Правительство Санкт-Петербурга

Комитет по науке и высшей школе

**Санкт-Петербургское государственное бюджетное**

**профессиональное образовательное учреждение**

**«Политехнический колледж городского хозяйства»**

**КУРСОВОЙ ПРОЕКТ**

Разработка локального WinForms приложения «GameEnergy»

Пояснительная записка

Листов 41

**СОДЕРЖАНИЕ**

[Введение 2](#_Toc1675836459)

[1 Анализ предметной области 3](#_Toc845366941)

[1.1 Описание предметной области 4](#_Toc2082228637)

[1.2 Обзор аналогов 5](#_Toc1742957274)

[1.3 Требования к разрабатываемой ИС 6](#_Toc1292835863)

[1.4 Обоснование выбора стека технологий 7](#_Toc14334381)

[2 Проектирование 9](#_Toc1759432270)

[2.1 Проектирование системы 10](#_Toc1027394208)

[2.2 Разработка модели базы данных 12](#_Toc290867984)

[2.3 Проектирование интерфейсов 14](#_Toc2032465828)

[3 Реализация 18](#_Toc1082396506)

[3.1 Реализация основных функций 19](#_Toc669837063)

[3.2 Реализация интерфейсов 22](#_Toc1188699960)

[3.3 Тестирование 23](#_Toc1005568689)

[4 Руководство администратора/пользователя 24](#_Toc1999669649)

[4.1 Руководство для администратора 25](#_Toc1905885667)

[4.2 Руководство для пользователя (покупателя) 26](#_Toc577377978)

[5 Мероприятия по информационной безопасности 28](#_Toc2029722468)

[5.1 Определение угроз при эксплуатации приложения «GameEnergy» 29](#_Toc1558576184)

[5.2 Что сделано как разработчиком для предотвращения этих угроз 29](#_Toc153553753)

[5.3 Рекомендации пользователям по безопасной работе 31](#_Toc1533018416)

[Заключение 31](#_Toc1930091579)

[Список источников 33](#_Toc1558064271)

[Приложение А 35](#_Toc1713619098)

[Приложение Б 37](#_Toc635046577)

## Введение

Современные цифровые технологии активно проникают во все сферы жизни. Одним из наиболее востребованных сегментов является рынок цифровых игр, где пользователи предпочитают приобретать лицензионные ключи активации через специализированные платформы. Однако большинство подобных решений являются веб- или облачными сервисами, требующими постоянного подключения к интернету, что ограничивает их использование в условиях нестабильного сетевого соединения или в локальных сетях без доступа во внешнюю сеть.

В связи с этим актуальной задачей становится разработка локального десктопного приложения, позволяющего пользователям безопасно и удобно приобретать игровые ключи без зависимости от онлайн-сервисов. Такое решение особенно востребовано в образовательных учреждениях, корпоративных средах или домашних условиях, где требуется изолированная, надёжная и функциональная система цифровой коммерции.

Целью данного дипломного проекта является разработка локального приложения «GameEnergy» на платформе C# WinForms, обеспечивающего полный цикл взаимодействия пользователя с каталогом игр: от поиска и фильтрации до оформления заказа, получения ключа активации и оставления отзывов. Приложение поддерживает двухуровневую систему ролей (пользователь и администратор), работу с локальной базой данных MS SQL Server, а также реализует современные подходы к проектированию — включая паттерны MVC и Repository — для обеспечения читаемости и расширяемости кода.

Проект разработан в рамках подготовки к демонстрационному экзамену по компетенции «Разработка приложений на платформе .NET с использованием WinForms», что подчёркивает его практическую направленность и соответствие профессиональным стандартам.

## Анализ предметной области

## 1.1 Описание предметной области

Современный рынок цифровых развлечений демонстрирует устойчивый рост, особенно в сегменте компьютерных игр. Покупка лицензионных ключей активации стала стандартом для миллионов пользователей, однако большинство существующих платформ (таких как Steam, Epic Games Store, GOG и др.) представляют собой онлайн-сервисы, требующие постоянного подключения к интернету и привязки к облачным аккаунтам. Это создаёт определённые ограничения для использования в замкнутых сетях, учебных заведениях, а также в условиях нестабильного или отсутствующего интернет-соединения.

В связи с этим возникает потребность в локальном решении, позволяющем пользователям безопасно и удобно приобретать игровые ключи без зависимости от внешних серверов. Такое приложение особенно актуально в образовательной и корпоративной среде, где важны автономность, контроль над данными и защита от внешних угроз.

Приложение GameEnergy разрабатывается как локальный цифровой магазин игр, работающий на базе C# WinForms с использованием локальной базы данных MS SQL Server. Оно предназначено для двух групп пользователей:

1. Покупатели (обычные пользователи) — могут:
   1. просматривать каталог игр с возможностью поиска, фильтрации и сортировки;
   2. добавлять игры в избранное или корзину;
   3. оформлять заказы и получать ключи активации;
   4. оценивать игры по 5-балльной шкале и оставлять текстовые отзывы;
   5. редактировать свой профиль (включая аватарку);
   6. просматривать профили других пользователей;
   7. читать системные уведомления от администратора.
2. Администраторы — обладают расширенными правами:
   1. добавлять, редактировать и удалять игры;
   2. загружать обложки и указывать коды YouTube-трейлеров;
   3. публиковать системные уведомления, видимые всем пользователям;
   4. управлять каталогом (жанры, разработчики и т.д.).

Таким образом, GameEnergy представляет собой автономную, многофункциональную систему цифровой коммерции, ориентированную на удобство, безопасность и независимость от внешних ресурсов.

## 1.2 Обзор аналогов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Платформа** | **Тип** | **Интернет-зависимость** | **Возможность локального использования** | **Ключевые функции** |
| **Steam** | Онлайн-магазин | Полная | Нет | Каталог, корзина, отзывы, профиль, облачные сохранения |
| **GOG Galaxy** | Онлайн + клиент | Частичная | Только с авторизацией | DRM-free игры, офлайн-режим ограничен |
| **Epic Games Store** | Онлайн-магазин | Полная | Нет | Бесплатные игры, эксклюзивы |
| **GameEnergy (наш проект)** | **Локальное WinForms-приложение** | **Минимальная** (только восстановление пароля и трейлеры) | **Полностью локально** | Каталог, фильтрация, корзина, избранное, отзывы, уведомления, роли пользователей |

Преимущества GameEnergy перед аналогами:

* Полная автономность (данные хранятся локально);
* Не требует облачного аккаунта;
* Поддержка двухуровневой системы ролей «из коробки»;
* Все изображения и ключи хранятся в локальной БД;
* Возможность развёртывания в закрытых сетях (например, в колледже или компании);
* Простота установки и обслуживания без веб-сервера.

## 1.3 Требования к разрабатываемой ИС

### 1.3.1 Функциональные требования

Приложение GameEnergy должно обеспечивать следующие функции:

1. Регистрация и авторизация пользователей с поддержкой двух ролей: пользователь и администратор.
2. Восстановление пароля через email (с отправкой хешированного кода подтверждения).
3. Просмотр каталога игр с возможностью:
   1. поиска по названию или разработчику;
   2. фильтрации по жанру, цене, разработчику;
   3. сортировки по: популярности, цене (возрастание/убывание), рейтингу, алфавиту, размеру скидки, дате выхода, дате добавления.
4. Просмотр карточки игры, включающей:
   1. название, разработчика, дату выхода;
   2. описание, обложку, трейлер (YouTube-код);
   3. текущую и старую цену, размер скидки;
   4. отзывы и оценки других пользователей.
5. Добавление игры в корзину или избранное.
6. Оформление заказа и автоматическое отображение купленной игры с ключом активации в разделе «Мои заказы».
7. Оценка и отзыв к игре (по 5-балльной шкале + текст).
8. Жалоба на игру по трём категориям: неверные данные, дубль, другое.
9. Управление профилем:
   1. изменение ФИО, аватара, email, пароля;
   2. просмотр профилей других пользователей.
10. Просмотр системных уведомлений, созданных администратором.
11. Функции администратора:
    1. добавление/редактирование/удаление игр;
    2. загрузка изображений и указание YouTube-кода трейлера;
    3. публикация уведомлений.

Необходимо реализовать:

1. Современный и интуитивно понятный дизайн с использованием библиотек Guna UI2 и MaterialSkin.
2. Единый стиль оформления всех форм и контролов.
3. Поддержка кастомных пользовательских контролов:
   1. MainGameControl — отображение игры в каталоге;
   2. GameSlideControl — карусель новинок;
   3. NavigationControl — панель навигации;
   4. UserCommentsControl — отображение отзывов;
   5. CartAndOrderItemsControl, OrderControl — элементы корзины и заказов.
4. Отзывчивость интерфейса — плавная навигация без «подвисаний» даже при большом количестве записей.
5. Визуальная обратная связь (анимации, иконки, подсветка активных элементов).

## 1.4 Обоснование выбора стека технологий

1. Язык программирования: C#

Выбор обусловлен следующими факторами:

* 1. C# является стандартом для разработки Windows-приложений;
  2. мощная экосистема .NET и Visual Studio;
  3. высокая производительность и типобезопасность;
  4. нативная поддержка WinForms и Entity Framework.

