FileName = thing.name,  
 Filter = "Images (\*.jpg;\*.png)|\*.jpg;\*.png"  
 };  
   
 if (sfd.ShowDialog() == true)  
 {  
 System.IO.File.Copy(thing.image, sfd.FileName, true);  
 h.showEmpty($"Вы купили: {thing.toString()}");  
 }  
 }  
}  
catch (Exception ex)  
{  
 h.showError($"Не удалось купить товар {thing.toString()}",  
 "Ошибка покупки");  
 h.debug(ex);  
 h.consoleLog(ex);  
}

Продажа товара

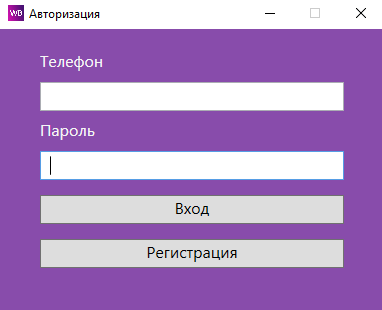
var clothName = ClothName.Text;  
var clothDesc = ClothDesc.Text;  
var clothImage = ClothImage.Text;  
  
if (!double.TryParse(ClothPrice.Text, out var clothPrice))  
{  
 throw new Exception($"Не удалось сконвертировать {ClothPrice.Text} в число");  
}  
  
if (clothName.Length == 0 || clothDesc.Length == 0 || clothImage.Length == 0)  
{  
 throw new Exception("Заполни все поля");  
}  
  
System.IO.File.Copy(fullPath, "img/" + ClothImage.Text, true);  
  
try  
{  
 if (h.showConfirm("Добавить?") == System.Windows.MessageBoxResult.**Yes**)  
 {  
 App.AppDbContext.things.Add(new thing  
 {  
 name = clothName,  
 description = clothDesc,  
 image = "img/" + clothImage,  
 price = clothPrice,  
 seller = App.User.id  
 });  
 App.AppDbContext.SaveChanges();  
 h.showEmpty("Вещь успешно добавлена!", "Успех");  
 }  
}  
catch (Exception ex)  
{  
 throw new Exception("Не удалось добавить товар", ex);  
}

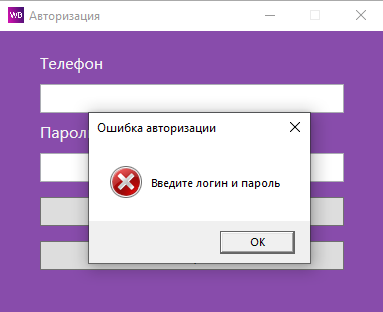
**Задание 7.** Отладка программного модуля.

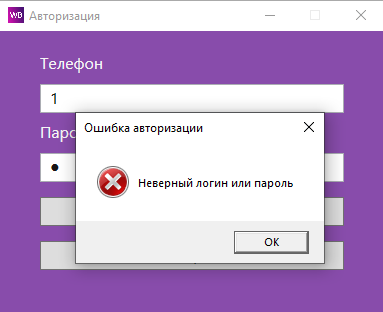
|  |  |  |
| --- | --- | --- |
| Дата | Порядок выполнения | Баллы |
| 22.03.2024 – 23.03.2024 | Выполнить отладку программного обеспечения с использованием инструментальных средств. Сохранить и представить результаты в скриншотах | 1 |
| Определить наборы входных данных и выполнить функциональное тестирование модуля по определенному сценарию. | 1 |
| Провести тестирование для проверки функциональности программы (хотя бы 1 тест на 1 функцию). | 1 |
| Использовать инструментальные средства для тестирования. | 1 |
| Представить результаты тестирования в виде протокола тестирования, в соответствии со стандартами | 1 |

