**Многопользовательское приложение**

**Задание 1**

**АРМ работника склада**

**Хранение данных:**

* Данные о товарах, поставках и пользователях хранятся в warehouse.json.
* При изменении информации (например, поступление нового товара) файл обновляется.

**Локальная регистрация и авторизация:**

* Пользователь входит в систему, сверяя логин и хеш пароля.
* В файле users.json хранятся данные сотрудников.

**Обмен данными:**

* Сообщения о новых поступлениях отправляются через **Memory-Mapped Files**.

Чат для сотрудников склада через **Named Pipes**.

**Код**

namespace Task1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private readonly DataService \_dataService;

private readonly AuthService \_authService;

private readonly CommunicationService \_communicationService;

private User \_currentUser;

public MainWindow()

{

InitializeComponent();

\_dataService = new DataService();

\_authService = new AuthService(\_dataService);

\_communicationService = new CommunicationService();

\_communicationService.NotificationReceived += OnNotificationReceived;

\_communicationService.ChatMessageReceived += OnChatMessageReceived;

// Создаем начальные данные, если их нет

InitializeDefaultData();

LoadProducts();

}

private void InitializeDefaultData()

{

// Создаем администратора по умолчанию, если файл пользователей пуст

var users = \_dataService.LoadUsers();

if (users.Count == 0)

{

users.Add(new User

{

Id = 1,

Username = "admin",

PasswordHash = \_authService.HashPassword("admin"),

FullName = "Администратор",

Role = "Admin",

LastLogin = DateTime.Now

});

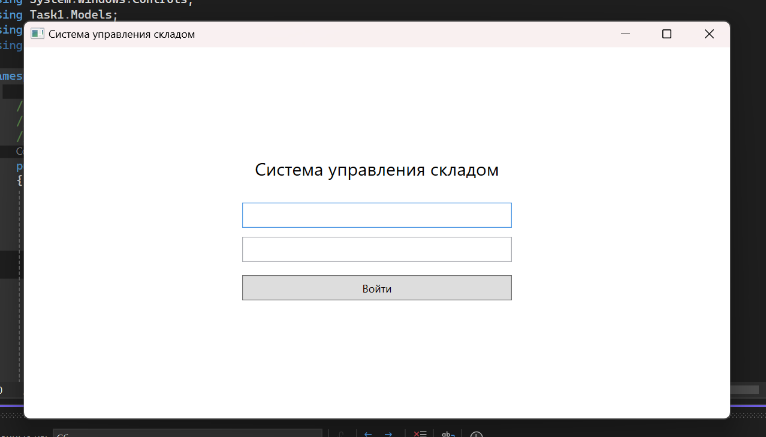
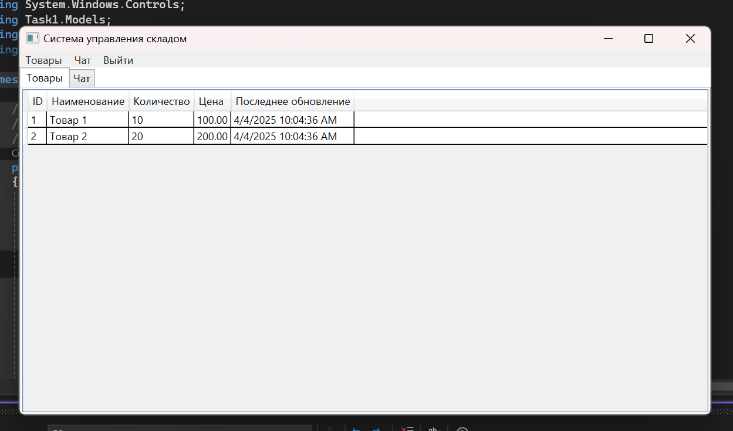
\_dataService.SaveUsers(users);

}

**Таблица 17 – Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **admin**  **admin** | **товар1**  **товар** |

**Анализ результата**

****