1. Фреймворк: Windows Forms (WinForms)
   1. Простота разработки десктопных приложений с графическим интерфейсом;
   2. Подходит для локальных решений без веб-интерфейса;
   3. Богатая библиотека контролов (особенно с Guna UI2);
   4. Хорошо документирован и поддерживается Microsoft.
2. СУБД: Microsoft SQL Server (локальный экземпляр)
   1. Надёжная реляционная СУБД с полной поддержкой транзакций;
   2. Лёгкая интеграция с Entity Framework Core;
   3. Хранение BLOB-данных (изображений) прямо в таблицах;
   4. Подходит для локального развёртывания (SQL Server Express или LocalDB).
3. ORM: Entity Framework Core
   1. Упрощает работу с БД через объектную модель;
   2. Поддержка паттерна Repository;
   3. Автоматическая генерация SQL-запросов с защитой от инъекций.
4. Безопасность: BCrypt.Net-Next
   1. Надёжное хеширование паролей и кодов восстановления;
   2. Защита даже в случае утечки БД.
5. Дополнительные библиотеки:
   1. Guna UI2 и MaterialSkin — для современного внешнего вида;
   2. Newtonsoft.Json и DnsClient — для отправки кода восстановления на email через SMTP-подобный сервис на виртуальной машине.

**Примечание:** Приложение работает локально. Единственная функция, требующая интернета — **восстановление пароля** (отправка кода на email) и **воспроизведение трейлера** (загрузка превью по YouTube-ссылке). При отсутствии сети трейлер заменяется заглушкой, а восстановление пароля недоступно — что не критично для основного функционала.

## Проектирование

## 2.1 Проектирование системы

### 2.1.1 Определение группы пользователей

В приложении GameEnergy выделены две основные группы пользователей:

1. **Обычный пользователь** (покупатель)

* проходит регистрацию и авторизацию;
* может просматривать каталог игр, искать, фильтровать и сортировать их;
* добавляет игры в избранное или корзину;
* оформляет заказ и получает ключ активации;
* оставляет оценки и текстовые отзывы;
* может подать жалобу на игру;
* редактирует свой профиль (ФИО, аватар, email, пароль);
* просматривает профили других пользователей;
* читает системные уведомления от администратора.

1. **Администратор**

* обладает всеми правами обычного пользователя;
* может добавлять, редактировать и удалять игры;
* загружает изображения обложек и указывает YouTube-код трейлера;
* создаёт системные уведомления, видимые всем пользователям;
* управляет справочниками (жанры, разработчики и др., если реализовано).

При первом запуске приложения администратор создаётся вручную (например, через начальную миграцию или seed-данные), либо первый зарегистрированный пользователь назначается администратором.

### 2.1.2 Функциональное моделирование

Для описания взаимодействия пользователей с системой разработана диаграмма прецедентов (в текстовом виде, так как изображения будут в Приложении).

**Акторы:**

* Пользователь
* Администратор

**Основные прецеденты:**

|  |  |
| --- | --- |
| **Актор** | **Прецедент** |
| Пользователь | Регистрация |
| Пользователь | Авторизация |
| Пользователь | Восстановление пароля |
| Пользователь | Поиск игры по названию или разработчику |
| Пользователь | Фильтрация по жанру, цене, разработчику |
| Пользователь | Сортировка по цене, рейтингу, дате и др. |
| Пользователь | Просмотр карточки игры |
| Пользователь | Добавление в избранное / корзину |
| Пользователь | Оформление заказа |
| Пользователь | Получение ключа активации |
| Пользователь | Оценка игры и написание отзыва |
| Пользователь | Жалоба на игру |
| Пользователь | Редактирование профиля |
| Пользователь | Просмотр профилей других |
| Пользователь | Просмотр уведомлений |
| Администратор | Добавление/редактирование/удаление игры |
| Администратор | Загрузка изображения обложки |
| Администратор | Указание YouTube-кода трейлера |
| Администратор | Создание системного уведомления |

**User Flow (описательно):**

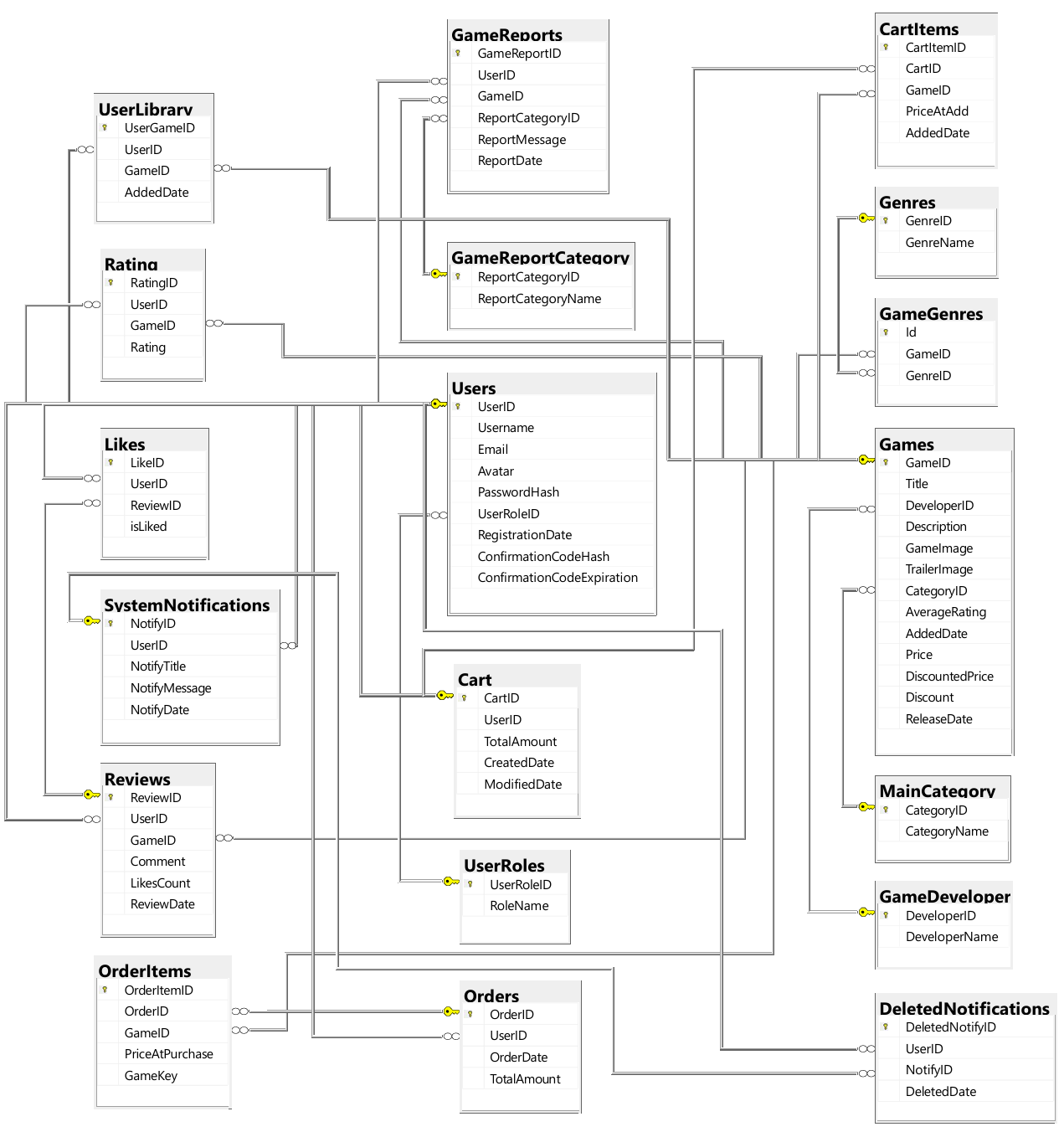
* **Пользователь**: регистрация → авторизация → просмотр каталога → фильтрация → выбор игры → добавление в корзину → оформление заказа → получение ключа → оставление отзыва.
* **Администратор**: вход → каталог → добавление новой игры → загрузка обложки → публикация → создание уведомления.

## 2.2 Разработка модели базы данных

База данных реализована на основе MS SQL Server и спроектирована с использованием реляционной модели, структура которой представлена в виде ER-диаграммы (см. Рисунок 2). Все данные хранятся локально. Аватары пользователей сохраняются в виде **BLOB (varbinary(max))** внутри таблиц.

* **Основные таблицы и их назначение:**

1. **Users:** Основная таблица, хранящая общую информацию о пользователях (логин, email, хеш пароля, дата регистрации и т.д.).
2. **UserRoles:** Хранит роли пользователей (например, «пользователь», «администратор» и т.п.).
3. **Games:** Хранит информацию об играх (название, описание, разработчик, дата выпуска, цена, изображение, трейлер и т.д.).
4. **GameDevelopers:** Хранит информацию о разработчиках игр (название, сайт и другая контактная информация).
5. **Genres:** Хранит список игровых жанров (например, «шутер», «RPG» и т.д.).
6. **GameGenres:** Связующая таблица между играми и жанрами (многие-ко-многим).
7. **Reviews:** Хранит отзывы пользователей на игры (текст отзыва, дата, оценка, количество лайков).
8. **Likes:** Фиксирует, какие отзывы лайкнул пользователь.
9. **UserLibrary:** Хранит список игр, добавленных пользователем в свою библиотеку.
10. **Cart:** Хранит информацию о корзинах пользователей (общая сумма, дата создания и т.д.).
11. **CartItems:** Содержит игры, добавленные в корзину (с ценой на момент добавления).
12. **Orders:** Хранит информацию о заказах (дата, итоговая сумма, пользователь).
13. **OrderItems:** Детализация заказа — какие игры были куплены, по какой цене, с ключом активации.
14. **GameReports:** Хранит жалобы или сообщения пользователей о проблемах с играми (категория, текст сообщения, дата).
15. **GameReportCategory:** Справочник категорий отчётов (например, «баг», «нарушение контента» и т.д.).
16. **SystemNotifications:** Хранит системные уведомления, отправляемые пользователям (заголовок, сообщение, дата).
17. **Rating:** Хранит оценку игры отправленную пользователем (цифра от 1 до 5)
18. **MainCategory:** Хранит название категории для отображения игры на главной форме (например, «Хит продаж», «Популярное» и т.д.).