Отладка

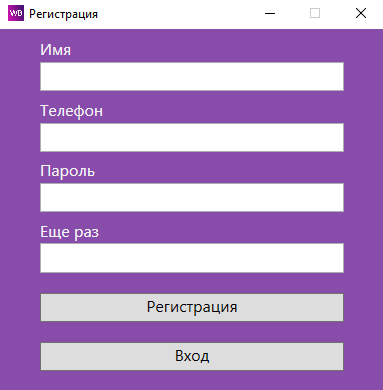
Авторизация

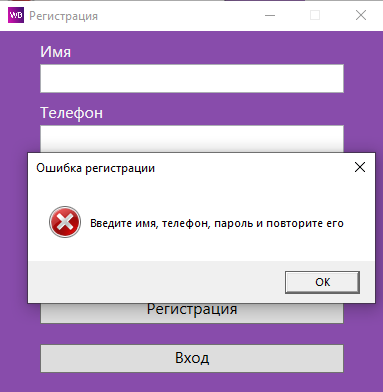


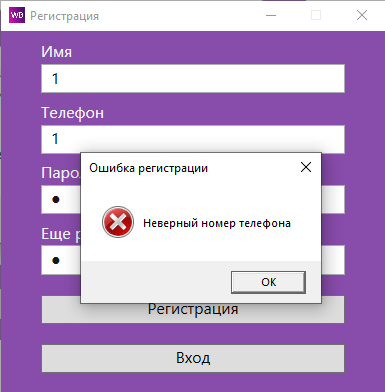




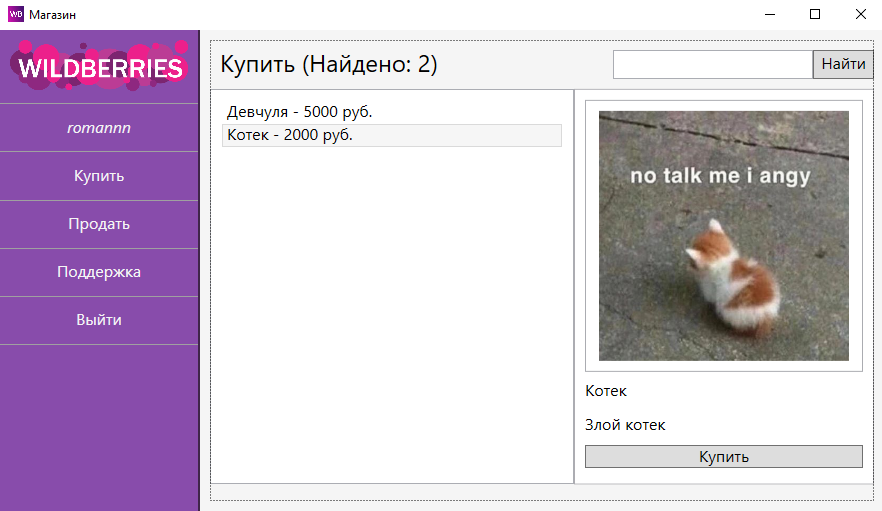
Регистрация

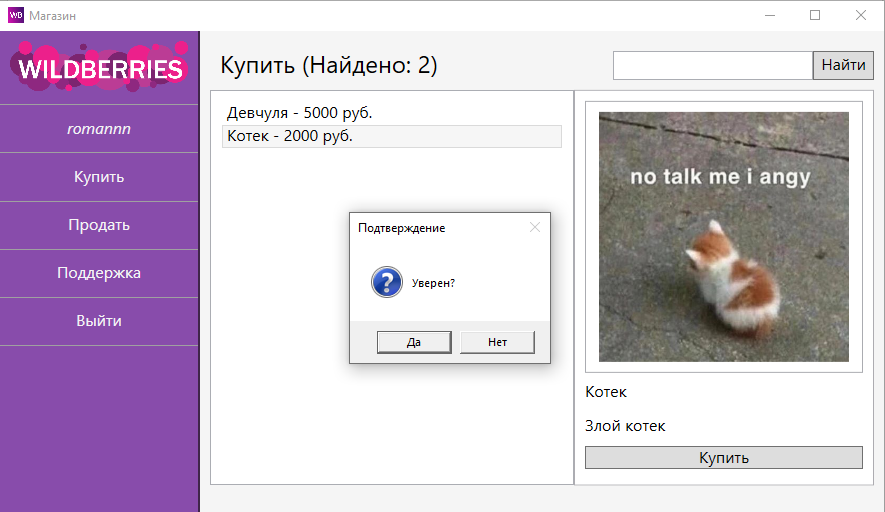
****

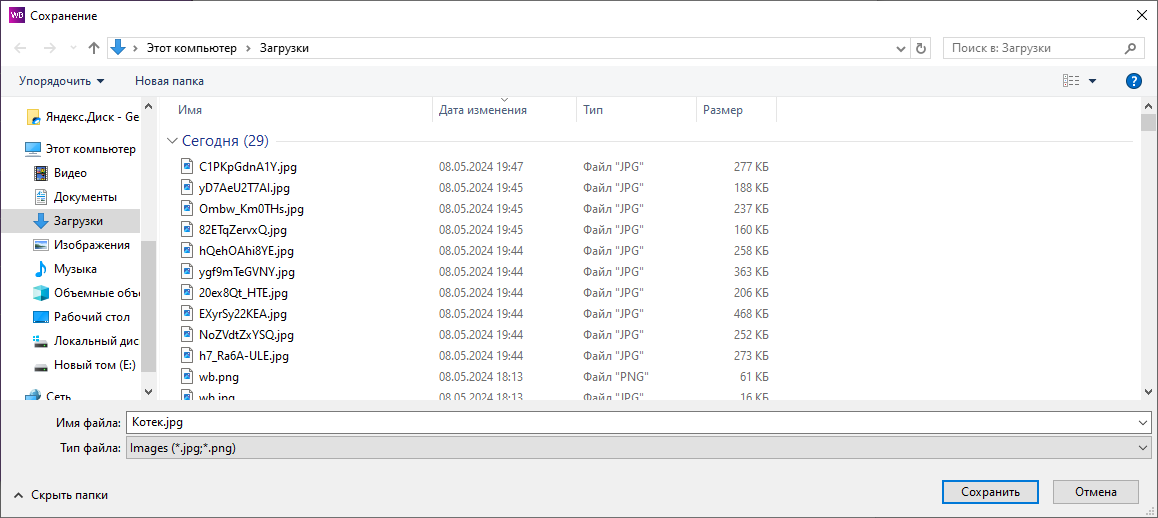


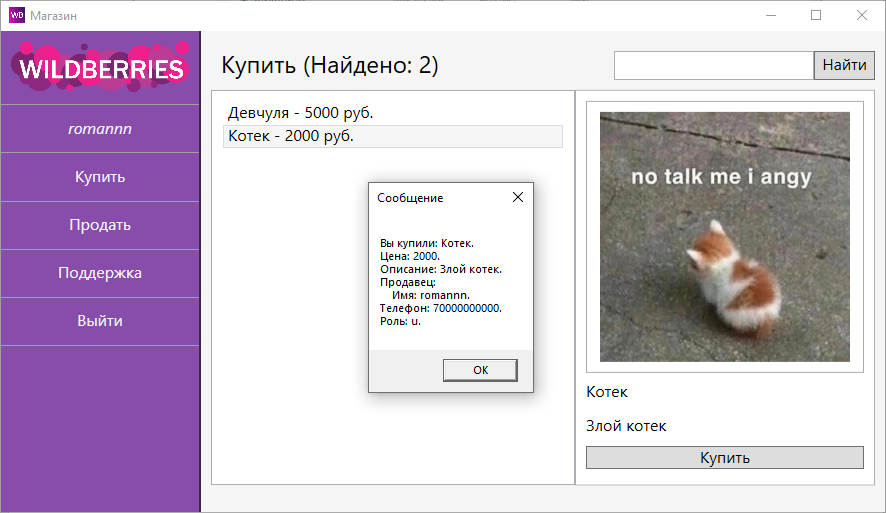


Покупка вещи

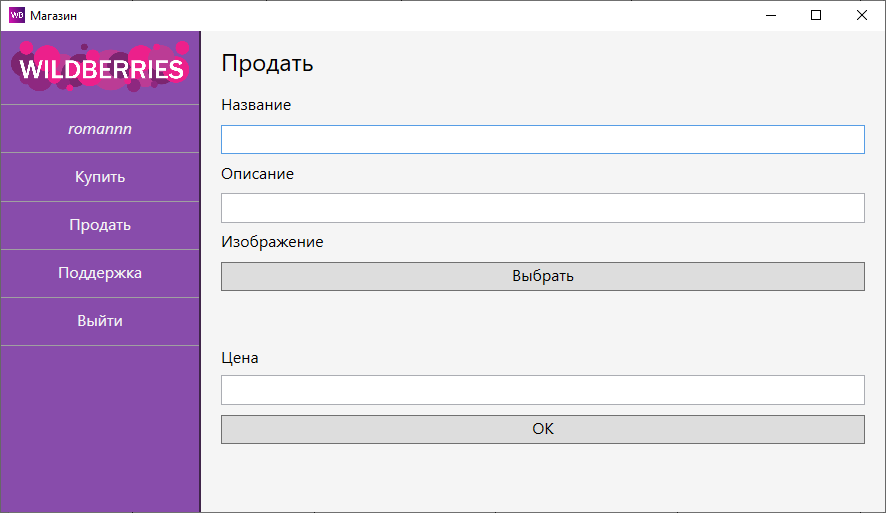


