

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С. А. Виденин
подпись инициалы, фамилия
«___» _____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

ИС «Сервис распространения цифрового лицензионного контента»

Руководитель	_____	<u>Зав. каф., доцент</u>	<u>С. А. Виденин</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>В. К. Касперович</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	<u>Старший преподаватель</u>	<u>Ю. В. Шмагрис</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1 РЫНОК ЦИФРОВОГО КОНТЕНТА.....	6
1.1 Музыкальные сервисы.....	6
1.2 Онлайн-кинотеатры	8
1.3 Игровые сервисы.....	10
1.4 Продажа ПО.....	12
1.5 Перспективы разрабатываемой системы.....	14
ГЛАВА 2 ПРОЕКТНАЯ ЧАСТЬ	16
2.1 Диаграммы сценариев использования	16
2.2 Диаграмма развёртывания.....	27
2.3 Диаграммы баз данных.....	28
ГЛАВА 3 РЕАЛИЗАЦИЯ.....	39
3.1 Шаблон проектирования	39
3.2 Архитектурное решение.....	41
3.3 Интерфейс приложения	42
ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	53
ПРИЛОЖЕНИЕ А	55

ВВЕДЕНИЕ

В текущей экономической и политической ситуации в Российской Федерации, присутствует и активно поддерживается правительством необходимость в импортозамещении продукции, в том числе и программного обеспечения.

В рамках данной работы выбор был сделан на системы цифрового распространения электронных товаров. В настоящий момент уже существует огромное множество платформ, на которых происходит продажа/предоставление лицензированного цифрового контента. Практически все из них предоставляют лишь узко направленную специализацию. Примерами могут послужить следующие сервисы: Battle.NET, Origin, Amediateka, ivi, Netflix, Steam, Spotify, Apple Music и другие.

При наличии такого уровня конкуренции сложно будет разработать продукт, который сможет конкурировать с существующими системами. Именно поэтому возникает вопрос в создании не просто отечественного продукта, который реализует весь функционал аналогов, а продукта, который должен обладать своей особенностью, способной заинтересовать клиентов.

Главной и отличительной особенностью разрабатываемой системы является её широкая направленность. Это значит, что сервис будет предоставлять услуги купли/продажи не только одной категории электронной продукции (музыка, фильм, игра, программа и др.), а их всех вместе взятых.

Система также будет поддерживать полный социальный пакет. Что это значит? Многим пользователям Steam это уже и так понятно. Подразумевается, что все пользователи смогут различными способами взаимодействовать между собой путём личных сообщений, комментариев в ветках каждого товара, или же просто форумов.

Данное решение достаточно актуально на данный момент, так как это облегчает жизнь многих пользователей интернета. Ведь хорошо, когда тебе не

нужно пользоваться кучей различных сервисов для удовлетворения своих нужд. Например, вы можете запустить какой-нибудь редактор (Photoshop и др.), слушать музыку и общаться с коллегами, не покидая единой среды.

Таким образом, **объект** рассмотрения в рамках выполняемой работы – весь цифровой лицензионный контент интернета, а **предмет**, в рамках которого предполагается реализация контента, – Информационная система «Сервис распространения цифрового лицензионного контента». В дальнейшем – ИС «Сервис распространения цифрового лицензионного контента» или ИС «СРЦЛК».

ГЛАВА 1 РЫНОК ЦИФРОВОГО КОНТЕНТА

На сегодняшний день в сети Интернет работает огромное множество сервисов, предоставляющих доступ к тому или иному цифровому контенту (фильм, программа, игра, музыка и др.). Далее рассмотрим несколько из них.

1.1 Музыкальные сервисы

С развитием технологий классические методы получения и прослушивания музыки уходят в прошлое. К таким методам относятся:

- использование CD-дисков;
- прослушивание загруженных песен .mp3 формата на электронный носитель (телефон, флэш-карта и др.).

На их замену приходят своего рода стриминговые сервисы, способные предоставить пользователю полный доступ ко всей библиотеке сервиса с любого устройства, имеющего выход в интернет. Поскольку проблем с доступом в интернет в настоящее время просто на просто нет, его наличие не является проблемой.

Практически все музыкальные сервисы предоставляют следующие возможности:

- создание собственных плейлистов;
- сохранение музыки на локальный носитель;
- музыка в хорошем качестве;
- объёмные библиотеки музыки;
- музыкальная подборка в соответствии со вкусами пользователя.

Различные сервисы предлагают свои «фишки» для того, чтобы люди выбрали именно их сервис. К ним можно отнести:

- эксклюзивные альбомы;

- прайс листы;
- методы подборки песен;
- тематика сервиса;
- и другие.

Различия между всеми сервисами имеют довольно незначительный характер, если вы не какой-нибудь меломан, которому важно иметь в плейлисте тот или иной эксклюзивный альбом или же вам важны все тонкости при подборе музыки.

Если же вы обычный человек и хотите просто послушать музыку, то лучшим выбором будет Spotify, Apple Music, Google Play Music.

На рисунках 1 и 2 можно увидеть примеры интерфейсов музыкальных стриминговых сервисов. При это, на Рисунке 1 можно увидеть интерфейс сервиса Spotify, а на рисунке 2 – Apple Music.

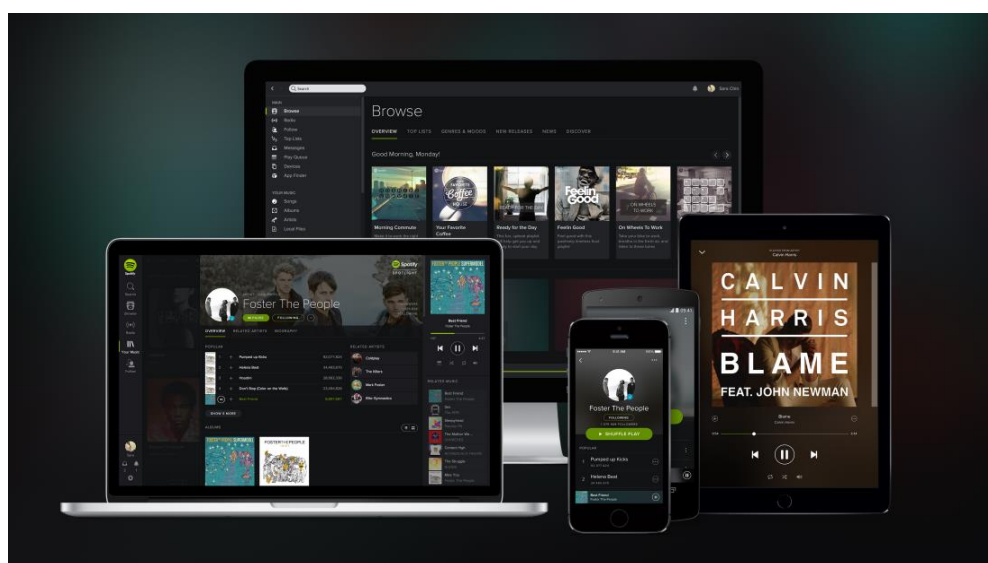


Рисунок 1 – Spotify

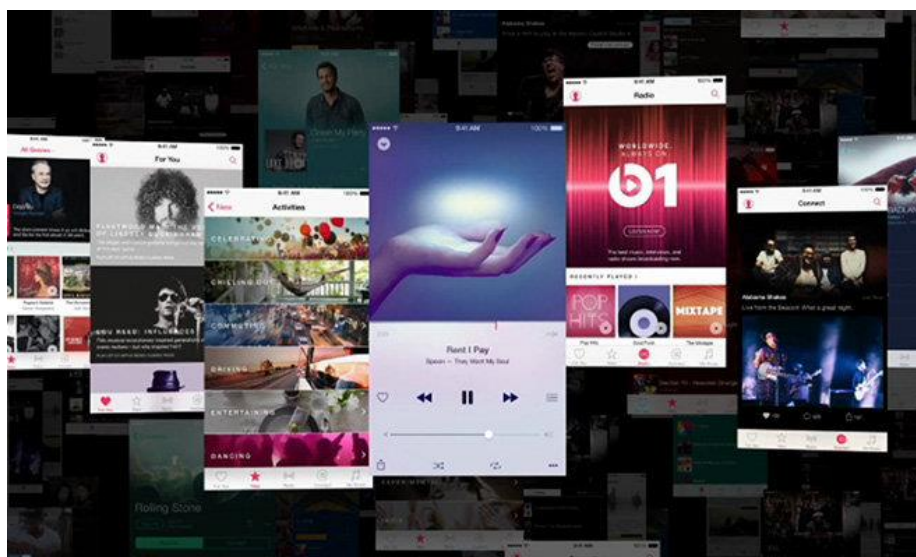


Рисунок 2 – Apple Music

Существенных недостатков у всех музыкальных стриминговых сервисов просто быть не может. К мелким недостаткам можно отнести лишь пару вещей – это плата за интернет и, возможно, за сам сервис, а также некоторые неудобства в пользовательском интерфейсе.

1.2 Онлайн-кинотеатры

Раньше, чтобы посмотреть какое-нибудь кино/сериал, приходилось покупать DVD-проигрыватели и диски у ритейлеров города. Но с развитием технологий интернета и общества в целом появилась возможность просмотра медиа-контента через интернет. Если опустить возможности «пиратства», одним из основных способов просмотра фильмов/сериалов является использование интернет сервисов (или онлайн кинотеатров).

Услуги онлайн кинотеатров очень популярны в наши дни. К таким сервисам можно отнести Amediateka, ivi.ru, Megogo, Netflix и многие другие. Ни для кого не секрет, что просмотр фильмов и сериалов является одним из основных видов проведения досуга в настоящее время. И данные сервисы позволяют легко получить доступ к легальному просмотру фильмов и сериалов на любом устройстве в любое время.

Что касается основных преимуществ использования онлайн-кинотеатров, то к ним можно отнести следующее:

- доступ к библиотеке сервиса;
- медиа контент имеет выборочное качество (от плохо до отличного) в зависимости от возможностей интернет соединения, а также предпочтения клиента;
- доступность с любого устройства;
- система подписок.

Все представленные преимущества никак нельзя отнести абсолютно ко всем онлайн кинотеатрам. Давайте посмотрим на каждое из них в отдельности.

Библиотека – это основополагающее преимущество онлайн-кинотеатра. У каждого сервиса имеется своя библиотека мультимедиа контента, и, соответственно, в зависимости от выбранного кинотеатра, вам откроется лишь определённая перечень наименований, которая доступна в рамках выбранного кинотеатра. Именно поэтому библиотеку нельзя назвать преимуществом абсолютно всех сервисов, так как далеко не все люди имеют одинаковые предпочтения и каждый выбирает себе сервис по вкусу.

Качество – также далеко не мало важный фактор, влияющий на выбор онлайн-кинотеатра. У каждого клиента имеется своё устройство для просмотра мультимедиа контента, а множество вариантов девайсов далеко от единицы. Ведь это может быть дисплей FullHD-монитора, экран смартфона / планшета или же современный 4К-телевизор, не говоря уже о VR-очках. Следовательно, если от сервиса требуется поддержка какого-нибудь 4К видео, то данное преимущество плавно перетекает в недостаток для некоторых сервисов, не имеющих данной поддержки.

Доступность. Не поспоришь, что если онлайн-кинотеатр открывается в браузере на домашнем компьютере, то он также откроется на любом другом устройстве, способном выходить в интернет через браузер. Но тут возникает вопрос удобства использования сервиса через браузер. Ведь наличие специального приложения под отдельные устройства в разы облегчает

использование онлайн-кинотеатра. Именно поэтому, отсутствие широкого спектра приложений под разные устройства может отрицательно сказаться на клиентуре каждого отдельного сервиса.

Лишь система подписок является более-менее сбалансированной, поскольку все онлайн-кинотеатры держат средний уровень цен на одном уровне.

На рисунке 3 можно увидеть один из онлайн кинотеатров MEGOGO, а также перечень поддерживаемых им устройств.



Рисунок 3 – MEGOGO

1.3 Игровые сервисы

Игры – это неотъемлемая часть жизни интернета. В наше время игры имеют крайне высокую популярность среди людей молодого и среднего возраста. Соответственно, и потребность в играх у всего «игрового» населения также велика.

Именно поэтому в сети интернет зародилось множество платформ для продажи и предоставления игр. К таким платформам можно отнести Battle.Net, GameNet, Игры Mail, Steam, Uplay и другие.

К преимуществам их использования можно отнести следующее:

- простота использования;
- все игры выбранной платформы в одном месте;
- возможность загрузки любой игры;
- динамическое обновление контента;
- новости из мира игр;
- социальная составляющая (контакты, сообщения);
- и другие.

Естественно, использование игровых сервисов упрощает всем жизнь, так как вам больше не придётся ходить в магазины и покупать обычные диски, как это делалось ранее. Тем более, это сделать стало немного сложнее поскольку количество данных магазинов заметно уменьшилось.

В подавляющем большинстве игровых сервисов есть одно огромное «НО» - издатель предоставляет только те игры, которые выпускаются лишь им самим. Это и сковывает некоторых людей от использования данных платформ, поскольку они не способны предоставить полный спектр игр, который будет интересен каждому пользователю.

Например, есть платформа GameNet, но вам не интересна ни одна из предоставляемых игр (Black Desert, Aika и другие), то незачем и пользоваться данной платформой. Или же наоборот, существует компания Blizzard Entertainment со своей платформой Battle.Net (рисунок 4). Многим людям интересны их игры, и играть в них можно только используя их систему.