*Рисунок 1 - Диаграмма базы данных*

## 2.3 Проектирование интерфейсов

Пользовательский интерфейс разработан с учётом принципов **современного UX/UI** для десктопных приложений:

* Использованы библиотеки **Guna UI2** и **MaterialSkin** для создания визуально привлекательного и согласованного дизайна.
* Все формы имеют единый стиль: цветовая палитра, шрифты, отступы, анимации.
* Интерфейс не перегружен элементами: акцент сделан на контент (игры, обложки, отзывы).

1. **Главная форма (MainForm)**
   1. Отображает игры по категориям;
   2. содержит GameSlideControl — карусель новинок или популярных игр.
   3. использует кастомный контрол MainGameControl для отображения игр по категориям;
2. **Каталог игр (CatalogForm)**
   1. отображает каталог игр с панелью фильтров и поиска;
   2. использует кастомный контрол MainGameControl для отображения каждой игры;
3. **Форма игры (GameDetailsForm)**
   1. отображает полную информацию об игре;
   2. включает вкладки: «Обзор», «Отзывы», «Трейлер»;
   3. использует UserCommentsControl для списка отзывов.
4. **Корзина и заказы**
   1. CartForm: редактирование состава корзины;
   2. OrdersForm: история заказов с отображением ключей активации;
   3. используются контролы CartAndOrderItemsControl и OrderControl.
5. **Профиль пользователя (ProfileForm)**
   1. редактирование данных, смена аватара;
   2. просмотр профилей других — через отдельную модальную форму.
6. **Админ-панель (AdminPanelForm)**
   1. форма добавления/редактирования игры;
   2. загрузка изображения через OpenFileDialog;
   3. поле ввода для YouTube-кода трейлера;
   4. форма создания уведомления.
7. **Навигация**
   1. реализована через NavigationControl — боковая или верхняя панель с кнопками перехода между разделами (Каталог, Избранное, Корзина, Профиль, Уведомления и т.д.).
8. **Системные диалоги**
   1. авторизация, регистрация, восстановление пароля — вынесены в отдельные модальные формы.

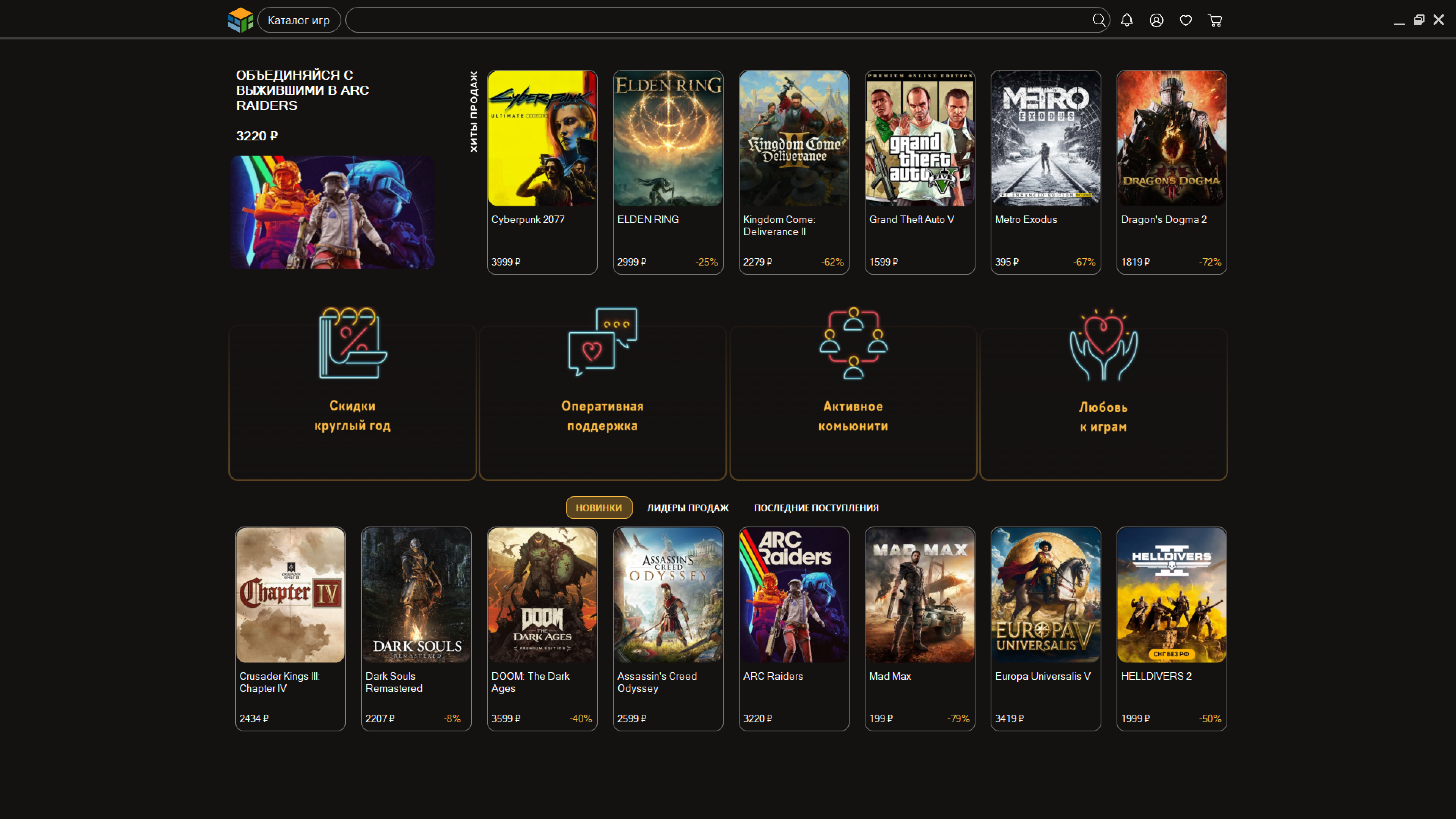
Все формы поддерживают плавную навигацию, визуальную обратную связь (например, анимация при добавлении в корзину) и обработку ошибок (некорректный ввод, отсутствие интернета при восстановлении пароля и т.д.).

Интерфейс не требует адаптивности под мобильные устройства, так как приложение — десктопное. Однако учитывается разное разрешение экрана (поддержка 1366×768 и выше).

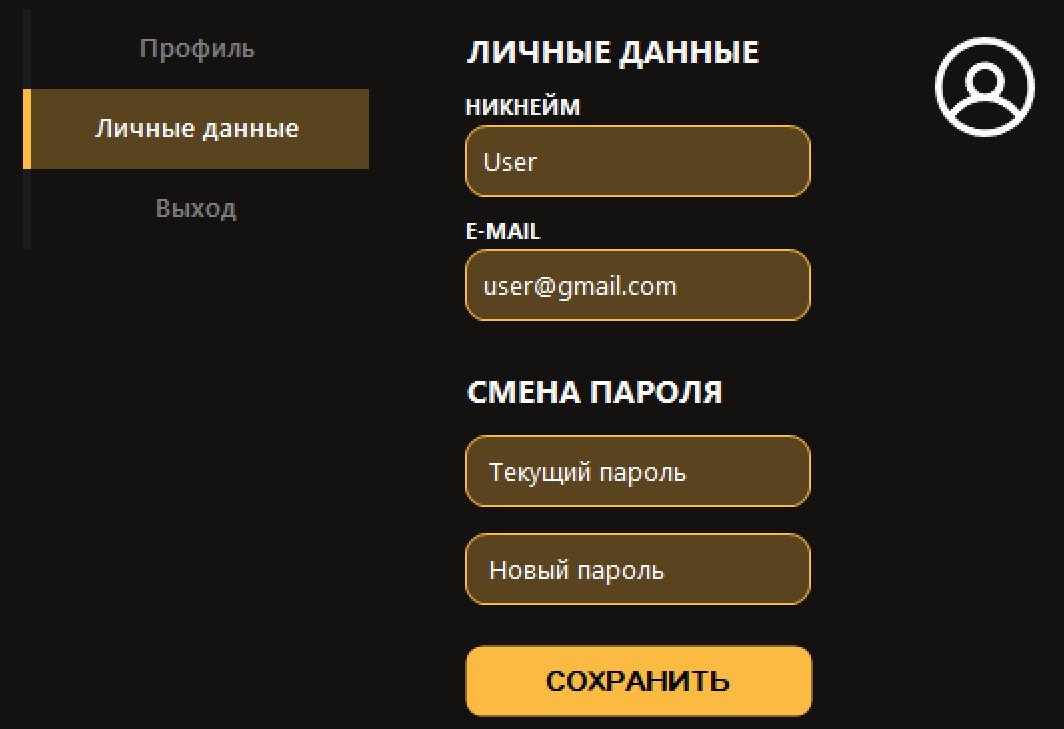
**Прототипы основных форм:**

Для визуализации концепции интерфейса были разработаны прототипы основных страниц системы (см. Рисунки 2-5):

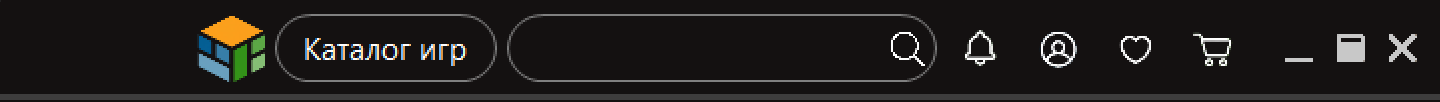
* **Главная форма (Рисунок 2)**.