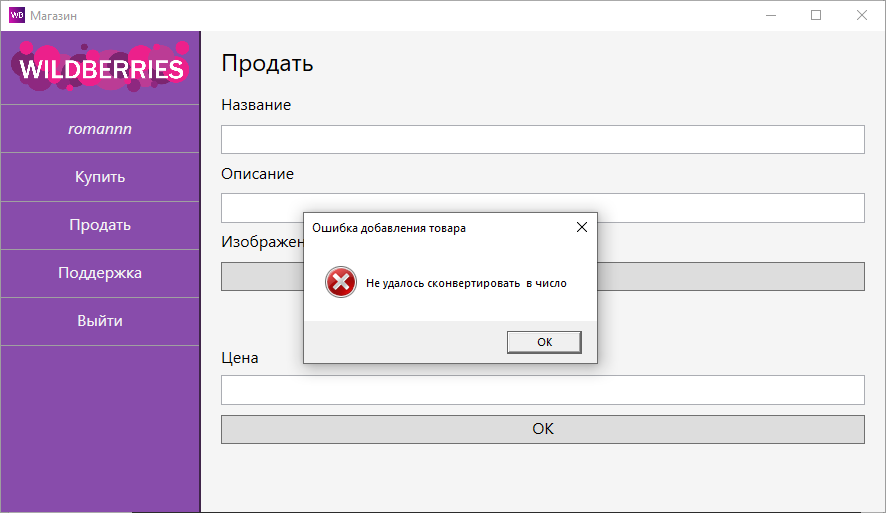


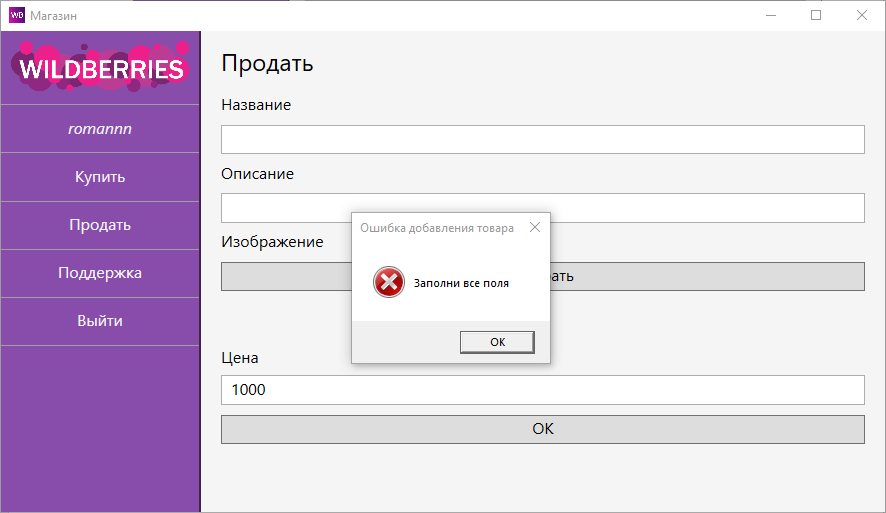


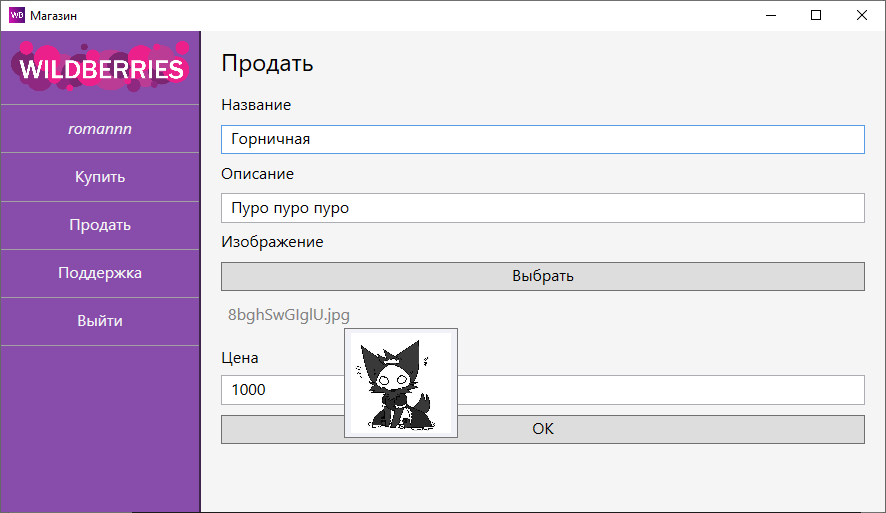


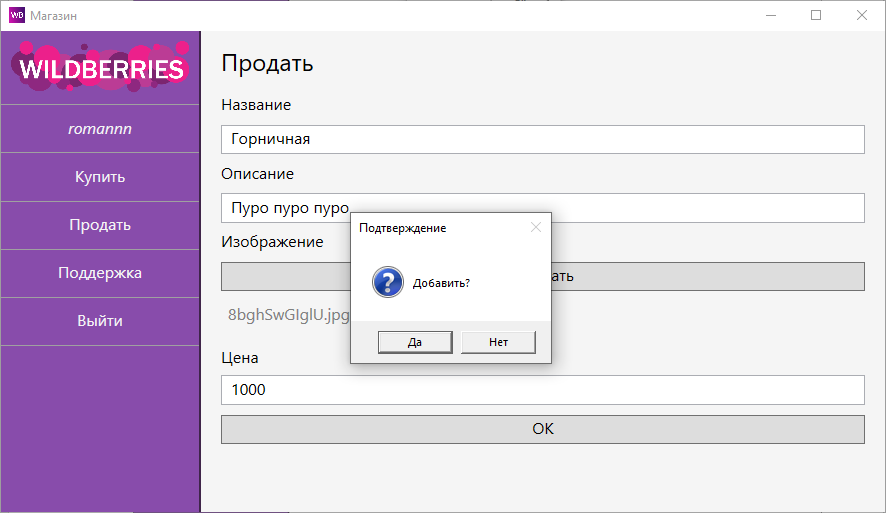
Добавление вещи

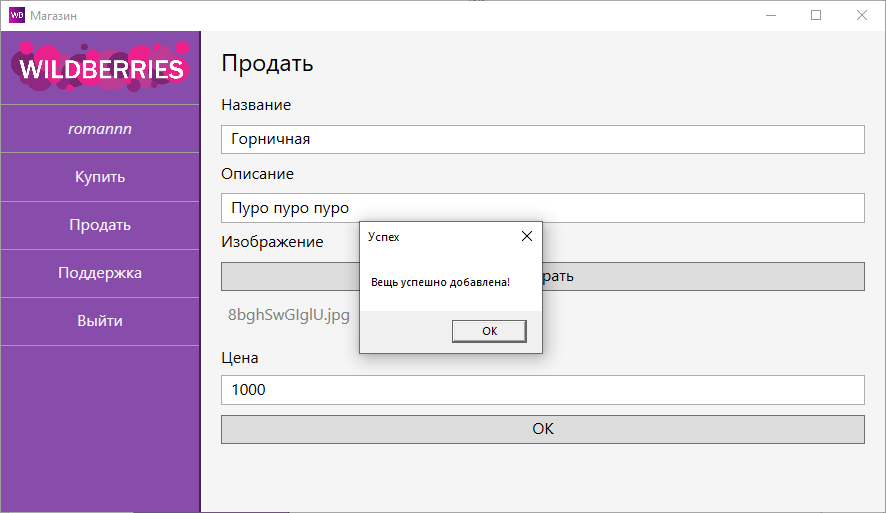


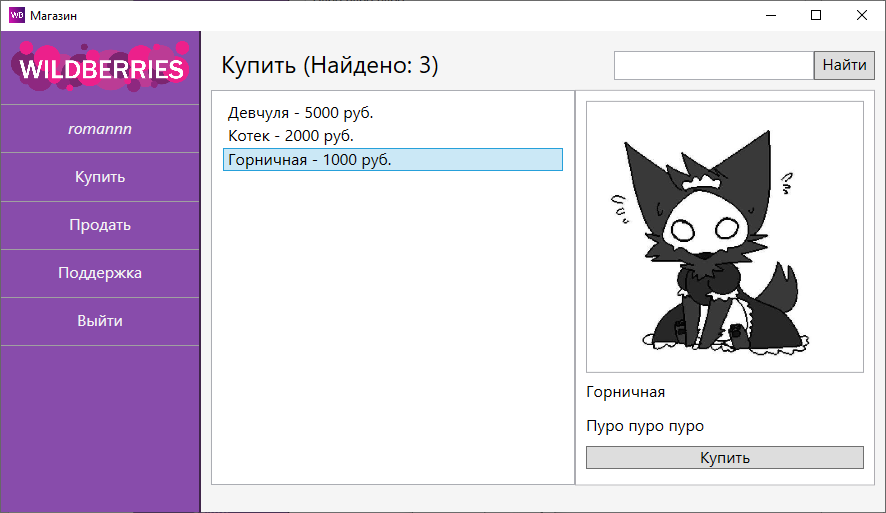




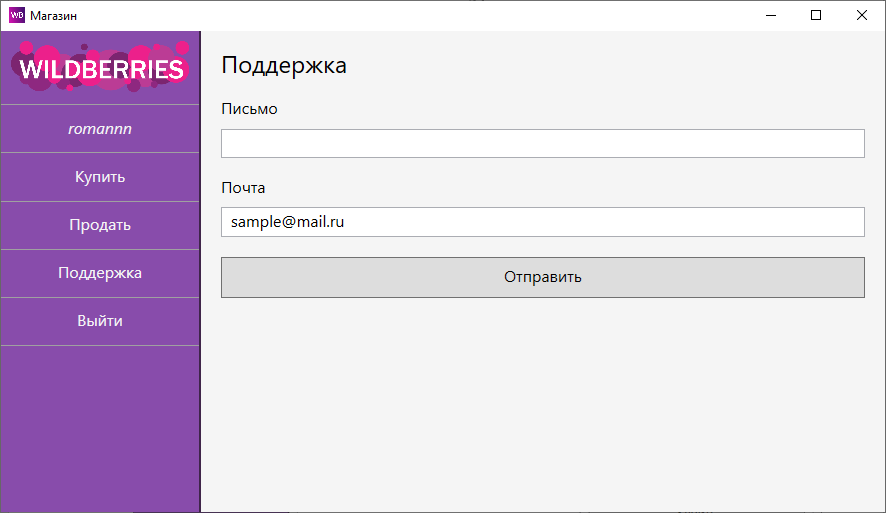


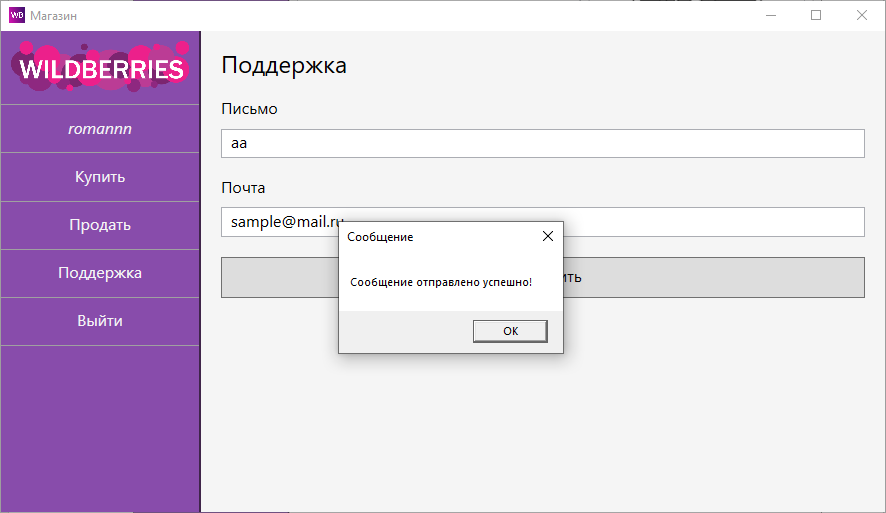




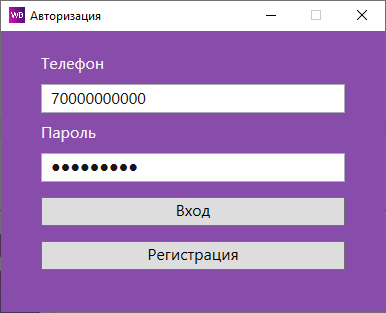