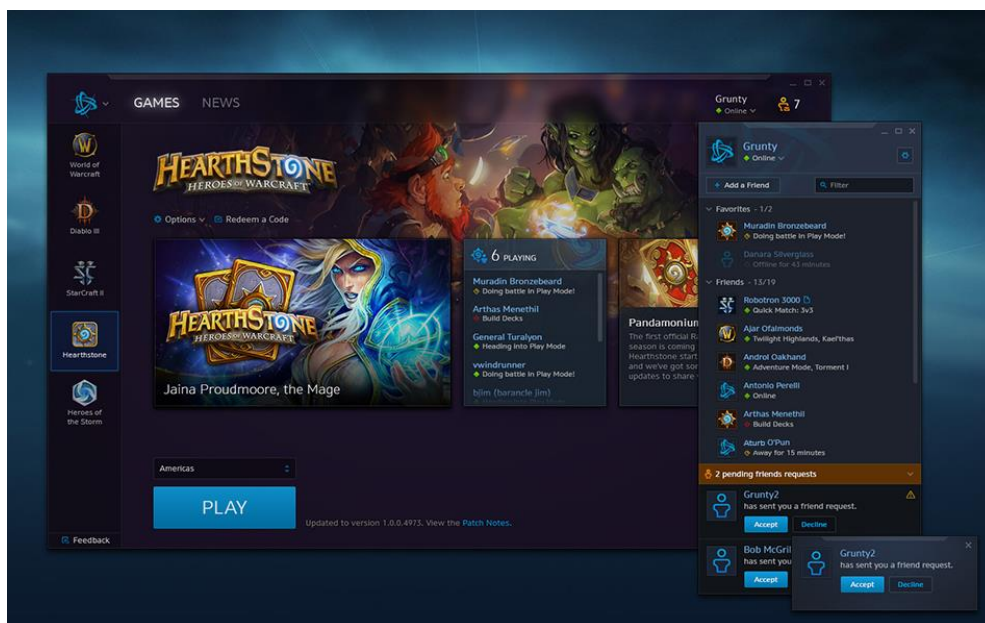


Рисунок 4 – Battle.Net

1.4 Продажа ПО

С продажей программного обеспечения ситуация значительно отличается от предыдущих видов цифровой продукции.

В наше время есть несколько способов приобретения ПО в интернет:

1. покупка непосредственно у издателя (на основном сайте);
2. покупка ключей на сторонних сайтах;
3. покупка через систему Steam.

Опуская довольно очевидные вещи, пропускаем первый пункт.

Что касается второго способа, то покупка разнородных программ в одном месте предоставляет покупателям возможность найти практически любой интересующий их продукт. Но данный способ не обладает какими-либо яркими преимуществами перед покупкой у издателя. Ведь каждый из нас может написать название нужной программы и по первой же ссылке скачать/купить интересующий товар. Поэтому данный способ покупки ПО является лишь одним из методов приобретения программ, имеющий внутренний поиск в своём каталоге. Пример простого сайта по продаже ПО изображён на рисунке 5.

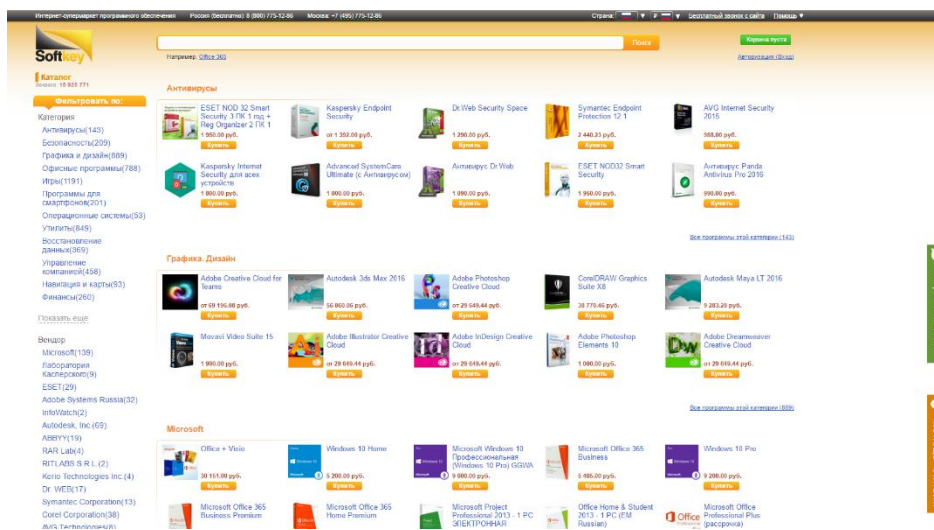


Рисунок 5 - SoftKey

Первых два варианта являются достаточно классическими и используются уже довольно длительное время. Наиболее интересным является последний вариант. Несмотря на то, что платформа Steam изначально являлась исконно игровой, то со временем в ней стали появляться и программы.

Данный сервис обладает всеми преимуществами, присущими его игровой составляющей, а именно:

- продажа предоставление покупателю разнородного ПО;
- социальные обсуждения о товарах;
- информирование о новостях, связанных с продуктом;
- контроль версий;
- и другое.

Таким образом, Steam является единственной платформой по продаже ПО в рамках Desktop-приложения, обладающей рядом преимуществ перед обычными веб-сайтами. Интерфейс платформы можно увидеть на рисунке 6.

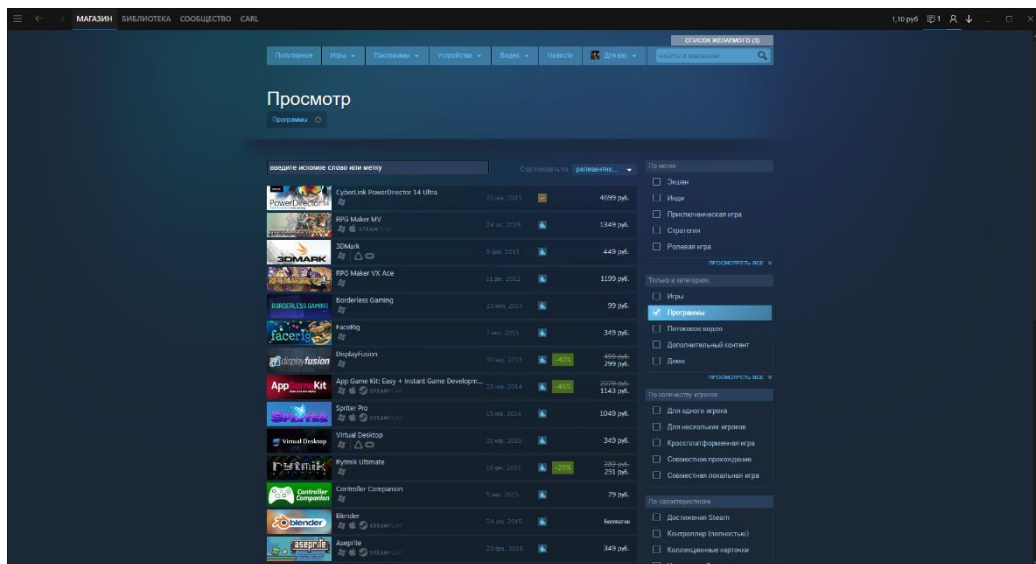


Рисунок 6 - Steam

1.5 Перспективы разрабатываемой системы

ИС «Сервис распространения цифрового лицензионного контента» будет крайне сложно составить конкуренцию текущим гигантам на рынке купли/продажи электронной продукции. Именно поэтому разрабатываемая система не будет работать только в одной сфере электронных товаров. Вместо этого она позволит объединить всё, что можно предоставить людям через интернет в одном месте.

ИС «СРЦЛК» должна обладать всеми преимуществами ранее описанных сервисов. Кроме того, в ней будут максимально упразднены все минусы тех систем, если таковы имеются.

На момент старта планируется внедрить 3 представленных выше категорий электронной продукции: фильмы, игры и программы. Но со временем будет возможно добавление и других категорий (например, книги). Для этого ей понадобится довольно гибкая архитектура, с помощью которой платформа сможет расширяться и улучшаться. В данном случае выбрана архитектура приложения MVVM (Model-View-ViewModel), но о ней будет рассказано далее.

Разрабатываемая система будет создаваться с максимальным фокусом клиентуры. Для этого, нельзя забывать о простоте и лаконичности интерфейса,

которым должна обладать система. Ведь система предназначена для абсолютно разных людей с разными интересами.

ГЛАВА 2 ПРОЕКТНАЯ ЧАСТЬ

При разработке крупных информационных систем необходимо создать некий проект (план) данной системы.

Понятие проект в наше время имеет множество определений, но главное это то, что проект – это своего рода замысел, образ, а также совокупность целенаправленных действий, предназначенных для создания целевой системы.

Проект разрабатываемой системы создавался при помощи набора инструментов Architect Enterprise 10 и Microsoft Management Studio. В данных инструментах были созданы следующие виды диаграмм:

- диаграммы сценариев использования (use case);
- диаграмма развёртывания (deployment);
- диаграммы баз данных.

Помимо диаграмм при дальнейшем проектировании и реализации использовалось сформированное техническое задание (см. Приложение А).

2.1 Диаграммы сценариев использования

Сценарий использования, вариант использования, прецедент использования (англ. use case) — в разработке программного обеспечения и системном проектировании это описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды. Система может отвечать на внешние запросы Актёра (англ. actor), может сама выступать инициатором взаимодействия. Другими словами, сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой, или что система может сделать с «кем» или «чем». Методика сценариев использования применяется для выявления требований к поведению системы, известных также как пользовательские и функциональные требования. [4]

Диаграмма вариантов использования (Use case diagram, диаграмма прецедентов) — диаграмма, на которой отражены отношения, существующие между актёрами и вариантами использования.

Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы. [4]

В системе изначально подразумевается работа следующих 6 актёров:

- гость;
- клиент;
- издатель;
- товаровед;
- тех. поддержка;
- администратор.

В дальнейшем возможно появление новых актёров (например, бухгалтер).

Краткое описание актёров представлено в таблице 1.

Таблица 1 – Выявления актёров

Актёр	Краткое описание
Гость	Пользователь, не авторизованный в системе, который имеет доступ к описанию возможностей системы и способный пройти процесс регистрации.
Клиент	Целевой покупатель в рамках системы. Обладает доступом к покупке и использованию всех товаров, продаваемых в магазине, а также средствами коммуникации с другими пользователями в рамках системы.

Окончание таблицы 1

Актёр	Краткое описание
Издатель	Публикация запросов на добавление собственного контента в систему, мониторинг статистики продаж, получение

	прибыли с продаж, поддержка актуальности контента (в зависимости от категории).
Товаровед	Обработка запросов на добавление товаров, редактирование товаров (категории), проведение акций.
Тех. поддержка	Информационная помощь всем пользователям с работой системы, получение и обработка запросов в тех. Поддержку.
Администратор	Тотальный контроль над системой, отслеживание полной статистики работы системы, управление движением денежных средств.

Для каждого актёра соответствует определённый набор возможностей, которыми он обладает в рамках работы с ИС СРЦЛК. Именно для того, чтобы наглядно описать то, что может делать каждый из них, использовались диаграммы прецедентов (или use case диаграммы).

Первая, она же основная, диаграмма – это общее представление всех модулей разрабатываемой системы. Визуально её можно увидеть на рисунок 7.

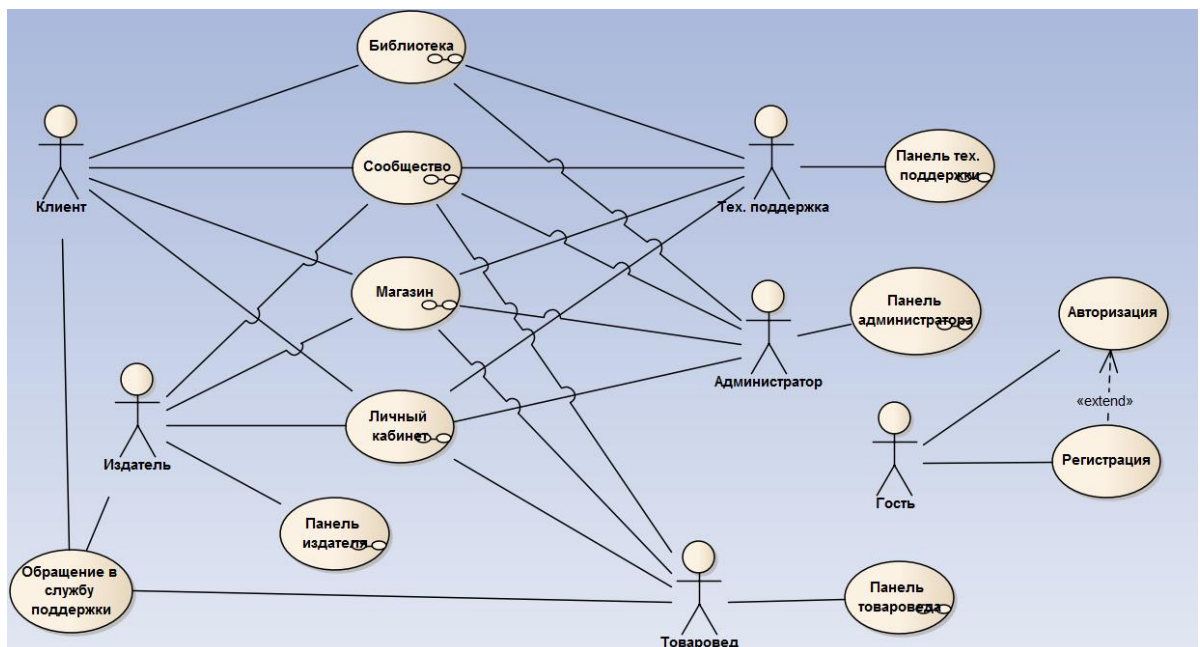


Рисунок 7 – Главная use case диаграмма

На основной диаграмме присутствуют все актёры данной системы, а также все основные модули, которыми обладает платформа, а именно:

- регистрация;
- авторизация;
- личный кабинет;
- библиотека;
- сообщество;
- магазин;
- панель администратора / тех. поддержки / товаровед / издателя;
- обращение в службу поддержки.