*Рисунок 2 - Прототип главной формы приложения*

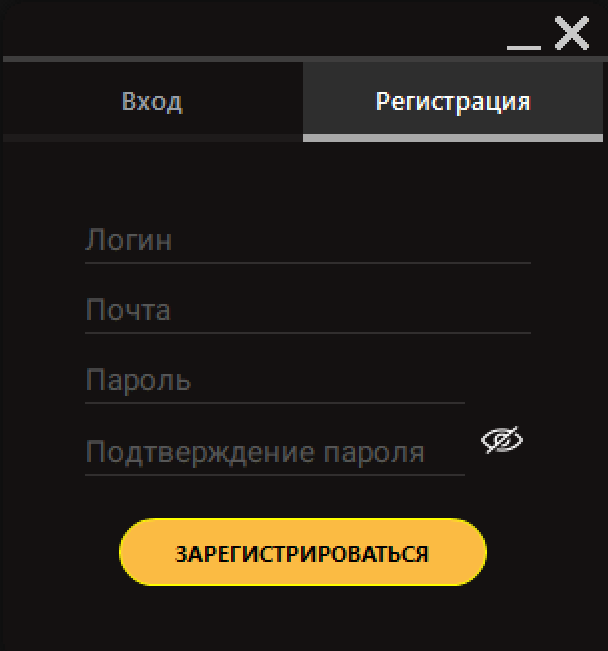
* **Профиль пользователя (Рисунок 3)**.

*Рисунок 3 - Прототип формы профиля пользователя*

* **Навигация (Рисунок 4)**.

*Рисунок 4 - Прототип контрола навигации*

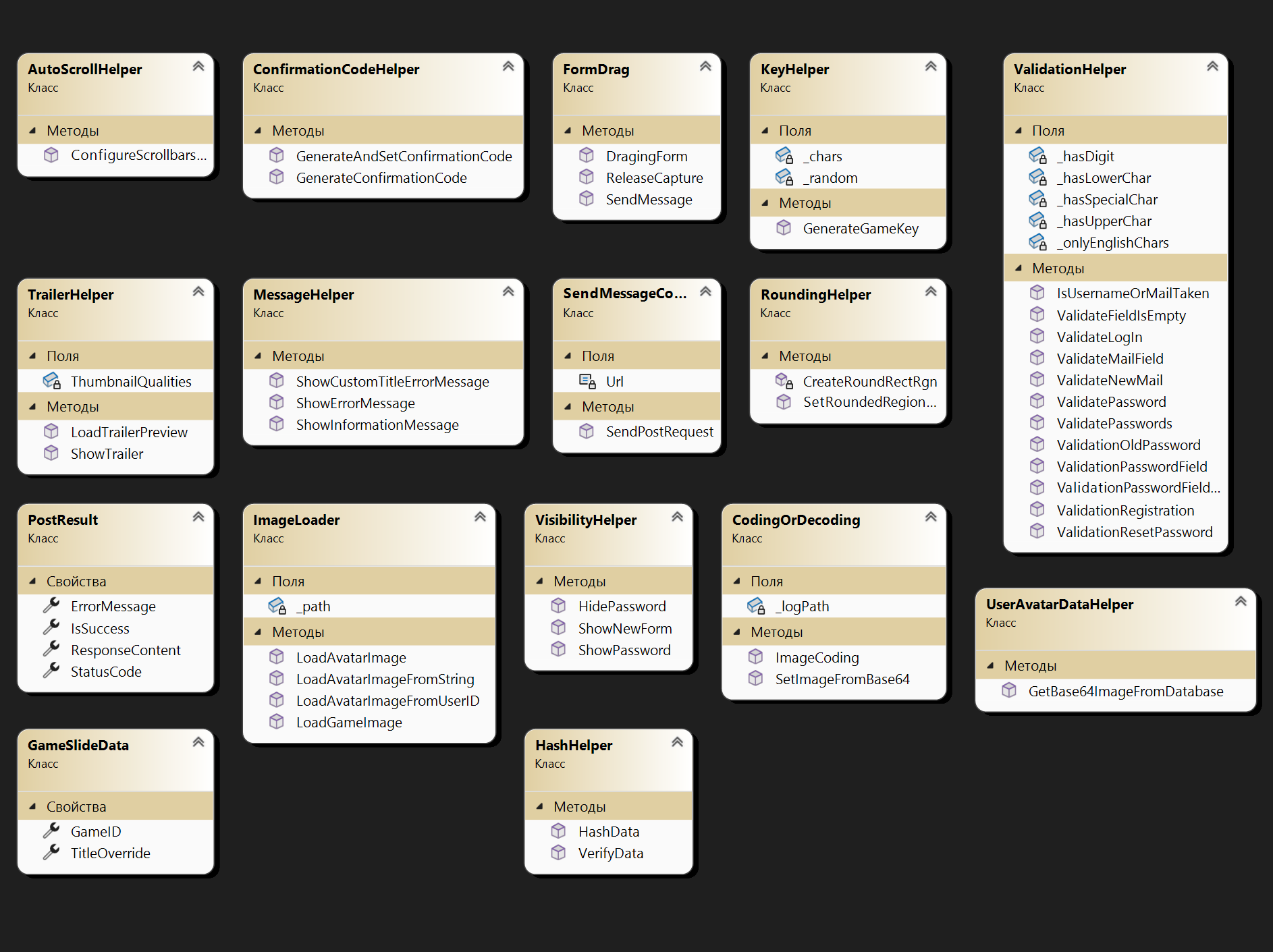
* **Системные диалоги (Регистрация) (Рисунок 5)**.



*Рисунок 5 - Прототип формы регистрации*

## Реализация

## 3.1 Реализация основных функций

*Рисунок 6 - Диаграмма классов*

Разработка приложения GameEnergy выполнена с использованием архитектурных паттернов MVC (Model–View–Controller) и Repository, что позволило чётко разделить обязанности между компонентами системы:

* **Model — сущности Entity Framework (Game, User, Order, Review и др.), отражающие структуру данных;**
* **View — WinForms-формы и кастомные контролы (MainGameControl, GameDetailsForm, ProfileForm и т.д.);**
* **Controller / Service-уровень — классы-репозитории и сервисы, реализующие бизнес-логику без привязки к UI.**

**Примеры реализации:**

**Работа с данными: Entity Framework + Repository**

Подключение к локальной базе данных MS SQL Server осуществляется через Entity Framework Core. Для изоляции логики работы с БД от интерфейса реализованы репозитории:

public interface IGameRepository  
{  
 List<Game> GetAll();  
 Game GetById(int id);  
 void Add(Game game);  
 void Update(Game game);  
 void Delete(int id);  
}

Аналогичные интерфейсы созданы для IUserRepository, IOrderRepository, IReviewRepository и др. Это упрощает модификацию и тестирование (даже без unit-тестов — логика легко читается и поддерживается).

Пример вызова из формы:

var gameRepo = new GameRepository();  
var games = gameRepo.GetAll()  
 .Where(g => g.Price <= maxPrice && g.GenreId == selectedGenre)  
 .OrderBy(g => g.Title)  
 .ToList();

**Авторизация и безопасность:**

1. Регистрация и вход реализованы с проверкой логина/пароля через хеш-сравнение:

var storedHash = user.PasswordHash;  
if (BCrypt.Net.BCrypt.Verify(inputPassword, storedHash)) { /\* авторизован \*/ }

1. Пароли и коды восстановления хешируются с помощью BCrypt.Net-Next — даже при утечке БД злоумышленник не сможет восстановить исходные значения.
2. При восстановлении пароля:
   1. Пользователь вводит email;
   2. Система генерирует одноразовый код, хеширует его и сохраняет в БД;
   3. Код отправляется на почту через SMTP-подобный запрос с использованием Newtonsoft.Json и DnsClient на локальный сервер (развёрнутый на виртуальной машине);
   4. Пользователь вводит код → система сравнивает хеш → разрешает смену пароля.

Восстановление пароля требует интернета, но основной функционал (покупка, просмотр, отзывы) работает полностью офлайн.