Отправка сообщения





Выход



Тестирование

|  |  |  |  |
| --- | --- | --- | --- |
| Действие | Сценарий | Ожидаемый результат | Действительный результат |
| Авторизация | Пустые поля | Ошибка | Совпадает с ожидаемым результатом |
| Неверные данные | Пользователь не найден | Совпадает с ожидаемым результатом |
| Верные данные | Открытие главного окна | Совпадает с ожидаемым результатом |
| Регистрация | Пустые поля | Ошибка | Совпадает с ожидаемым результатом |
| Неверный телефон | Ошибка | Совпадает с ожидаемым результатом |
| Все данные верны (поля заполнены и телефон верный) | Открытие главного окна | Совпадает с ожидаемым результатом |
| Покупка | Покупка товара | Покупка товара | Совпадает с ожидаемым результатом |
| Продажа | Пустые поля | Ошибка | Совпадает с ожидаемым результатом |
| Поля заполнены и изображение выбрано | Добавление товара в каталог | Совпадает с ожидаемым результатом |
| Отправка сообщения | Отправка сообщения | Сообщение отправлено | Совпадает с ожидаемым результатом |
| Выход | Выход из аккаунта | Открытие окна авторизации | Совпадает с ожидаемым результатом |

|  |  |  |  |
| --- | --- | --- | --- |
| № задания | Max балл | Рейтинг, % | Рейтинг, баллы |
| 1 | 5 | 61% - 75% – 3 (уд)  76% - 90% – 4 (хор)  91% -100% - 5 (отл) | 25 – 31 – 3 (уд)  32 – 37 – 4 (хор)  38 – 42 – 5 (отл) |
| 2 | 6 |
| 3 | 6 |
| 4 | 6 |
| 5 | 4 |
| 6 | 10 |
| 7 | 5 |
| **Итого** | **42** |
| Практика, сданная не в срок: – 6 баллов | |

|  |  |  |  |
| --- | --- | --- | --- |
| Итого | | | |
| Баллы | Оценка | Дата | Подпись |
|  |  |  |  |