Модули регистрации и авторизации предназначены для Гостя. Это значит, что любой пользователь интернета может зайти в данную систему и произвести регистрацию / авторизацию в системе. Сама процедура достаточно примитивна и не требует дальнейшего разъяснения.

Остальные модули нельзя назвать примитивными и описать их в двух словах. Поэтому появляется необходимость большей конкретизации каждого модуля. С этой целью были созданы отдельные use case диаграммы, описывающие каждый модуль.

Личный кабинет – модуль, предназначенный для отображения персональной информации каждого пользователя. Use case диаграмма данного модуля представлена на рисунок 8.

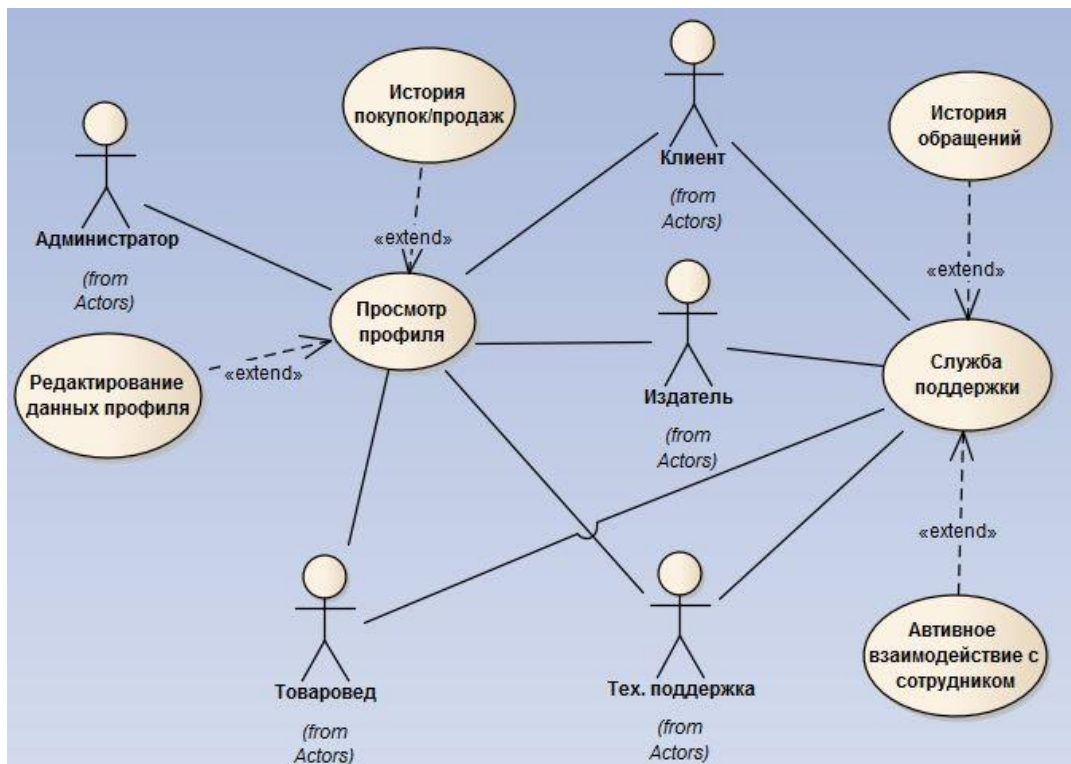


Рисунок 8 – Use case. Личный кабинет

На диаграмме видно, что все актёры данной системы в рамках личного кабинета могут просматривать профили пользователей системы. Но на данный прецедент наложено ряд ограничений.

Ограничения, наложенный на прецедент «Просмотр профиля», а также его расширения «Редактирование данных профиля» и «История покупок», следующие:

- в чужом профиле отображена лишь общедоступная информация, или информация, которую позволил видеть каждый конкретный пользователь;
- каждый пользователь может редактировать только редактируемые поля своего профиля;
- доступ к полному редактированию всех данных любого пользователя системы имеют только администратор и сотрудник тех. поддержки;
- история покупок/продаж отображает лишь ту информацию, с которой связан конкретный пользователь, но администратор и сотрудник тех.

поддержки могут просматривать подробную информацию обо всех пользователях системы.

Прецедент «Служба поддержки» предназначен для решения конфликтных ситуаций, возникших у всех пользователей системы. В его рамках через личный профиль все пользователи смогут в режиме реального времени отслеживать статусы своих обращений в службу, а также взаимодействовать с сотрудником для решения проблемных ситуаций. Как и в ситуации с «Просмотром профиля», системные учётные записи (администратор, сотрудник тех. поддержки) обладают большими правами и могут просматривать истории обращений всех пользователей в системе, когда остальные учётные записи могут просматривать лишь персональные обращения.

Данный модуль не решает ни одну из задач, поставленных разрабатываемой системе. Но он является неотъемлемой частью любой системы, поэтому наличие «Личного кабинета» необходимо.

Следующий модуль наиболее интересен, поскольку он является центральным в функционировании все системы. Название данного модуля «Магазин». Use case данного модуля представлен на рисунок 9.

скрытые/не опубликованные/временно заблокированные товары, а также отслеживать всю статистику продаж, касающуюся его публикаций.

Прецедент «Публикация товаров» показывает, что для публикации товара в каталог магазина необходимо, чтобы данный товар прошёл верификацию товароведа. Это значит, что «Издателю» необходимо выполнить следующие шаги.

1. Заполнить/изменить сведения о товаре, а также медиа файлы (иконки, главное изображение, скриншоты, видео и другие), описывающие товар.

2. Отправить запрос на добавление. При этом товар будет иметь статус «В обработке».

3. При успешном прохождении проверки «Товароведа» товар публикуется в магазине. В ином случае «Издателю» необходимо устранить найденные несоответствия с требованиями (переход к пункту 1).

Последний, но при этом самый важный прецедент – «Покупка товаров». Он доступен только целевому покупателю («Клиент»). Реализация покупки ничем не отличается от существующих аналогов, а если быть более точным – это наполнение корзины покупок с последующим проведением оплаты.

Таким образом, модуль «Магазин» позволяет решить первые 3 задачи, поставленные разрабатываемой системе, а именно:

1. покупка/продажа цифрового контента;
2. аккумулярование разностороннего цифрового контента в одном месте;
3. предоставление полной информации о каждом товаре пользователю.

Следующий основной модуль разрабатываемой системы называется «Библиотека». Данный модуль, собственно, предоставляет клиентам системы пользоваться всем приобретённым контентом. Наглядное представление библиотеки можно увидеть на рисунок 10.

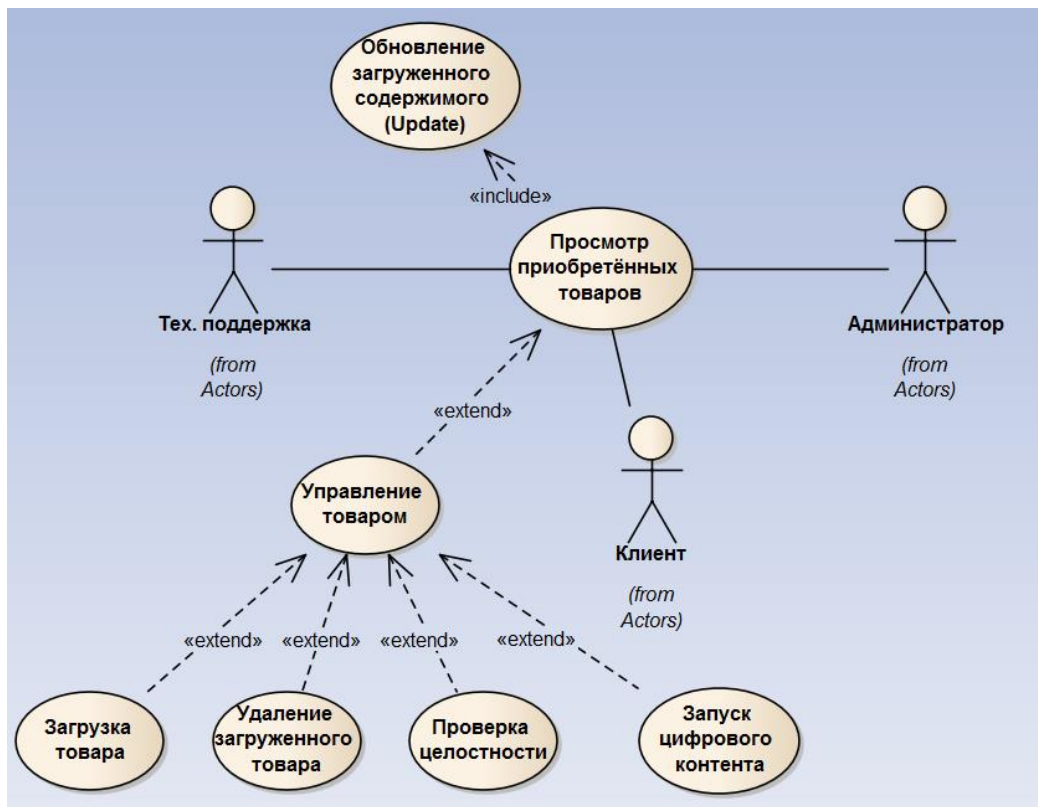


Рисунок 10 – Use case. Библиотека

Основной прецедент рассматриваемой диаграммы – «Просмотр приобретённых товаров». На него навешаны следующие ограничения:

- «Клиент» имеет доступ только к товарам, приобретённым им самим;
- «Тех. поддержка» и «Администратор» не имеют личной библиотеки, но могут просматривать и управлять библиотеками клиентов системы.

Как можно заметить, при инициализации библиотеки будет происходить проверка наличия обновлений каждого купленного и загруженного товара. Это необходимо для поддержания актуальности контента. Соответственно, при просмотре товаров «Клиент» может управлять каждым из них. Действия, которые может выполнять «Клиент» с товарами:

- загрузка на локальный носитель;
- удаление с локального носителя;
- запуск;
- проверка целостности.

Все последующие модули – управляющие панели. Для каждой из них отведена определённая перечень возможностей, предоставляемая каждому конкретному типу пользователей. При этом, для их реализации может быть задействовано стороннее решение, но основные возможности приведены на следующих диаграммах (возможности могут быть не полными).