**Каталог и поиск:**

1. Все фильтры (жанр, разработчик, цена) и сортировки (по цене, рейтингу, дате и т.д.) реализованы на стороне приложения через LINQ-запросы к EF-контексту.
2. Поиск по названию или разработчику — без учёта регистра:

Program.context.Where(g => g.Title.Contains(searchTerm, StringComparison.OrdinalIgnoreCase))

**Корзина и заказы:**

1. Корзина хранится локально в памяти во время сессии (или в отдельной таблице CartItems, если реализовано постоянное хранение).
2. При оформлении заказа:
   1. Создаётся запись в Orders;
   2. Для каждой игры в корзине генерируется уникальный ключ активации (например, GUID);
   3. Ключ сохраняется в OrderItems.ActivationKey;
   4. После оплаты (в рамках приложения — мгновенно) заказ отображается в разделе «Мои заказы».

**Отзывы, оценки и жалобы:**

1. Каждый пользователь может оставить только один отзыв на игру проверка:

Program.context.Reviews.Where(r => r.UserId == current && r.GameId == game).Any()

1. Оценка — целое число от 1 до 5.
2. Жалобы сохраняются в таблицу Complaints и могут просматриваться администратором (в будущем — для модерации).

**Администрирование:**

1. При входе проверяется роль пользователя:

if (CurrentUser.Role == "Admin") ShowAdminPanel();

1. Админ может:
   1. Добавлять игры через форму с OpenFileDialog для загрузки изображения;
   2. Указывать код YouTube-трейлера (например, dQw4w9WgXcQ);
   3. При отображении трейлера приложение пытается загрузить превью по ссылке [https://img.youtube.com/vi/{code}/mqdefault.jpg](https://img.youtube.com/vi/%7Bcode%7D/mqdefault.jpg); при ошибке — показывает заглушку (из ресурсов проекта), что гарантирует работу без интернета.

## 3.2 Реализация интерфейсов

Интерфейс приложения разработан с использованием современных UI-библиотек:

1. Guna UI2 — для кнопок, панелей, текстовых полей с анимациями;
2. MaterialSkin — для реализации Material Design (тени, акцентные цвета, карточки).

**Кастомные контролы:**

Для повторного использования и единого стиля созданы следующие пользовательские контролы:

|  |  |
| --- | --- |
| **Контрол** | **Назначение** |
| MainGameControl | Отображает игру в каталоге: обложка, название, цена, скидка |
| GameSlideControl | Карусель новинок на главной форме |
| NavigationControl | Боковая панель навигации (Каталог, Избранное, Корзина и т.д.) |
| UserCommentsControl | Отображает отзыв одного пользователя (аватар, имя, оценка, текст) |
| CartAndOrderItemsControl | Элемент корзины или заказа с кнопками и итоговой ценой |
| OrderControl | Полная карточка заказа с ключом активации |

Каждый контрол инкапсулирует свою логику отображения и может быть легко обновлён без изменения основной формы.

**Формы:**

1. Все формы наследуются от базового шаблона с общей цветовой схемой;
2. Навигация реализована через Panel и BringToFront() или открытие дочерних форм;
3. Обработка ошибок: при исключениях показывается MessageBox с понятным пользователю сообщением («Ошибка подключения к базе данных», «Интернет недоступен» и т.д.).

## 3.3 Тестирование

Поскольку проект разрабатывался в рамках демонстрационного экзамена, специализированные unit- или интеграционные тесты не писались. Однако было проведено ручное функциональное тестирование всех ключевых сценариев.

**Основные тест-кейсы:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Сценарий** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| TC-01 | Регистрация нового пользователя | Учётная запись создана, хеш пароля сохранён | Успешно | ✅ |
| TC-02 | Авторизация под пользователем | Открывается форма каталога | Успешно | ✅ |
| TC-03 | Авторизация под админом | Доступна админ-панель | Успешно | ✅ |
| TC-04 | Поиск игры по названию | Отображаются совпадения | Успешно | ✅ |
| TC-05 | Фильтрация по жанру и цене | Корректный набор игр | Успешно | ✅ |
| TC-06 | Добавление в корзину и оформление заказа | Заказ создан, ключ активации отображён | Успешно | ✅ |
| TC-07 | Оценка и отзыв | Отзыв появляется в карточке игры | Успешно | ✅ |
| TC-08 | Добавление игры админом с изображением | Игра отображается в каталоге | Успешно | ✅ |
| TC-09 | Восстановление пароля (с интернетом) | Код приходит на почту, пароль меняется | Успешно | ✅ |
| TC-10 | Просмотр трейлера без интернета | Отображается заглушка, нет ошибки | Успешно | ✅ |

Все выявленные баги (например, дублирование игр в корзине, ошибка при удалении отзыва) были устранены в процессе отладки.

## Руководство администратора/пользователя

## 4.1 Руководство для администратора

Администратор обладает расширенными правами и отвечает за наполнение каталога, управление контентом и взаимодействие с пользователями через системные уведомления.

* **Авторизация:**
  + Запустите приложение GameEnergy.
  + На форме авторизации введите логин и пароль учётной записи с ролью «Admin».
  + Нажмите кнопку «Войти».
  + После успешной авторизации откроется главная форма с доступом к админ-панели (обычно отображается кнопка «Админка» или отдельный пункт в меню).
* **Добавление новой игры:**
  + Перейдите в Каталог → «Добавить игру».
  + Заполните поля:
    - Название — полное название игры;
    - Разработчик — название студии;
    - Жанр — выберите из списка (справочник);
    - Дата выхода — укажите дату;
    - Описание — краткое описание (до 3000 символов);
    - Цена — стоимость в рублях;
    - Старая цена — если есть скидка (необязательно);
  + Нажмите «Загрузить обложку» — выберите изображение (поддерживаются JPG, PNG до 5 МБ).
  + Укажите код YouTube-трейлера (например, dQw4w9WgXcQ из ссылки <https://youtu.be/dQw4w9WgXcQ>).
  + Нажмите «Сохранить».

Изображение и ключ сохраняются в локальной базе данных. Трейлер будет отображаться как превью; при отсутствии интернета покажется заглушка.

* **Редактирование и удаление игр:**
  + В списке игр нажмите на нужную запись → кнопка «Удалить».
  + Подтвердите удаление (операция необратима).
* **Создание системного уведомления:**
  + В админ-панели откройте раздел «Уведомления».
  + Нажмите «Новое уведомление».
  + Укажите:
    - Заголовок (например, «Технические работы»);
    - Текст (до 500 символов).
  + Нажмите «Опубликовать».
  + Создание резервных копий базы данных.
  + Восстановление базы данных из резервной копии.
  + Обновление системы до новых версий.

Все зарегистрированные пользователи увидят это уведомление при следующем запуске приложения в разделе «Уведомления».

* **Просмотр жалоб:**
  + В админ-панели откройте раздел «Уведомления».
  + Выведется список всех жалоб
  + Нажмите на интересующую вас жалобу
  + Откроется отдельное модальное окно, в котором отобразится информация по этой жалобе, а именно отправитель, причина репорта и сообщение
  + Если закрыть окно уведомление автоматически удалится из списка уведомлений, но останется в БД

## 4.2 Руководство для пользователя (покупателя)

* **Регистрация и авторизация:**
* Запустите приложение.
* На стартовой форме нажмите «Регистрация».
* Заполните:
* Логин (уникальный);
* Email (для восстановления пароля);
* Пароль (минимум 8 символов);
* Повтор пароля.
* Нажмите «Зарегистрироваться».
* После регистрации войдите через форму «Авторизация», указав логин и пароль.

Пароль хранится в зашифрованном виде. При утере — восстановите через email (требуется интернет).

* **Поиск и фильтрация книг:**
* На главной форме каталога:
* Введите название или имя разработчика в поле «Поиск»;
* Используйте фильтры: жанр, диапазон цены, разработчик;
* Выберите сортировку: по цене, рейтингу, дате и т.д.
* Результаты обновляются автоматически.
* **Просмотр карточки игры:**
* Кликните на игру в каталоге.
* Откроется подробная карточка с:
* Обложкой;
* Описанием;
* Трейлером (если указан код YouTube);
* Отзывами других пользователей;
* Возможностью добавить в корзину, в избранное, оценить или написать отзыв.
* **Покупка игры:**
* Нажмите «Добавить в корзину».
* Перейдите в раздел «Корзина» (через навигацию).
* Нажмите «Оформить заказ».
* После подтверждения игра появится в разделе «Мои заказы» с указанием ключа активации.

Покупка в данной версии симулируется — деньги не списываются, ключ выдаётся мгновенно.

* **Избранное, отзывы и профиль:**
* **Избранное**: игры, добавленные через кнопку «⭐», доступны в отдельном разделе.
* **Отзыв**: в карточке игры укажите оценку (1–5) и текст → нажмите «Отправить».
* **Профиль**:
* Нажмите на аватар или имя в шапке;
* В форме профиля можно изменить:
  + - * Никнейм;
      * Email;
      * Пароль;
      * Аватар (через загрузку файла).

Вы можете перейти в профиль другого пользователя, кликнув на его имя в отзыве.

* **Восстановление пароля:**
  + На форме авторизации нажмите «Забыли пароль?».
  + Введите email, указанный при регистрации.
  + На почту придет код подтверждения (требуется интернет).
  + Введите код и новый пароль.
  + Авторизуйтесь с новым паролем.

Без доступа к email восстановление невозможно. Без интернета — функция недоступна.