Рисунок 11 – Use case. Панель издателя



Рисунок 12 – Use case. Панель товароведа

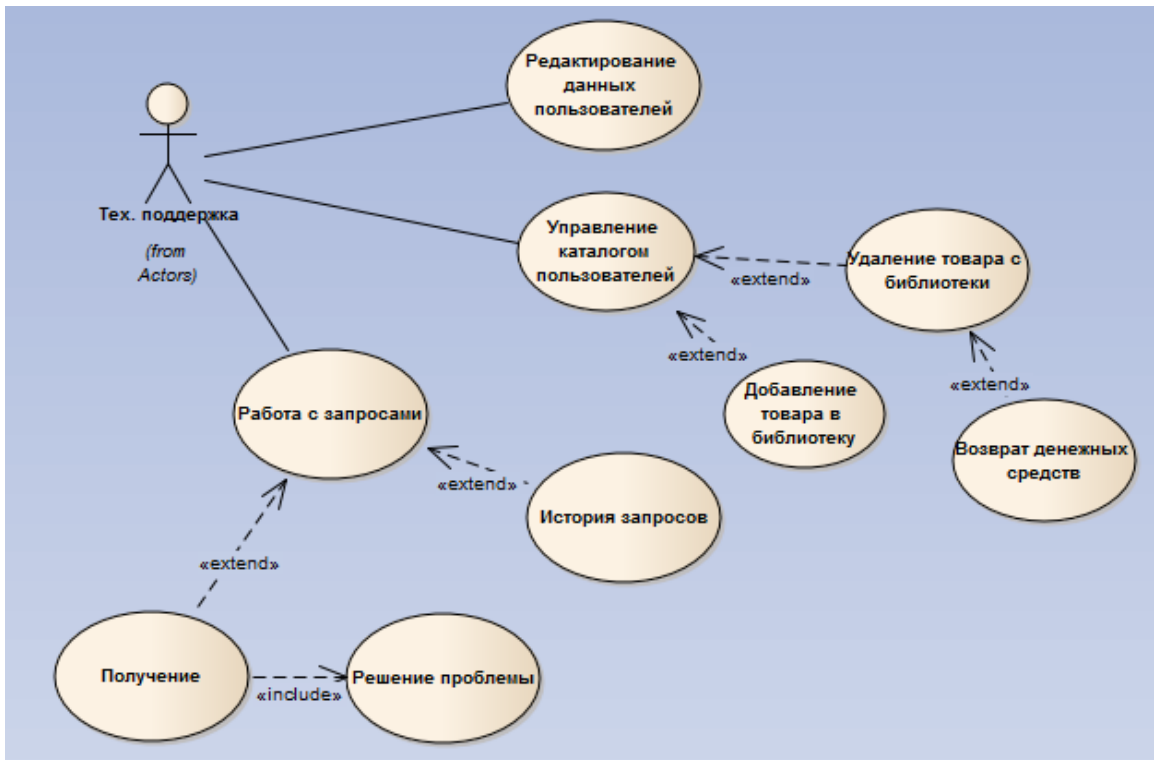


Рисунок 13 – Use case. Панель тех. поддержки

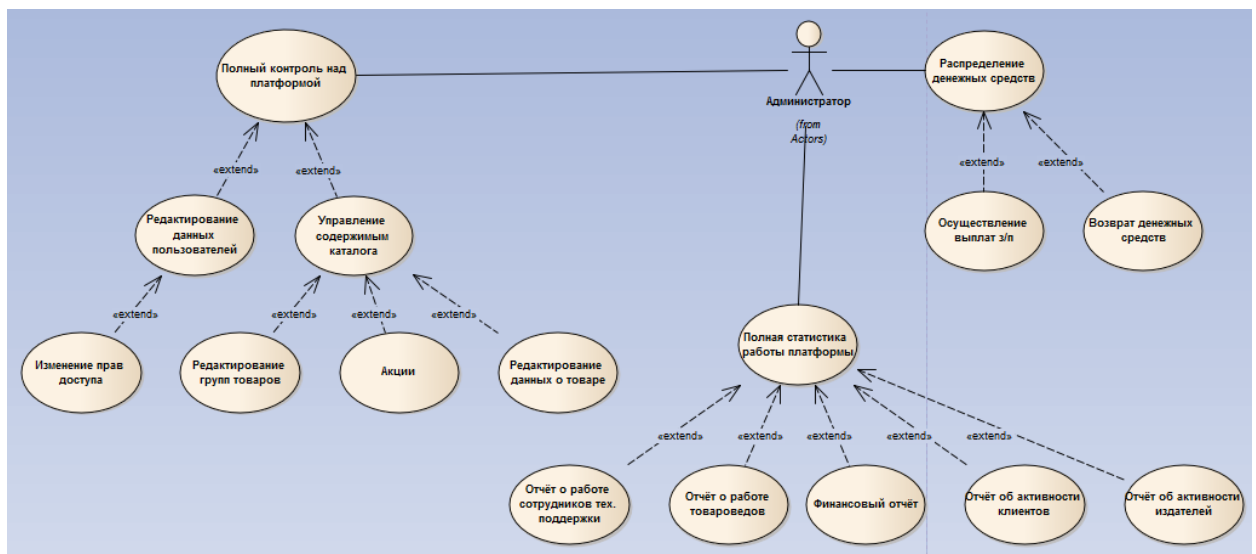


Рисунок 14 – Use case. Панель администратора

На приведённых выше диаграммах представлены лишь основные функции, выполняемые каждым из пользователей системы (кроме «Клиента»). Для реализации данных модулей возможно использование существующих решений, которые будут способны на большее, чем приведено в Use Case диаграммах.

2.2 Диаграмма развёртывания

Диаграммы развёртывания отражают соответствие конкретных программных артефактов (например, выполняемых файлов) вычислительным узлам (выполняющим обработку). Они показывают размещение программных элементов в физической архитектуре системы и взаимодействие (обычно сетевое) между физическими элементами. На рисунке 15 наглядно показана физическая архитектура (или архитектура развёртывания) разрабатываемой системы. [1]

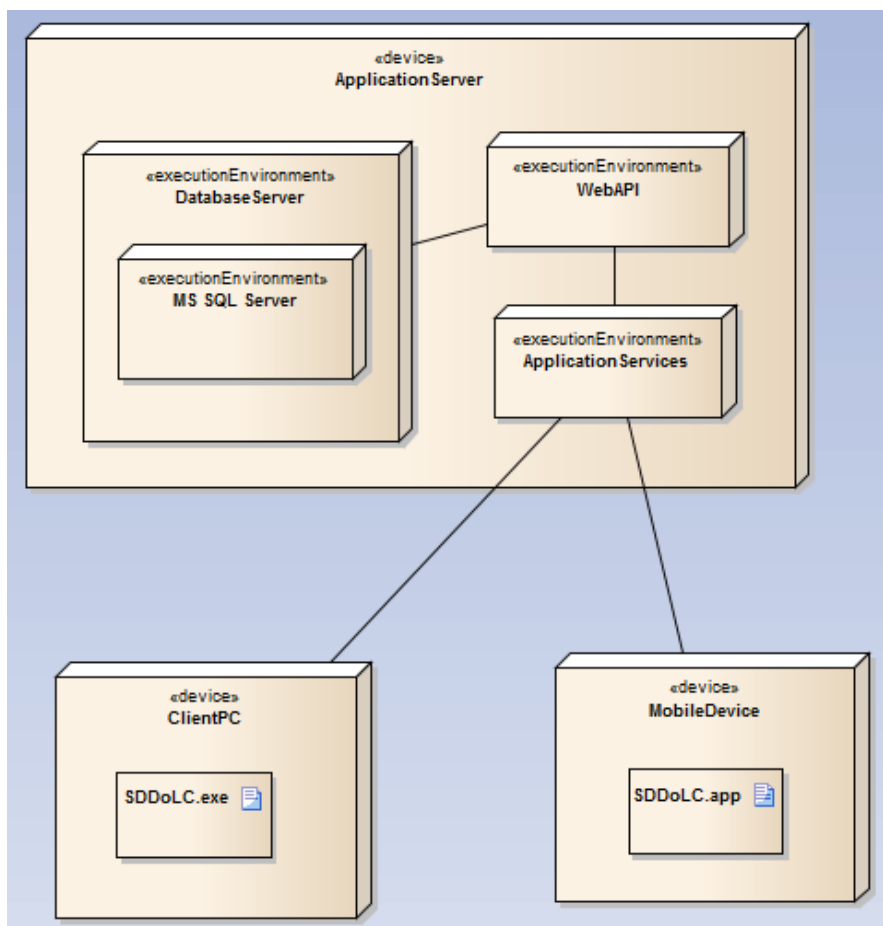


Рисунок 15 – Диаграмма развёртывания

По диаграмме можно увидеть, что физическая архитектура разрабатываемой системы является двухзвенной. Это значит, что система состоит из 2-х составных частей: клиентской и серверной.

Серверная часть содержит в себе как сервисы приложения, так и базу данных на одной физической машине. Клиентская часть будет представлена в виде клиента для PC (в будущем для iOS, Android и др.).

2.3 Диаграммы баз данных

База данных – это неотъемлемая часть любой разрабатываемой системы, поскольку от неё целиком и полностью зависит будущее системы. База данных хранит в себе всю информацию, связанную с работой системы, а именно: информацию о пользователях системы, какие-либо статистические данные, данные о товарах и многое другое.

При построении базы данных необходимо руководствоваться основными правилами нормализации. Под нормализацией понимается рассредоточение информации крупных таблиц по нескольким небольшим с целью исключения избыточности данных. [2]

Естественно, при этом приходится пересматривать отношения между таблицами, так как появляются новые сущности. Этапы нормализации носят название форм. Теоретически база данных может принадлежать к одной из пяти так называемых нормальных форм, естественно, если в ней вообще был начат процесс нормализации. Таким образом, проектирование базы данных — итерационный процесс: одни и те же действия выполняются снова и снова, пока не будут получены удовлетворительные результаты. В то же время полной нормализации базы данных обычно не проводят. Это вызвано тем, что все нормальные формы после третьей вызывают существенное снижение производительности (поскольку разделение таблиц приводит к созданию дополнительных объединений в запросах). К тому же иногда сложно определить направления дальнейшего разделения данных для приведения базы к четвертой и пятой нормальным формам. [2]

Нормальной формой называют набор правил, применяемых к таблице и гарантирующих отсутствие избыточности данных (не только дублирования

информации, но и данных, которые могут быть вычислены на основании уже существующих). К примеру, если хранить в базе дату рождения сотрудника, то бессмысленно наряду с этим хранить и его возраст (причем последний столбец придется обновлять каждый год). [2]

К разрабатываемой базе применялись первые 3 нормальные формы. Первая нормальная форма избавляет базу от повторяющихся данных. Во второй нормальной форме исключается избыточность данных. В третьей нормальной форме требуется, чтобы не существовало атрибутов (полей), которые зависят от других неключевых атрибутов.

В системе планируется ввести 2 (пользователи и товары). иерархически организованных структуры данных. Это значит, что у каждой из них имеется некий основной класс, у которого присутствует множество наследуемых классов. При этом в базе такие данные можно хранить в соответствии с одной из 3-х следующих стратегий: Table per Hierarchy (TPH), Table per Type (TPT), Table per Concrete class (TPC).

Table per Hierarchy (TPH). Этот подход предполагает одну таблицу для всей иерархии наследования классов. Таблица включает в себя столбец дискриминатора, который отличает классы наследования. Эта стратегия наследования используется по умолчанию в Entity Framework. Пример представлен на рисунке 16. [3]

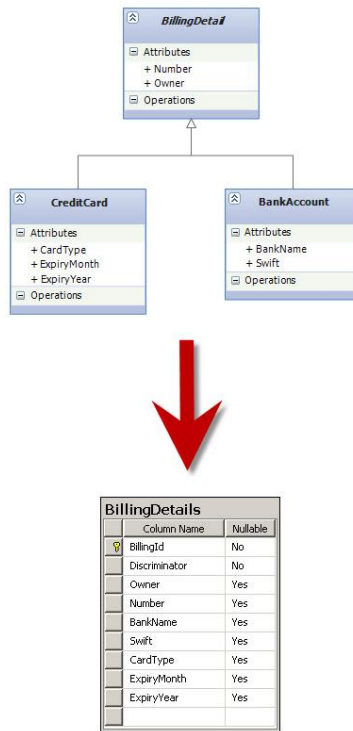


Рисунок 16 – Пример ТРН

Table per Type (TPT): Этот подход предполагает отдельную таблицу для каждого класса. Пример представлен на рисунке 17. [3]

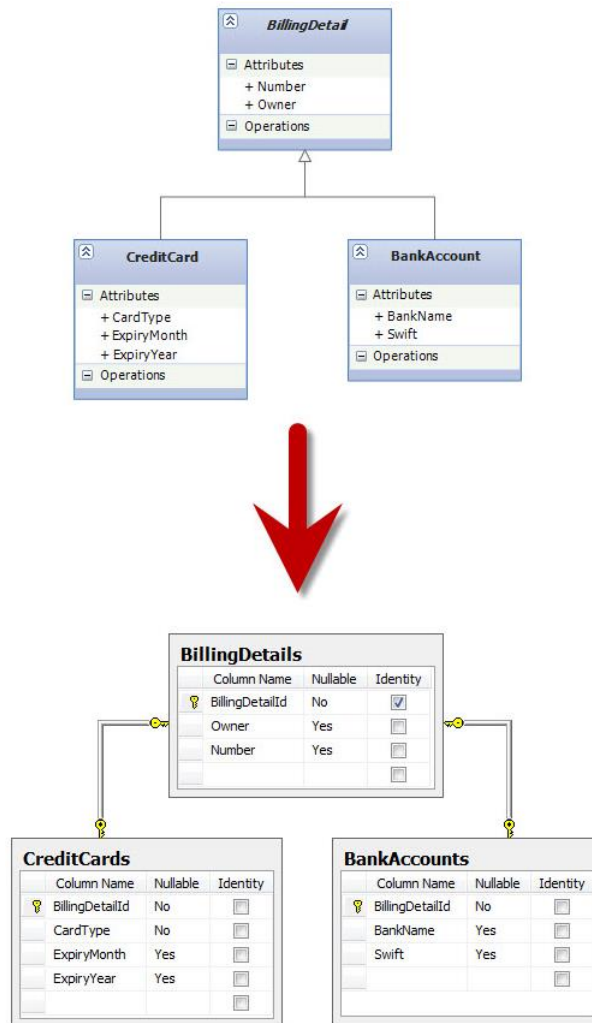


Рисунок 17 – Пример ТРТ

Table per Concrete class (TPC): Этот подход предполагает одну таблицу для одного конкретного класса, но не для абстрактного класса. Итак, если вы наследует абстрактный класс в нескольких конкретных классах, то свойства абстрактного класса будут частью каждой таблицы конкретного класса. Пример представлен на рисунке 18. [3]