## Мероприятия по информационной безопасности

## 5.1 Определение угроз при эксплуатации приложения «GameEnergy»

Несмотря на то, что приложение локальное и не использует веб-интерфейс, оно всё же подвержено ряду потенциальных угроз:

* **SQL-инъекции — при некорректной обработке пользовательского ввода возможна попытка внедрения вредоносных SQL-запросов (например, через поиск или форму авторизации).**
* **Кража учётных данных — если злоумышленник получит доступ к локальной базе данных, он может попытаться восстановить пароли.**
* **Неавторизованный доступ к ключам активации — если приложение не проверяет права пользователя при отображении заказов, возможна утечка цифровых ключей.**
* **Загрузка вредоносных изображений — через форму добавления игры администратор (или потенциально скомпрометированный аккаунт) может загрузить исполняемый файл под видом изображения.**
* **Брутфорс-атака на форму входа — перебор паролей при многократных попытках входа.**
* **Утечка персональных данных — email, аватар, имя пользователя хранятся в БД и могут быть извлечены при физическом доступе к файлу базы.**
* **Небезопасное хранение кодов восстановления — если код отправляется на почту и хранится в открытом виде, возможен взлом аккаунта.**

## 5.2 Что сделано как разработчиком для предотвращения этих угроз

* **Защита от SQL-инъекций:**
  + - Весь доступ к базе данных осуществляется через Entity Framework Core, который автоматически использует параметризованные запросы.
    - Пользовательский ввод (поиск, фильтрация) обрабатывается через LINQ-выражения, исключающие прямую конкатенацию строк.
* **Безопасное хранение паролей и кодов восстановления:**
  + - Пароли хешируются с помощью библиотеки BCrypt.Net-Next — современный адаптивный алгоритм с «солью».
    - Код восстановления пароля также хешируется и сохраняется в поле RecoveryCodeHash таблицы Users.
    - Даже при утечке БД восстановить исходный пароль или код вычислительно невозможно.
* **Контроль доступа на уровне приложения:**
  + - Все действия, связанные с изменением данных (добавление игры, удаление отзыва), проверяют роль текущего пользователя: if (CurrentUser.Role != "Admin") { MessageBox.Show("Доступ запрещён"); return; }
    - Ключи активации отображаются только в заказах текущего пользователя — нет общего списка всех ключей.
* **Безопасная загрузка изображений:**
* При добавлении обложки игры используется OpenFileDialog с фильтром: openFileDialog.Filter = "Image Files|\*.jpg;\*.jpeg;\*.png;";
* Файл читается как массив байтов и сохраняется в varbinary(max) — не сохраняется на диск, не исполняется.
* При отображении изображение загружается из БД, а не из файловой системы — исключена RCE (Remote Code Execution).
* **Защита от брутфорса:**
  + - Реализована блокировка после 5 неудачных попыток входа (в памяти или в БД с временной меткой).
    - После блокировки пользователь не может авторизоваться в течение 15 минут.
* **Минимизация утечки данных:**
  + - Все изображения, ключи, профили хранятся внутри локальной БД — нет внешних зависимостей.
    - Приложение не отправляет данные в сеть (кроме восстановления пароля), что снижает риск перехвата.
* **Защита восстановления пароля:**
  + - Код отправляется только на email из БД.
    - Даже если злоумышленник получит хеш кода — он не узнает его значение.
    - Код одноразовый и сроком действия 15 минут (реализовано через DateTime в БД).

## 5.3 Рекомендации пользователям по безопасной работе

* **Общие рекомендации:**
  + - Используйте сложный пароль (не менее 8 символов, с цифрами и символами).
    - Не используйте один пароль для нескольких приложений.
    - Регулярно меняйте пароль, особенно если компьютер используется в общем доступе.
    - Не передавайте файл базы данных (\*.mdf) третьим лицам — он содержит ключи и персональные данные.
* **Рекомендации администратору:**
  + - Не оставляйте учётную запись администратора без пароля.
    - При развёртывании на другом ПК — измените пароль админа.
    - Ограничьте физический доступ к компьютеру с базой данных.
    - Регулярно делайте резервные копии БД (через SQL Server Management Studio).
* **В случае компрометации:**
  + - Немедленно смените пароль через восстановление (если есть доступ к email).
    - Если email утерян — администратор может сбросить пароль вручную через БД (только при физическом доступе).

## Заключение

В ходе выполнения дипломного проекта было успешно разработано локальное десктопное приложение «GameEnergy», предназначенное для удобного и безопасного приобретения цифровых ключей активации компьютерных игр без зависимости от онлайн-сервисов. Приложение полностью соответствует заявленной цели — демонстрации навыков разработки на платформе C# WinForms в рамках подготовки к демонстрационному экзамену по компетенции «Разработка приложений».

Основные задачи проекта выполнены в полном объёме:

* Реализован каталог игр с расширенными возможностями поиска, фильтрации и сортировки;
* Обеспечена поддержка двух ролей пользователей: обычного покупателя и администратора;
* Разработаны модули корзины, избранного, оформления заказа, отзывов и оценок;
* Реализована система уведомлений, управляемая администратором;
* Обеспечена возможность редактирования профиля, включая смену аватара;
* Добавлена функция жалобы на игру с выбором категории нарушения;
* Реализованы безопасная регистрация, авторизация и восстановление пароля с использованием хеширования BCrypt;
* Все данные (включая изображения и ключи активации) хранятся локально в MS SQL Server, что гарантирует автономность и безопасность;
* Архитектура приложения построена на принципах MVC и паттерна Repository, что обеспечивает читаемость, расширяемость и сопровождаемость кода;
* Интерфейс реализован с использованием современных UI-библиотек Guna UI2 и MaterialSkin, а также кастомных контролов, обеспечивающих единый и приятный пользовательский опыт.

Приложение не требует подключения к интернету для основного функционала — за исключением восстановления пароля и загрузки превью трейлера (с корректной обработкой отсутствия сети). Это делает его пригодным для развёртывания в учебных заведениях, корпоративных сетях или на домашних компьютерах без стабильного интернета.

Проведённое ручное функциональное тестирование подтвердило корректную работу всех ключевых модулей: от регистрации до получения ключа активации. Выявленные ошибки были устранены в процессе отладки.

Практическая значимость проекта заключается в демонстрации возможности создания полноценной, защищённой и удобной системы цифровой коммерции без использования веб-технологий, что особенно актуально в условиях требований к локализации данных и отказа от облачных зависимостей.

В качестве направлений дальнейшего развития проекта можно рассмотреть:

* Добавление оффлайн-режима восстановления пароля (через секретные вопросы);
* Реализация системы рейтинга пользователей (по активности, количеству отзывов);
* Внедрение статистики продаж для администратора;
* Поддержка локализации (русский/английский интерфейс);
* Экспорт/импорт каталога игр в формате JSON для переноса между ПК.

Таким образом, приложение GameEnergy представляет собой завершённое, функционально насыщенное и технически грамотно реализованное решение, соответствующее требованиям профессионального стандарта и демонстрирующее высокий уровень владения стеком C# + WinForms + Entity Framework + MS SQL.

## Список источников

### Нормативная документация

1. ГОСТ 34.602–2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. [Электронный ресурс] - <https://protect.gost.ru/document1.aspx?control=31&id=241754>
2. ГОСТ 19.106–78. Единая система программной документации. Требования к программным документам, выполненным печатным способом. [Электронный ресурс] - <https://protect.gost.ru/document1.aspx?control=31&id=241756>
3. ГОСТ Р 2.105–2019. Единая система конструкторской документации. Общие требования к текстовым документам. [Электронный ресурс] - <https://protect.gost.ru/document1.aspx?control=31&id=217667>

### Официальная документация и технические ресурсы

1. Microsoft. .NET Documentation [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/>, свободный. — (дата обращения: 01.12.2025).
2. Microsoft. Entity Framework Core Documentation [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/ef/core/>, свободный. — (дата обращения: 01.12.2025).
3. Microsoft. Windows Forms Documentation [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/>, свободный. — (дата обращения: 01.12.2025).
4. Microsoft. SQL Server Documentation [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/sql/>, свободный. — (дата обращения: 01.12.2025).

### Библиотеки и инструменты

1. Guna UI2 Framework. Official GitHub Repository [Электронный ресурс]. — Режим доступа: <https://github.com/sunnyvalley/Guna.UI2>, свободный. — (дата обращения: 01.12.2025).
2. MaterialSkin for .NET. GitHub Repository [Электронный ресурс]. — Режим доступа: <https://github.com/IgnaceMaes/MaterialSkin>, свободный. — (дата обращения: 01.12.2025).
3. BCrypt.Net-Next. NuGet Package Documentation [Электронный ресурс]. — Режим доступа: <https://www.nuget.org/packages/BCrypt.Net-Next>, свободный. — (дата обращения: 01.12.2025).

### Учебные и научные издания

1. Троелсен Э., Джепикс Ф. Язык программирования C# 10 и платформа .NET 6. — 9-е изд. — СПб.: Питер, 2022. — 1296 с.
2. Либерти Дж., Харди Д. Освой самостоятельно C# за 21 день. — М.: Вильямс, 2021. — 672 с.
3. Макконнелл С. Совершенный код. — 2-е изд. — М.: Русская редакция, 2020. — 950 с.
4. Фаулер М. Рефакторинг. Улучшение существующего кода. — М.: Вильямс, 2021. — 432 с.
5. Васюков В.Т., Симбирцев А.В. Информационная безопасность: учебник для вузов. — 2-е изд., испр. и доп. — М.: Юрайт, 2023. — 379 с.

### Интернет-ресурсы

1. OWASP Foundation. Top 10 Web Application Security Risks [Электронный ресурс]. — Режим доступа: <https://owasp.org/www-project-top-ten/>, свободный. — (дата обращения: 01.12.2025).
2. YouTube Data API — Embedding Videos [Электронный ресурс]. — Режим доступа: <https://developers.google.com/youtube/player_parameters>, свободный. — (дата обращения: 01.12.2025).

## Приложение А

### Код для проверки почты на волидность:

public static bool ValidateNewMail(Guna2TextBox newMail)

{

string mail = newMail.Text?.Trim();

int currentUserId = Program.CurrentUser.UserID;

Users currentUser = Program.context.Users.FirstOrDefault(u => u.UserID == currentUserId);

if (mail != currentUser.Email)

{

currentUser.Email = mail;

Program.context.SaveChanges();

return true;

}

else

{

MessageHelper.ShowErrorMessage("Новая почта должна отличаться от старой");

return false;

}

}

### Класс для хэширования и верификации данных (пароля и кода подтверждения для восстановления пароля):

/// <summary>

/// Хэширование и верификация данных.

/// </summary>

internal class HashHelper

{

/// <summary>

/// Хэширование данных.

/// </summary>

/// <param name="field"></param>

/// <returns>хэшированное значение</returns>

public static string HashData(string field)

{

return BCrypt.Net.BCrypt.HashPassword(field);

}

/// <summary>

/// Верификация данных.

/// </summary>

/// <param name="field"></param>

/// <param name="storedHash"></param>

/// <returns>сравнение данных</returns>

public static bool VerifyData(string field, string storedHash)

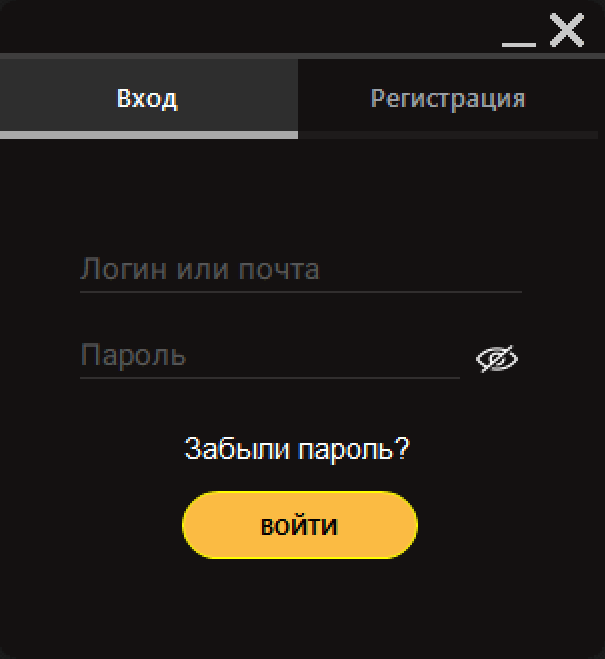
{

return BCrypt.Net.BCrypt.Verify(field, storedHash);