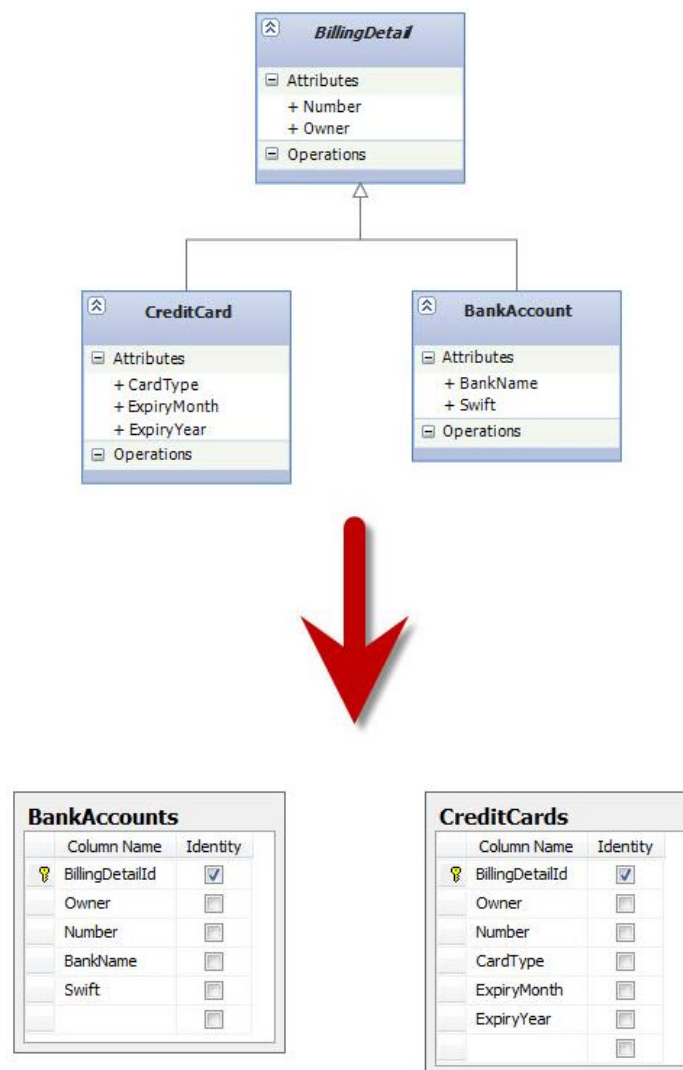


Рисунок 18 – Пример ТРС

Для начала рассмотрим те стратегии, которые не подходят для разрабатываемой системы. Стратегия ТРН подразумевает создание достаточно громоздкой таблицы, в которой, при развитии системы, в последствии можно будет запутаться, поскольку свойств у сущностей будет довольно много, а разбираться в таблице из более чем «100» полей достаточно тяжело. Стратегия ТРТ также крайне нерекомендуема для разрабатываемой системы, поскольку её использование сильно затруднит доступ к данным. Например, нам нужен список всех товаров в системе, а сделать это можно лишь путём объединения нескольких запросов через UNION. Кроме того, при необходимости изменить общие свойства, стратегия требует внесения изменений в каждую таблицу в отдельности.

Поэтому наиболее перспективной была выбрана стратегия ТРТ, поскольку только с её помощью система будет способна к расширению в будущем.

Первая и основополагающая диаграмма описывает пользователей, с которыми будет работать разрабатываемая система. Она представлена на рисунке 19.

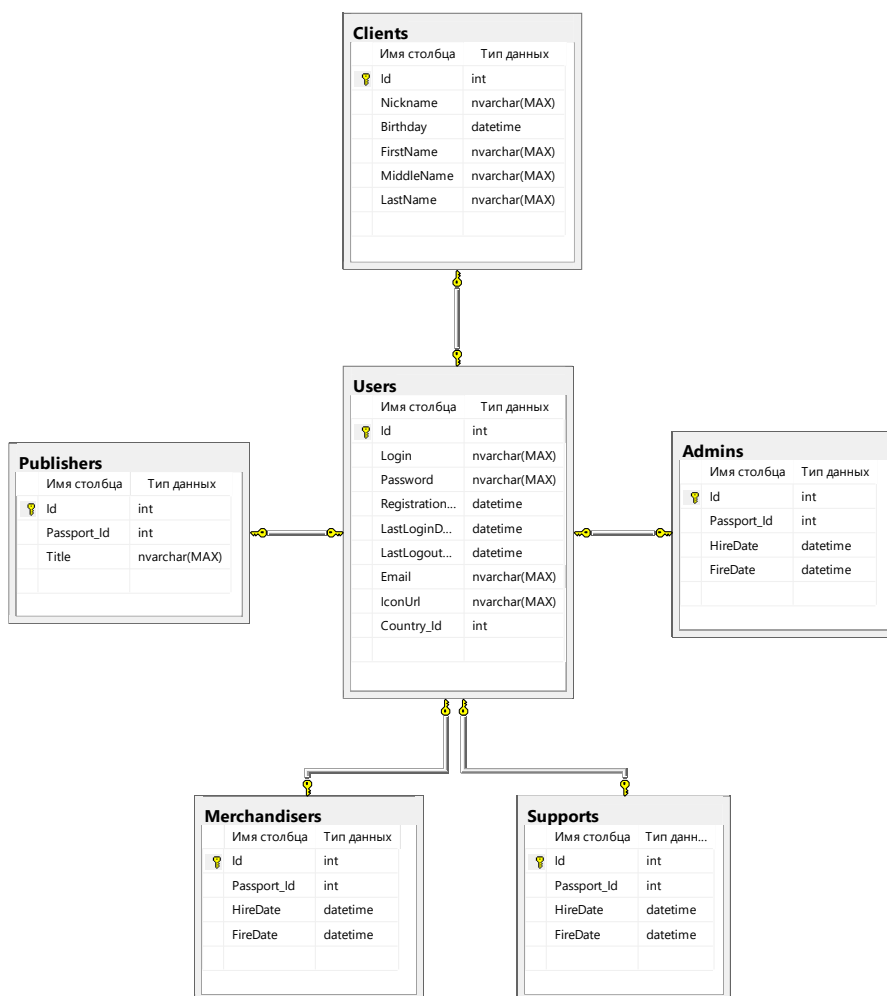


Рисунок 19 – Диаграмма БД. Пользователи системы

На приведённой выше диаграмме можно заметить, что в базе имеется один общий базовый класс **Users**, который описывает свойства, общие для всех категорий пользователей. Каждый же конкретный «участник» системы имеет отдельную таблицу со свойственными ему полями. Например, поля **Login** и **Password** – это свойства, которыми должны обладать все пользователи системы, а вот **Birthday** – это специфичное поле, использование которого сотрудниками

системы не является необходимым. И наоборот, клиенту нет необходимости заверять свои паспортные данные в системе. Паспортные данные будут содержать отдельная таблица Passports, на которую ссылается поле Passport_Id таблицы Users. Таблица Passport представлена на рисунке 20.

Passports	
Имя столбца	Тип данных
Id	int
Number	nvarchar(MAX)
PublishDate	datetime
PublishBy	nvarchar(MAX)
Registration	nvarchar(MAX)
Birthday	datetime
FirstName	nvarchar(MAX)
MiddleName	nvarchar(MAX)
LastName	nvarchar(MAX)

Рисунок 20 - Диаграмма БД. Паспортные данные

Организация таблиц для товаров, продаваемых в системе, выглядит аналогичным образом, как и для пользователей. Диаграмму можно увидеть на рисунке 21.

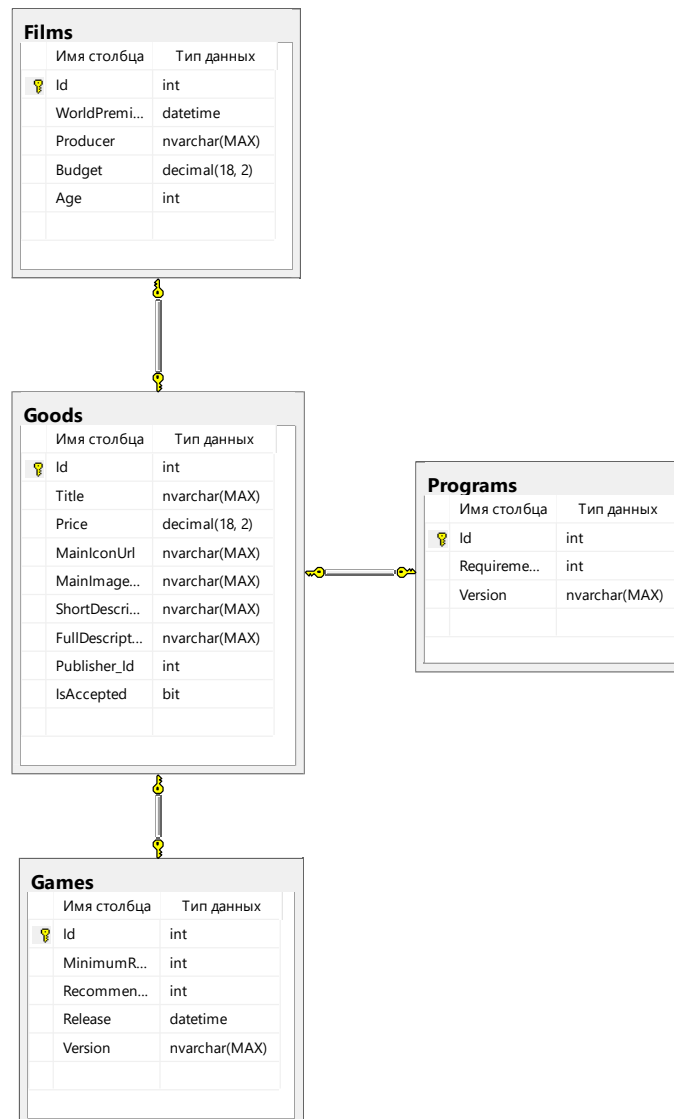


Рисунок 21 – Диаграмма БД. Товары

Примечание. По мере развития приложения структура приведённых выше таблиц может расширяться/изменяться.

Для организации непосредственной продажи контента конечному потребителю необходима вспомогательная таблица Orders, которая представлена на рисунке 22.

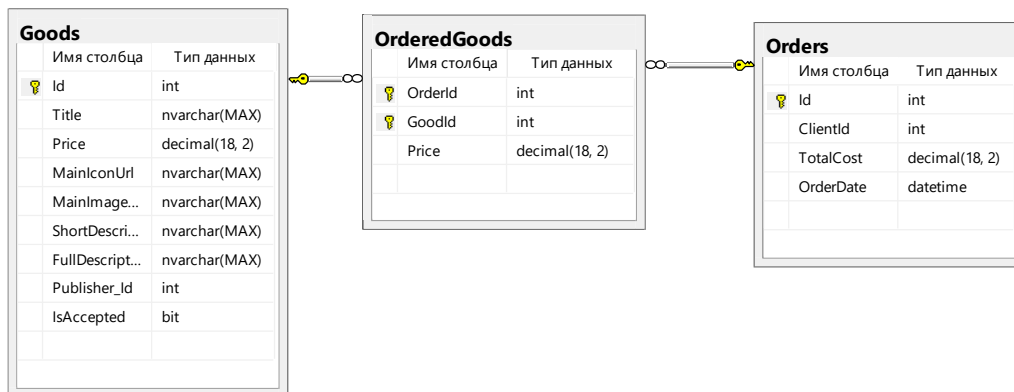


Рисунок 22 - Диаграмма БД. Заказы

На диаграмме можно заметить, что для установления связи между таблицами Orders и Goods была создана вспомогательная таблица OrderedGoods. Это необходимо, в первую очередь, для установления связи «Многие ко Многим» между Goods и Orders. Это значит, что в одном заказе может быть множество товаров, а один товар может быть во множестве заказов. Помимо установления данной связи, данная таблица может давать более детальную информацию, касающуюся и товара, и заказа одновременно. Например, цену товара в момент продажи, поскольку она может меняться (скидки, акции и другое).

Подобным образом организована связь между клиентами и приобретёнными ими товарами через таблицу Libraries (рисунок 23)

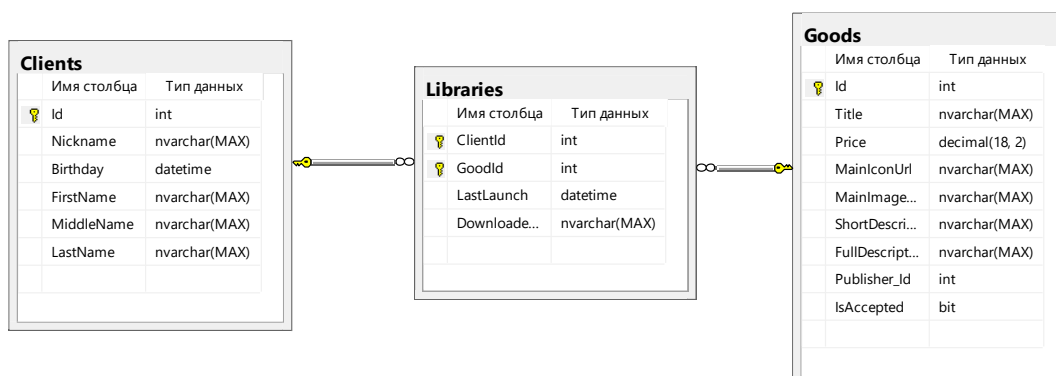


Рисунок 23 – Диаграмма БД. Библиотеки

Следующие 2 таблицы не имеют прямой связи с Users, поскольку это может создать в базе циклы, которых лучше не нужно создавать. Но наличие данных таблиц обязательно, поскольку функционал, который они могут

предоставить, необходим разрабатываемой системе. А именно, сообщения между пользователями (таблица Messages) и список контактов внутри системы (таблица Friends). Данные таблицы представлены на рисунке 24.

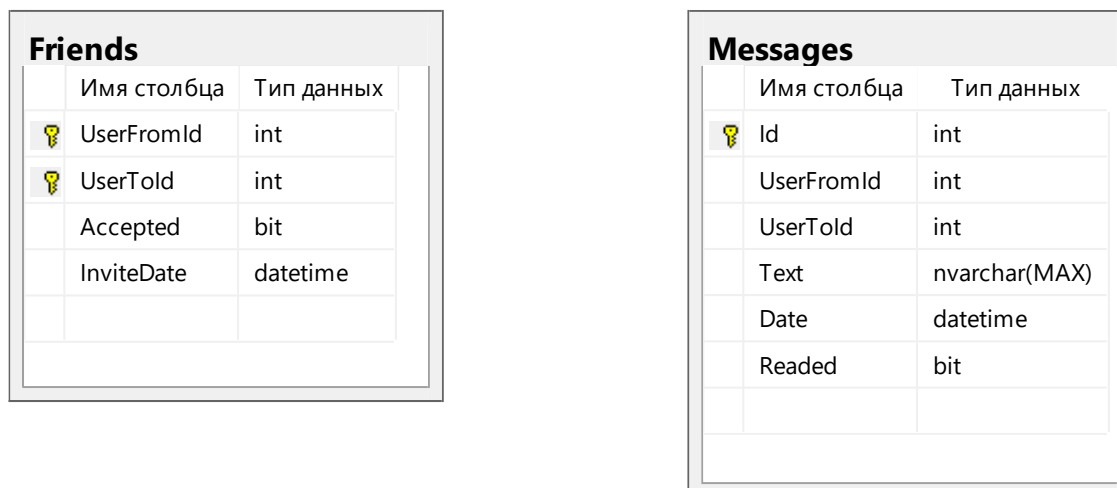


Рисунок 24 – Диаграмма БД. Сообщество

Для организации системы ролей будет использована система Permissions. Она позволяет определить ряд действий, которыми обладает каждая роль в системе. Имена ролей в системе соответствуют всем категориям пользователей (пользователь «Клиент» - роль «Клиент» и т.д.).

Именно для её реализации были добавлены следующие таблицы, представленные на рисунке 25.

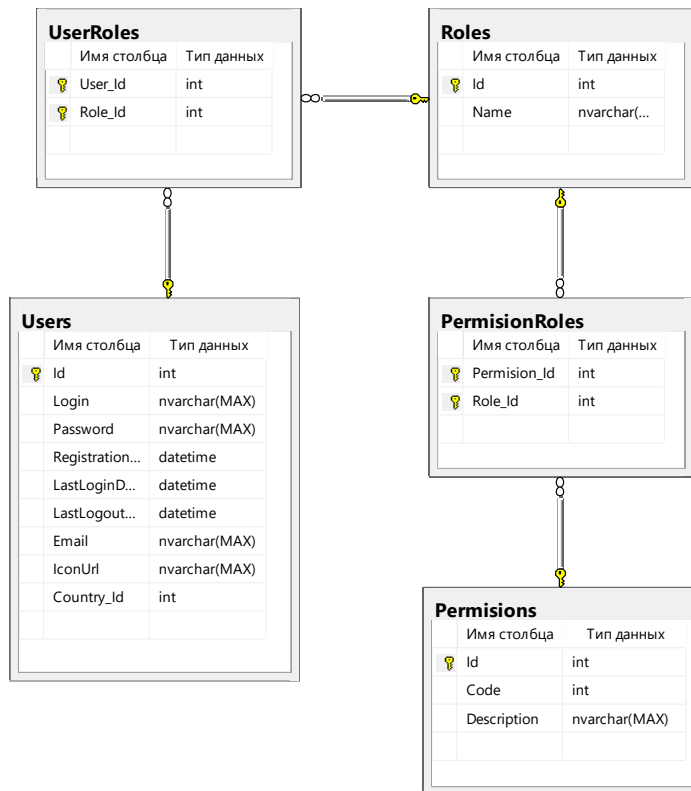


Рисунок 25 – Диаграмма БД. Роли

Поскольку в системе подразумевается отправка содержательных запросов товароведам (с целью добавления нового товара) или же запросов в службу технической поддержки (для решения возникших проблем), то для этих запросов необходимо подготовить структуру, которая представлена на рисунке 26.

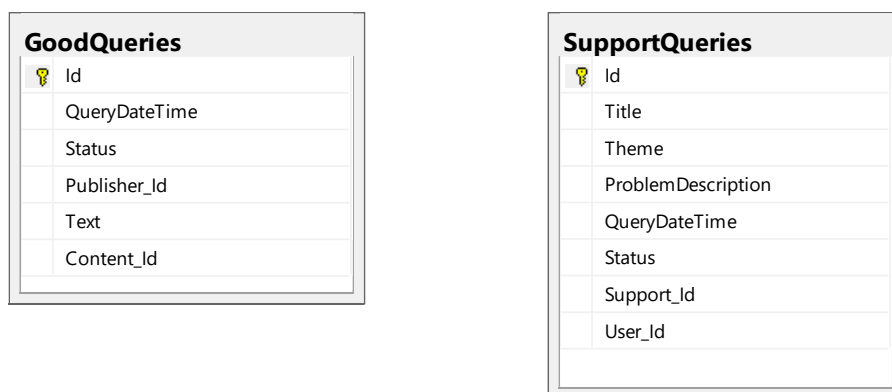


Рисунок 26 – Диаграмма БД. Запросы

ГЛАВА 3 РЕАЛИЗАЦИЯ

3.1 Шаблон проектирования

Первое, что необходимо сделать перед непосредственно разработкой ИС «Сервис распространения цифрового лицензионного контента» - выбрать паттерн (шаблон) проектирования разрабатываемого приложения.

На рассмотрении имеется 2 шаблона: MVC и MVVM.

MVC (Model-View-Controller). Model-View-Controller представляет собой шаблон проектирования, который отделяет различные части программного обеспечения. Такое разделение способствует повторному использованию кода и более структурированной архитектуры приложения. Так, например, вы могли бы разбить сайт на 3-х основных компонента.

- Первый, "View" – это то, что представлено пользователю. Это, как правило, HTML, CSS и JavaScript.
- Второй, "Model" описывает то, как веб-сайт взаимодействует с базой данных.
- Третий, "Controller" - это то, что соединяет "View" и "Model". Таким образом, когда пользователь запрашивает что-то из базы данных, "Controller" принимает этот запрос от "View" и отправляет его в "Model".

Второй паттерн - MVVM (Model-View-ViewModel). В MVVM, как следует из названия, содержится 3 основных компонента: Model, View и ViewModel. Каждый предлагает отличную и отдельную роль. На приведенном ниже рисунке 27 показаны отношения между этими тремя компонентами.

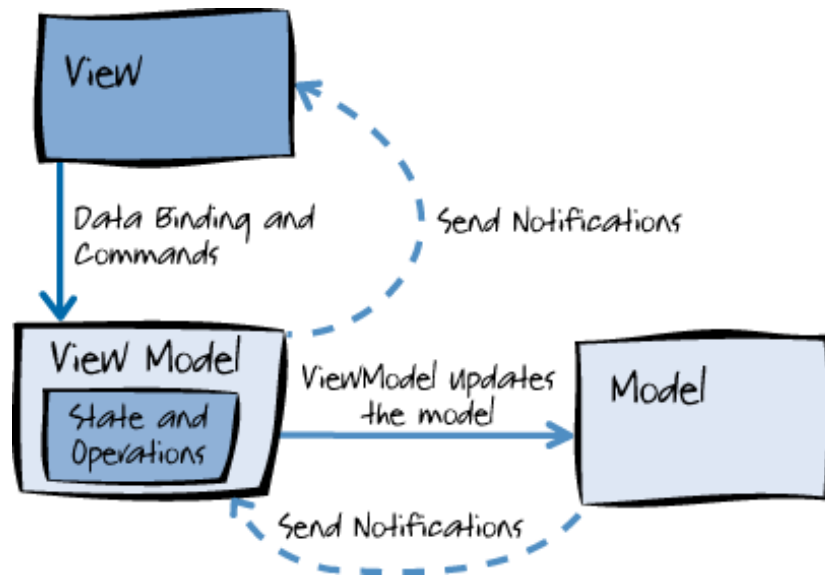


Рисунок 27 – Паттерн MVVM

View. View отвечает за определение структуры, компоновку и внешний вид того, что пользователь видит на экране. В идеале, View не содержит бизнес-логику приложения.

Model. Model в MVVM является реализация модели предметной области приложения, которая включает в себя модель данных вместе с business и validation логикой. Примеры модельных объектов включают в себя репозитории (хранилища), бизнес-объекты, объекты передачи данных (DTO), Plain Old CLR Objects (POCOs), сгенерированные сущности и прокси-объекты.

ViewModel выступает в качестве посредника между View и Model, и отвечает за обработку логики представления. Как правило, ViewModel взаимодействует с Model путем вызова методов в классах Model. Затем ViewModel предоставляет данные из Model в форме, которую View может легко использовать. ViewModel извлекает данные из Model, а затем делает данные доступными для View, и может переформатировать данные, что делает для View обработку проще. ViewModel также предоставляет реализации команд, которые пользователь приложения инициирует во View.

Среди разобранных выше паттернов, выбор был сделан на паттерне MVVM, поскольку он имеет следующий ряд преимуществ.

1. В процессе разработки, разработчики и дизайнеры могут работать более независимо друг от друга и одновременно над их компонентами. Дизайнеры могут сосредоточиться на View, и если они используют Expression Blend, они могут легко генерировать выборки данных для работы, в то время как разработчики могут работать над ViewModel и Model.

2. Разработчики могут создавать модульные тесты для ViewModel и Model без использования View. Модульные тесты для ViewModel могут осуществлять точно такую же функциональность, которая используется View.

3. Легко изменить дизайн пользовательского интерфейса приложения, не касаясь кода, так как View реализуется полностью в XAML. Новая версия View должна работать с существующей ViewModel.

3.2 Архитектурное решение

С программной точки зрения была выбрана архитектура MVVM, о выборе которой было рассказано ранее. Согласно данному паттерну, всё приложение разбивается на несколько частей: View, ViewModel и Model. Все перечисленные слои могут иметь только однонаправленную зависимость. Это значит, что View может ссылаться на ViewModel, но ViewModel не может ссылаться на View. Именно благодаря этому становится возможным независимая разработка различных компонентов системы, а также модульное тестирование.

Кроме программной архитектуры необходимо определиться и с аппаратной. (или физической). Приложение является клиент-серверным, что вызывает необходимость в использовании 2-х и более физических устройств: клиент и сервер. Почему их может быть более 2-х? Подразумевается, что сервер – может быть не единственным, а разбитым на несколько составных частей: сервер БД, сервер приложений и другие. На момент запуска в реальных условиях ИС «Сервис распространения цифрового лицензионного контента» вполне вероятно потребует наличие более одного сервера, поскольку нагрузка от приложения будет колоссальная. Но на момент проектирования достаточно

использование одного выделенного сервера, на котором будут работать все службы приложения: Microsoft SQL Server, EntityFramework/WebAPI и Сервер приложений. Их разделение в последствии не должно вызвать больших затруднений при реализации.

Таким образом, разрабатываемое приложение имеет двухзвенную структуру (1 клиент и 1 сервер) и 3-х уровневую архитектуру MVVM.

Solution проекта содержит все 3 уровня архитектуры MVVM: View, ViewModel и Model. Помимо данных 3-х слоёв присутствует вспомогательный слой DataAccess, который является дополнительным связующим звеном между ViewModel и Model посредством использования интерфейсов. Именно DataAccess позволяет подменять модель данных, не трогая саму логику приложения. Например, на данный момент в приложении используется EntityFramework, который впоследствии будет заменён на WebApi без внесения значительных изменений в само приложения.

3.3 Интерфейс приложения

При старте приложения для пользователя откроется стандартное окно авторизации, на котором от него потребуются ввод идентификационных данных (рисунок 28), после чего выполняется вход в систему. В случае, если пользователь заходит в первый раз, от него потребуются пройти процесс регистрации, окно которого представлено на рисунке 29.

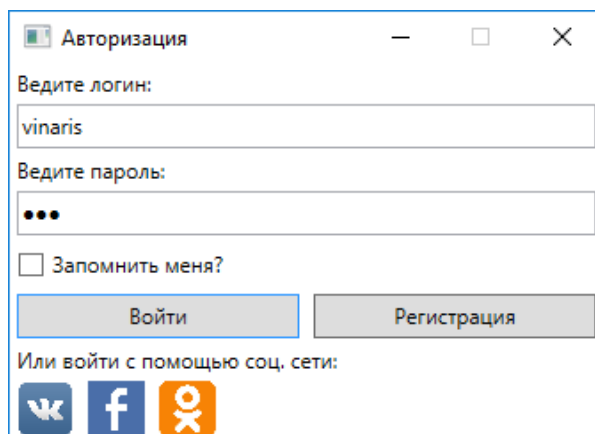


Рисунок 28 – Окно авторизации

Регистрация

Заполните следующую форму:

Логин:

Пароль:

Имя:

Фамилия:

Отчество:

Дата рождения: 31.05.2016

E-mail:

Страна:

Я принимаю условия Пользовательского соглашения

Создать аккаунт

Рисунок 29 – Окно регистрации

Форма регистрации в разрабатываемой системе имеет вполне себе стандартный вид, как и большинство других систем. А именно, она содержит в себе следующие поля для ввода: логин, пароль, имя, фамилия, отчество, дата рождения, электронная почта и страна проживания. Помимо заполнения приведённых выше данных, пользователь обязан при регистрации согласиться со всеми правилами пользовательского соглашения, которое будет решать все юридические вопросы, связанные с хранением данных, движением денежных средств, обязанностями сторон и так далее. На процесс регистрации наложены лишь 2 ограничения:

- уникальность логина;
- уникальность электронной почты.

Только при отсутствии дубликатов данных свойств в базе, пользователь сможет завершить процесс регистрации.

После успешного входа в систему отрывается главное окно приложения, представленное на рисунке 30.