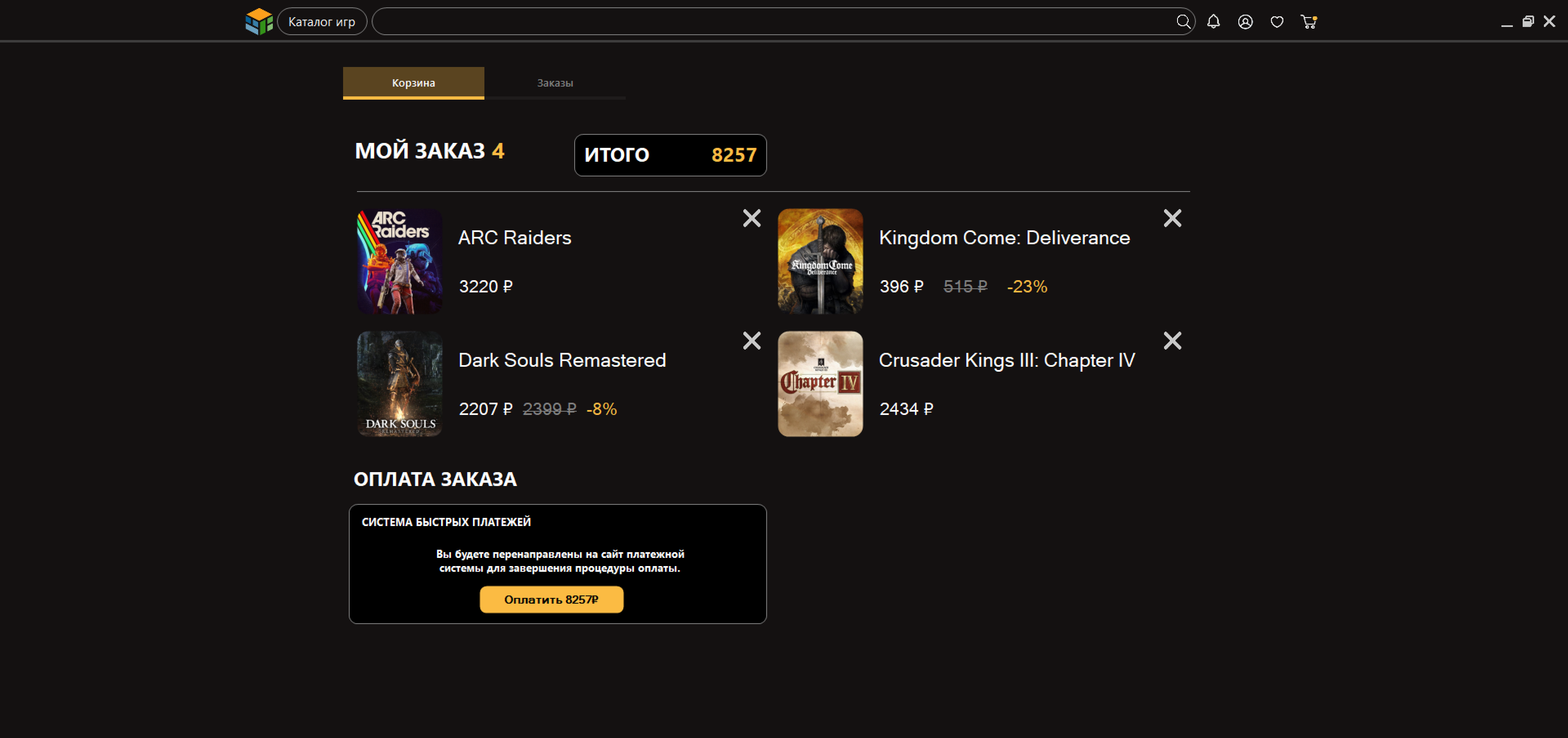
}

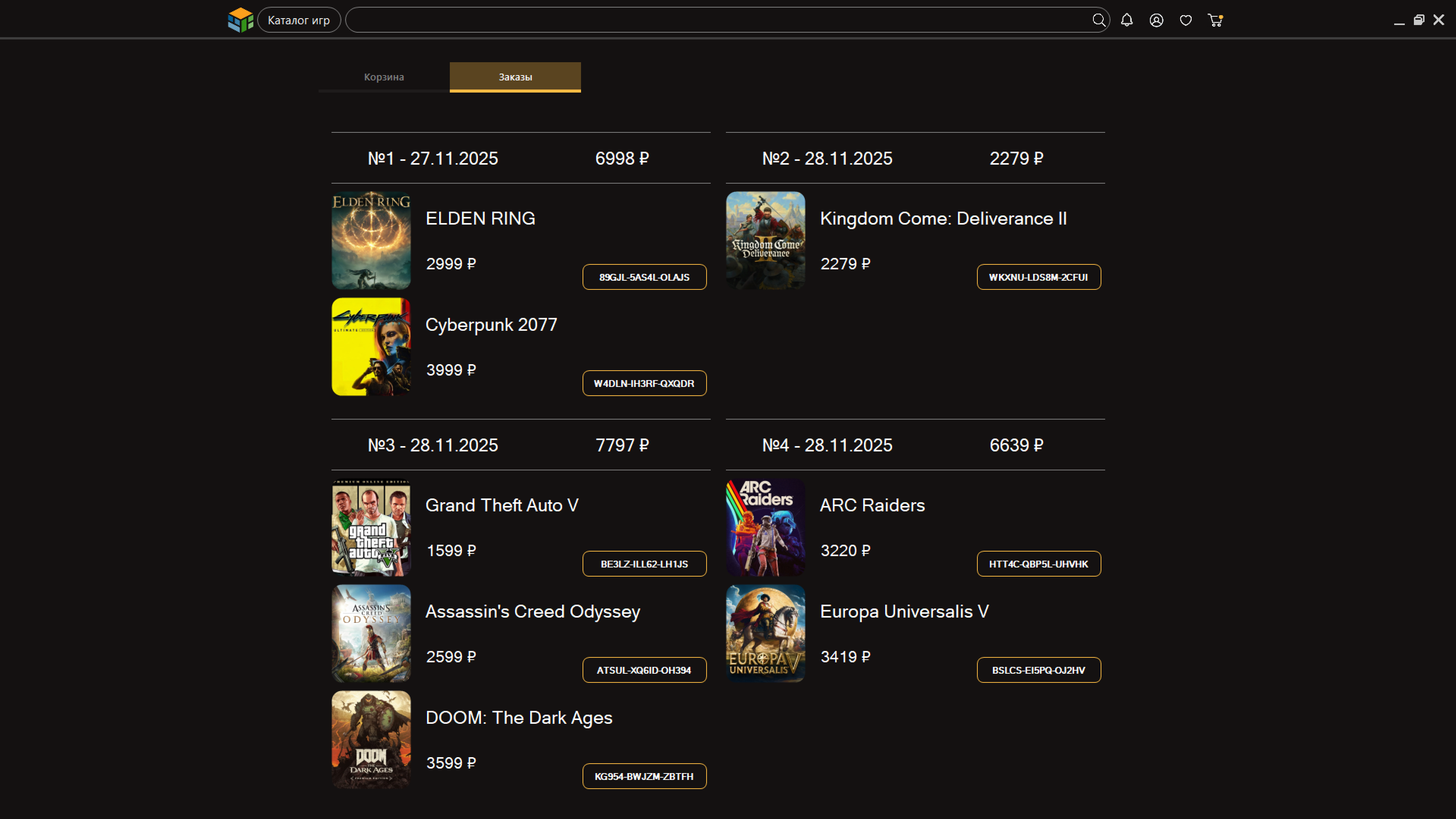
}

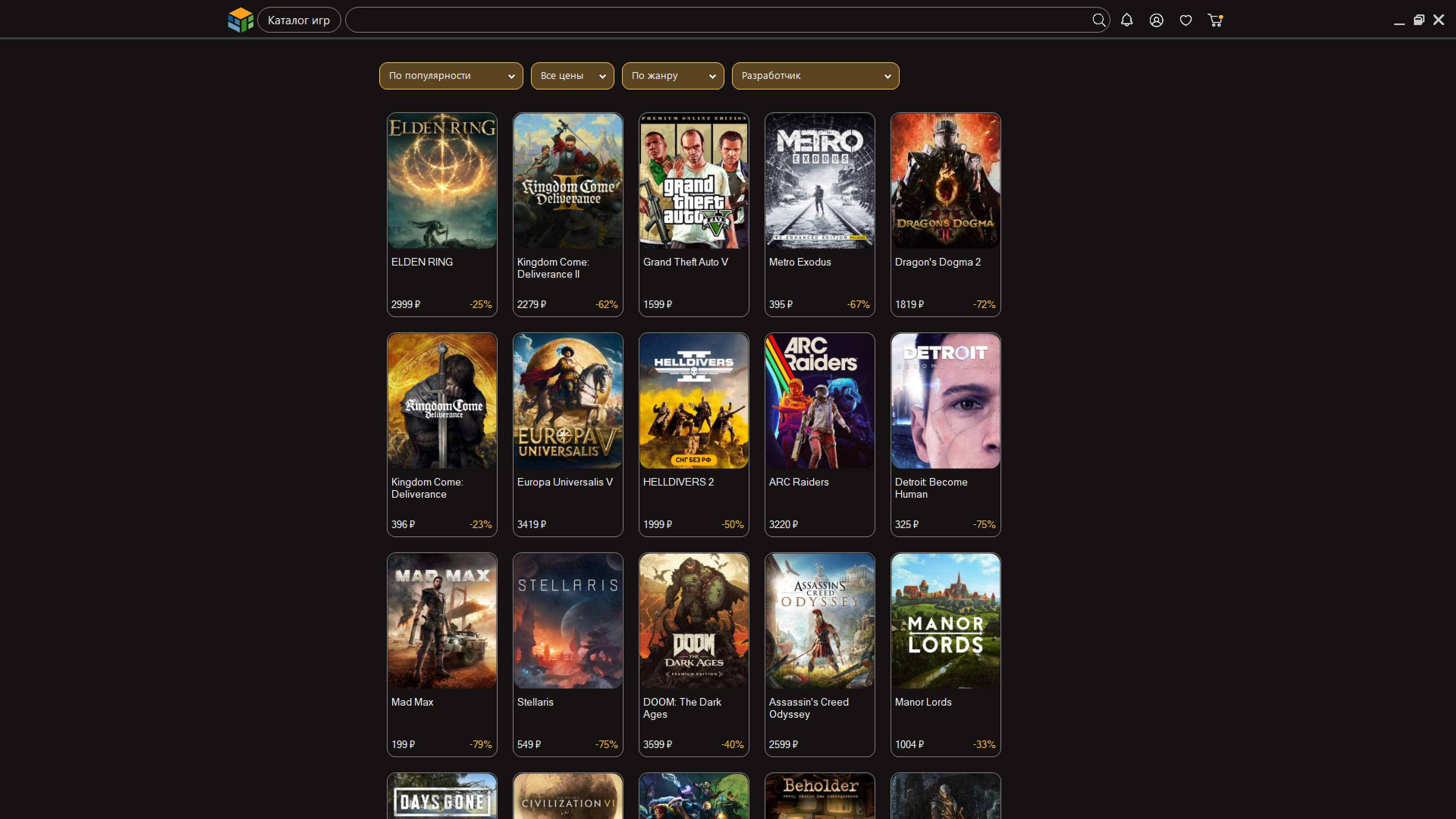
## Приложение Б

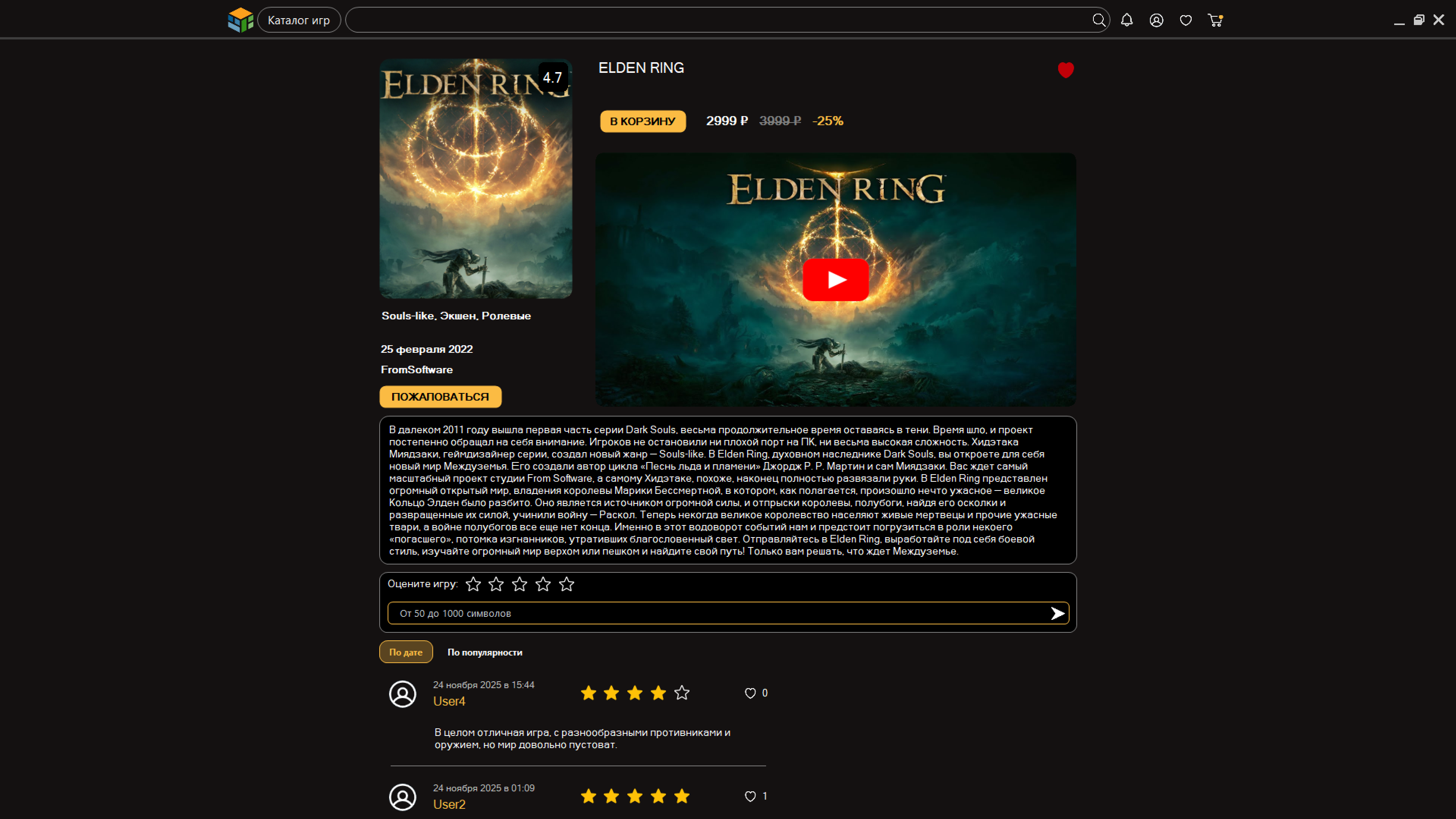


*Рисунок 7 – Форма авторизации*

*Рисунок 8 – корзина пользователя*

*Рисунок 9 – Заказы пользователя*

*Рисунок 10 – Форма каталога*

*Рисунок 11– Форма информации о игре*