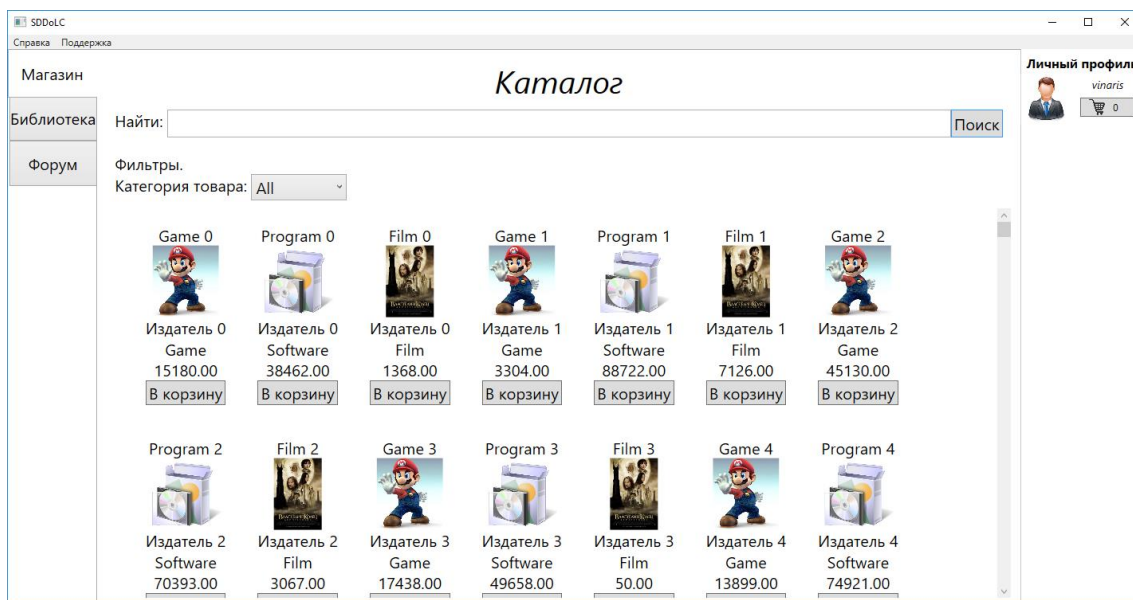


Рисунок 30 – Главное окно приложения

Общая структура окна состоит из 3-х основных разделов: верхнее меню, личный профиль вместе со списком контактов в правой части приложения и основной контент, который занимает всю оставшуюся часть окна.

На текущий момент главная страница – это сам каталог магазина. Каталог изначально загружает все товары, продаваемые в системе. Но при необходимости весь контент можно фильтровать. Поле «Найти» предназначено для ввода поисковой строки, благодаря которой будет осуществляться поиск в каталоге. Поиск осуществляется по следующим свойствам: наименование, издатель, цена и описание. Также присутствует фильтр поиска по категории контента. Поиск со временем будет расширяться добавлением новых фильтров, а также методов сортировки отображаемых товаров.

При просмотре каталога, кроме общих сведений о товарах, клиент сможет получить более детальную информацию о каждом конкретном товаре. При этом, описание товара зависит от его категории, поскольку перечень описываемых критериев у различных товаров отличителен друг от друга. На рисунках 31, 32 и 33 можно увидеть примеры детальных описаний для каждой категории товара.

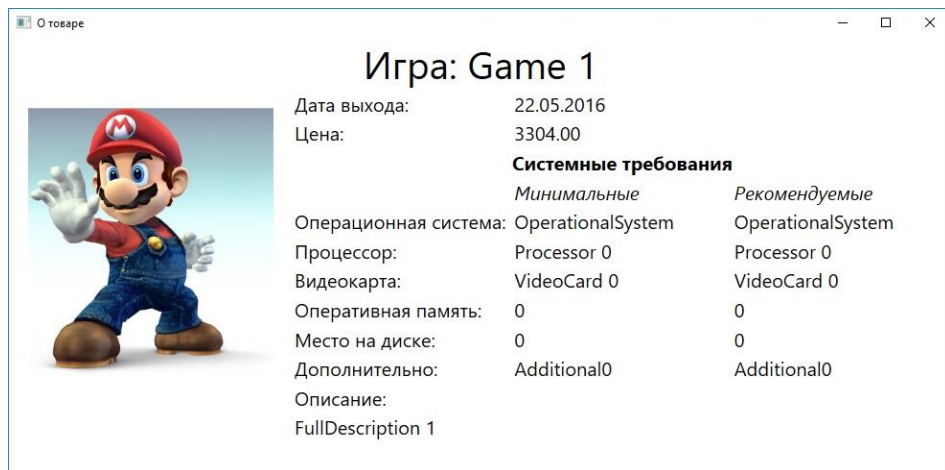


Рисунок 31 – Окно деталей об игре

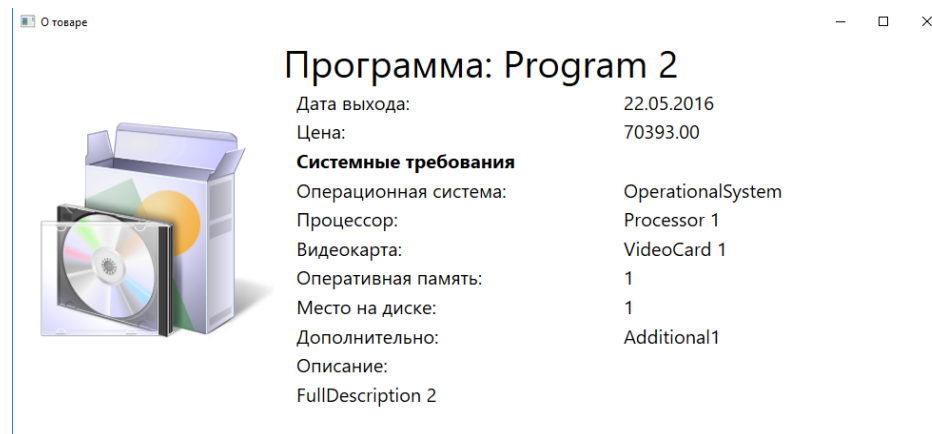


Рисунок 32 – Окно деталей об программе



Рисунок 33 – Окно деталей об фильме

В данный момент детали о товарах имеют достаточно схожий вид, но в перспективе описание каждого из них можно изменять в отдельности.

Одна из основных задач системы – это продажа контента. Она реализована при помощи корзины товаров. Это значит, что клиент, просматривая каталог магазина, может добавлять любые понравившиеся товары в «Корзину», после чего произвести оплату и получить приобретённый контент. Корзина представлена на рисунке 34.

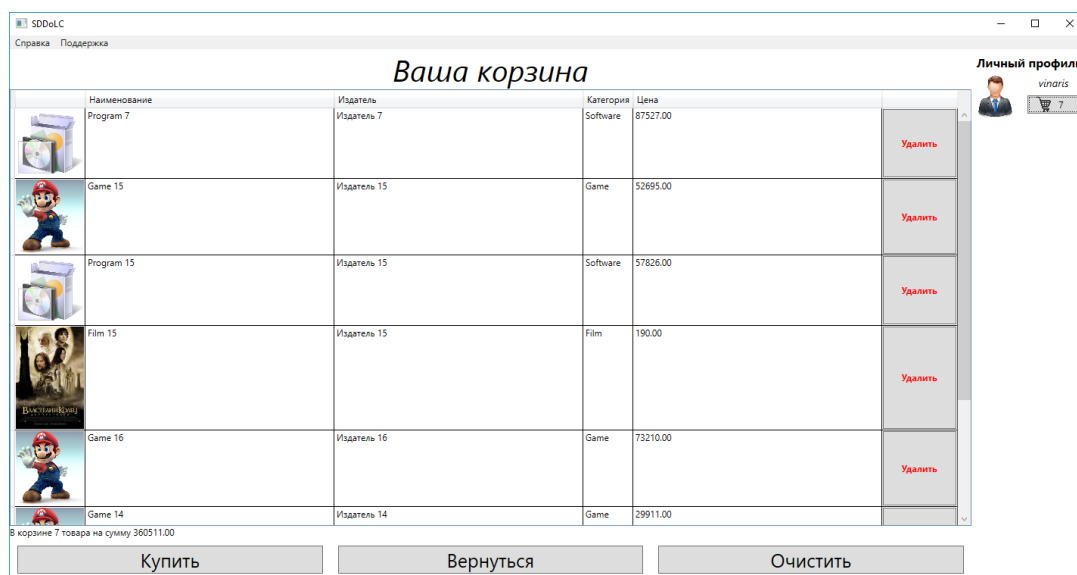


Рисунок 34 – Корзина товаров

Как видно на рисунке выше, корзина поддаётся дополнительному редактированию: удаление конкретного наименования, полная очистка корзины. При вызове метода покупки подразумевается вызов платёжной операции, завершением которой является подтверждение об оплате, после чего корзина очищается, а приобретённые товары отправляются в библиотеку к клиенту (рисунок 35).

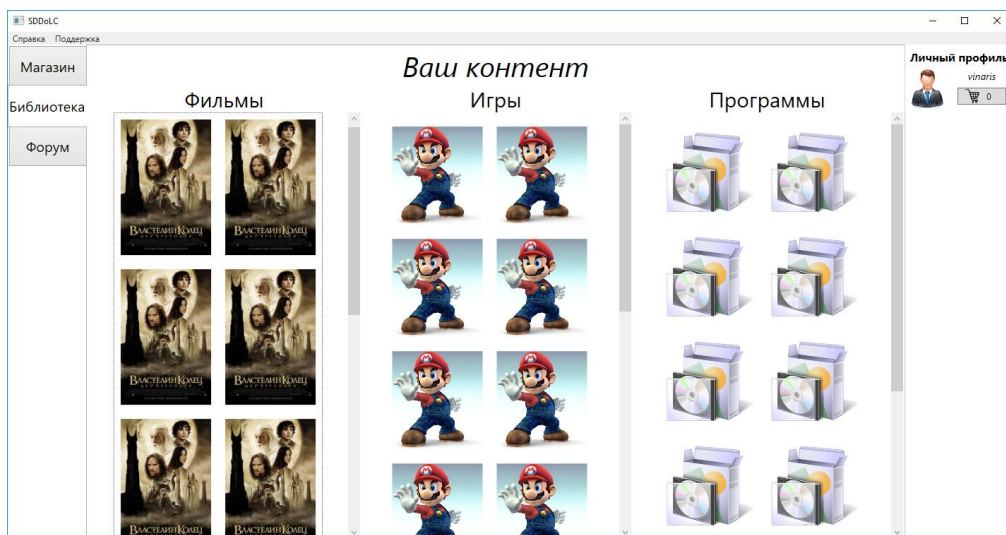


Рисунок 35 – Библиотека клиента

Поле библиотеки разбито на 3 блока, которые соответствуют каждой из 3-х предложенных категорий товаров: фильмы, игры и программы. Именно отсюда клиент сможет «управлять» приобретённым им контентом. Это управление реализовано через контекстное меню, которое для каждого товара может отличаться. Для программ и игр оно имеет схожие пункты: запуск, обновление, загрузка, проверка целостности и об игре/программе. Для фильмов: смотреть и о фильме. Контекстное меню для всех категорий товаров можно увидеть на рисунке 36.

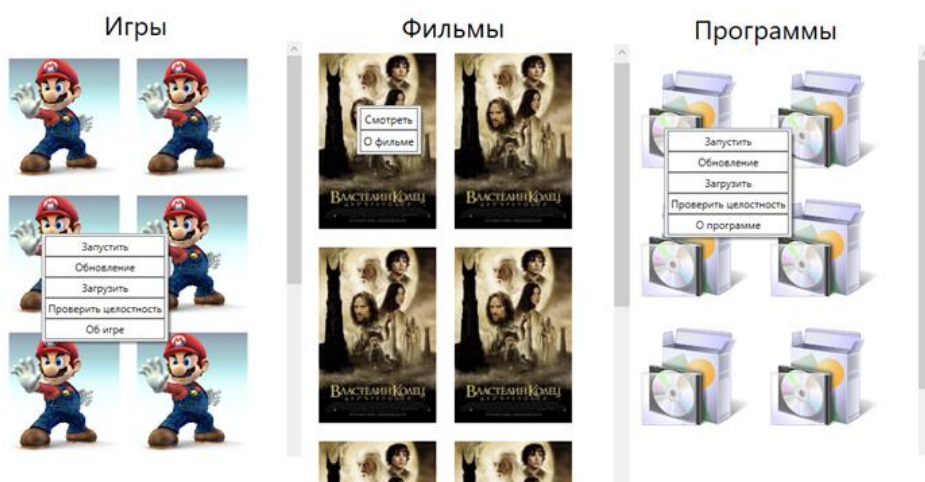


Рисунок 36 – Контекстное меню библиотеки

Помимо просмотра товаров, в системе предусмотрена возможность просмотра профилей пользователя. Владелец просматриваемого профиля видит полную информацию о себе, а также имеет возможность редактирования некоторых данных. Личный кабинет клиентов представлен на рисунках 37 и 38.



Рисунок 37 – Личный кабинет клиента. Контактные данные

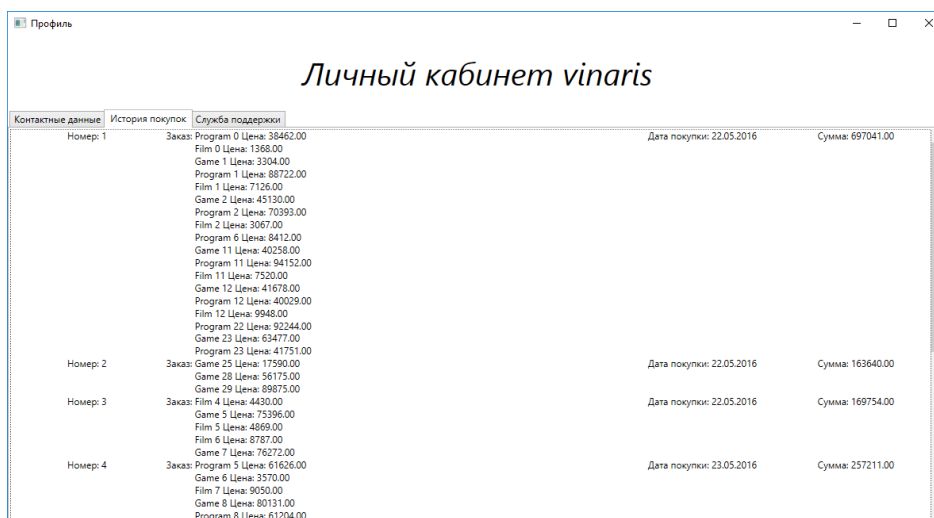


Рисунок 38 – Личный кабинет клиента. История заказов

Клиенты системы, не владеющие данной учётной записью, видят лишь общую для всех информацию и не имеют доступа к её истории покупок и обращений в службу поддержки. Вместо этого они могут добавить данную учётную запись в список контактов, а также написать ей сообщение. Исключение могут составить лишь учётные записи сотрудников технической поддержки и

администрации, которые имеют полный контроль над всей информацией, хранимой в системе. Только они могут изменять нередактируемые поля с целью разрешения затруднительных ситуаций. Каждый же клиент системы, в свою очередь, может редактировать только те поля, которые представлены на рисунке 39.

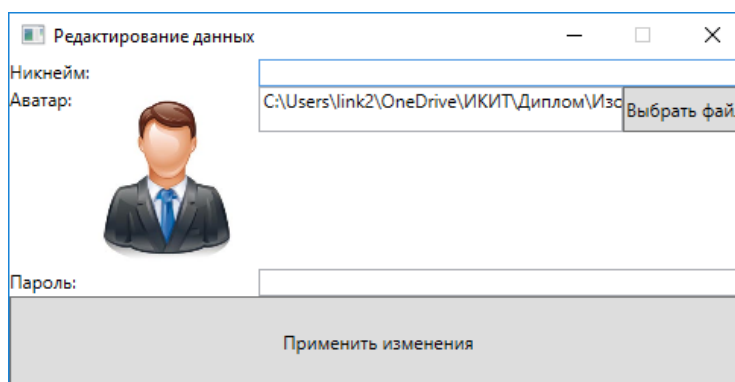


Рисунок 39 – Личный кабинет клиента. Редактор данных

Далеко не все пользователи имеют идентичный личный кабинет. Различные категории пользователей обладают различными привилегиями и, следовательно, личный кабинет должен адаптироваться под каждого из них. Далее представлены пара из оставшихся учетных записей: издатель – 40 рисунок, товаровед – 41 рисунок.

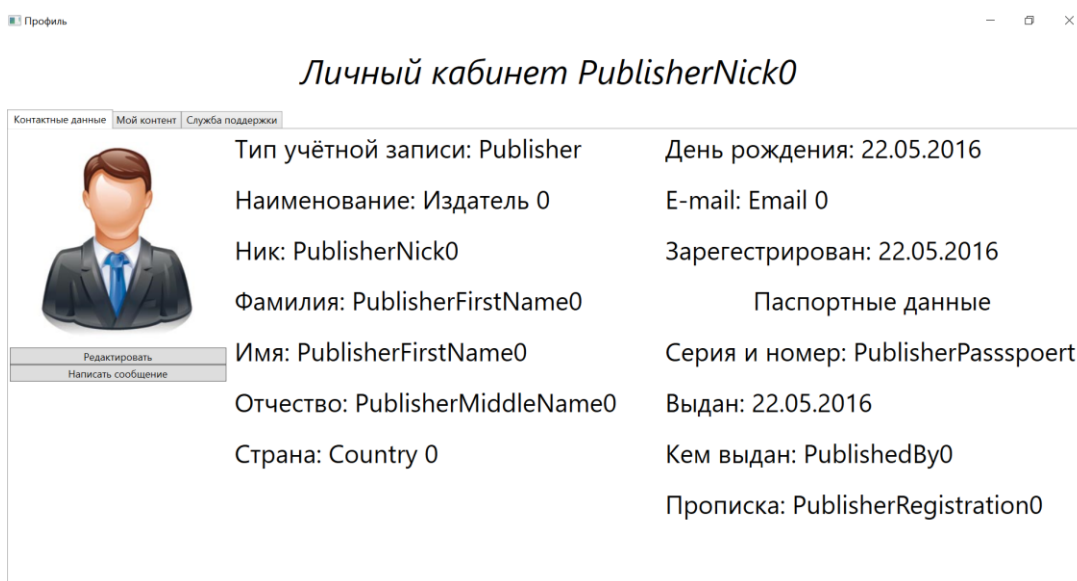


Рисунок 40 – Личный кабинет издателя. Контактная информация

Личный кабинет qwerty


Контактные данные	Обработка запросов	Служба поддержки
 Редактировать Написать сообщение	Тип учётной записи: Merchandiser	День рождения: 23.02.2016
	Ник: qwerty	E-mail: qwerty
	Фамилия: MerchandiserPassportFirst	Зарегистрирован: 22.05.2016
	Имя: MerchandiserPassportFirstNam	Паспортные данные
	Отчество: MerchandiserPassportMid	Серия и номер: MerchandiserPasspc
	Страна: Country 33	Выдан: 24.01.2016
	Кем выдан: MerchandiserPassportBy	
	Прописка: MerchandiserPassportReg	

Рисунок 41 – Личный кабинет товароведа. Контактная информация

Как можно увидеть выше, контактная информация у «административных» учетных записей отличается от клиентских наличием юридической информации. Это необходимо для однозначной идентификации личности, что позволит проводить платёжные транзакции. Помимо этого, у каждой категории пользователей имеется свой инструмент, позволяющий выполнять предназначённые им функции. Например, у издателя в личном кабинете имеется раздел «Мой контент», в котором он сможет просматривать общую статистику продаж, продажи каждого отдельного товара, редактировать данные о товаре (частично), публиковать новые товары (со статусом в обработке), отслеживать статус опубликованных товаров, получать выплаты с продаж. Пример интерфейса инструментария для издателя представлен на рисунке 42.

Профиль

Личный кабинет PublisherNick0

Контактные данные | Мой контент | Служба поддержки

Общая статистика продаж

Количество продаж: 5
Сумма: 108652.00

Добавление товара: Программа Фильм Игра




Game 0	Program 0	Film 0
Статус: На продаже	Статус: На продаже	Статус: На продаже
		
15180.00	38462.00	1368.00
2	2	1
30360.00	76924.00	1368.00

Рисунок 42 – Личный кабинет издателя. Инструментарий издателя

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была достигнута поставленная ей цель, а именно: оптимизация процессов купли/продажи и предоставления разнородной электронной продукции в сети интернет. Достигнута она была благодаря шагам, представленным далее.

1. Анализа предметной области, а также существующих систем, предоставляющих услуги по продаже цифрового контента (глава 1).
2. Создание проекта разрабатываемой системы (глава 2).
3. Непосредственная реализация системы в выбранной среде разработки (глава 3).

Разработанная система предоставляет огромную перечень возможностей.

- Единый магазин, включающий все категории электронного контента.
- Автоматизированный процесс покупки и предоставления контента.
- Аккумуляция всех покупок клиента в одном месте.
- Широкий спектр возможностей клиентов (загрузка и обновление контента, проверка целостности загруженного контента, личный кабинет, списки контактов, чаты, форумы).
- Отдельные личные профили и инструментариум у клиентов, издателей, товароведов, сотрудников тех. поддержки и администраторов.

На текущий момент система имеет достаточно сырой вид и нуждается в дальнейшей модификации. К основным направлениям дальнейшего развития можно отнести: улучшенный пользовательский интерфейс приложения; оптимизация потребления ресурсов; исправление ошибок; обеспечение безопасности; расширение функционала; мобильные клиенты приложения (iOS и Android).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Ларман, К. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е издание / К. Ларман ; пер. с англ. – Москва : ООО "И.Д. Вильямс", 2013. –736 с.
- 2) Кригель, А. SQL. Библия пользователя, 2-е издание / А. Кригель, Б. Трухнов ; пер. с англ. – Москва : ООО "И.Д. Вильямс", 2010. –752 с.
- 3) Стиллмен, Э. Изучаем С#. 3-е изд. / Э. Стиллмен, Д. Грин ; пер. с англ. – Санкт-Петербург : Питер, 2014. – 816 с.
- 4) Хейлсберг, А. Язык программирования С#. Классика Computers Science. 4-е изд. / А. Хейлсберг ; пер. с англ. – Санкт-Петербург : Питер, 2012. – 784 с.
- 5) Entity Framework Tutorial [Электронный ресурс] // EntityFrameworkTutorial.net. – Режим доступа: <http://www.entityframeworktutorial.net/>. – Заглавие с экрана. – (Дата обращения: 14.05.2016).
- 6) Сценарий использования [Электронный ресурс] // Справочная информационная система «Википедия». – Режим доступа: https://ru.wikipedia.org/wiki/Сценарий_использования/. – Заглавие с экрана. – (Дата обращения: 14.04.2016).
- 7) What is MVC? (Model-View-Controller) [Электронный ресурс] // Culttt. – Режим доступа: <http://culttt.com/2012/11/19/what-is-mvc-model-view-controller/>. – Заглавие с экрана. – (Дата обращения: 30.04.2016).
- 8) The MVVM Pattern [Электронный ресурс] // MSDN. – Режим доступа: <https://msdn.microsoft.com/en-us/library/hh848246.aspx/>. – Заглавие с экрана. – (Дата обращения: 30.04.2016).
- 9) Что такое проект. Определение проекта, признаки и примеры. [Электронный ресурс] // Stratmanager. – Режим доступа:

<http://stratmanager.ru/chto-takoe-proekt-opredelenie-proekta-priznaki-i-primery/>. –
Заглавие с экрана. – (Дата обращения: 20.03.2016).

10) Что такое проект? Определение и признаки. [Электронный ресурс] //
Международный научно-исследовательский журнал. – Режим доступа:
<http://research-journal.org/social/chto-takoe-proekt-opredelenie-i-priznaki/>. –
Заглавие с экрана. – (Дата обращения: 20.03.2016).

11) СТО 4.2–07–2014 Система менеджмента качества. Общие
требования к построению, изложению и оформлению документов учебной
деятельности. – Введ. 30.12.2013. – Красноярск: СФУ, 2014. – 60 с.

ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1 Общие сведения

Полное наименование системы: Информационная система «Сервис распространения цифрового лицензионного контента».

Условное обозначение: ИС «СРЦЛК».

Заказчик: Виденин Сергей Александрович, доцент, ИКИТ, СФУ.

Исполнитель: Касперович Валерий Константинович, студент, ИКИТ, СФУ.

Плановая дата начала выполнения работ: 08.02.2016

Плановая дата окончания выполнения работ: 29.06.2016

2 Назначение и цели создания системы

2.1 Назначение системы

ИС «СРЦЛК» предназначена для предоставления всем жителям Российской Федерации удобного и быстрого доступа ко всему цифровому контенту, распространяемому (бесплатно, условно-бесплатно или платно) через всемирную паутину, также известную как «Интернет». Помимо этого, система предоставляет набор инструментов для издателя, который в свою очередь сможет легко публиковать свой контент в системе.

2.2 Цель и задачи системы

Целью создания ИС «СРЦЛК» являются:

- создание дополнительного пункта купли/продажи и предоставления всех видов электронной продукции в единой среде.

Для достижения поставленной цели необходимо решить следующие задачи:

1. покупка/продажа цифрового контента;
2. аккумулярование разностороннего цифрового контента в одном месте;
3. предоставление полной информации о каждом товаре пользователю;
4. возможность загрузки/обновления купленного контента и последующее его использование;
5. социализация пользователей данной системы (сообщения, комментарии, форумы и др.);
6. полнофункциональный инструментарий для всех категорий пользователей системы (администратор – панель администратора, сотрудник тех. поддержки – панель технической поддержки и другие);
7. создание отечественного продукта, способного конкурировать на мировом рынке со своими аналогами, такими как Steam и др.

3 Требования к системе

Разрабатываемая ИС «СРЦЛК» должна полностью удовлетворять следующим бизнес-требованиям, пользовательским и функциональным требованиям, предъявленным данной системе.

Все требования будут разделены на роли. В данной системе необходимо организовать пять ролей пользователей: администратор, сотрудник технической поддержки, товаровед, издатель и клиент.

3.1 Бизнес-требования

Разработанная ИС «СРЦЛК» должна обеспечить издателей дополнительной торговой точкой, а пользователя – единым местом хранения всех купленных им товаров через интернет.

3.2 Пользовательские требования

Система должна быть удобна для каждого из своих пользователей: администратор, сотрудник технической поддержки, товаровед, издатель и клиент.

Такие модули, как личный кабинет и техническая поддержка доступны всем ролям системы, а вот об остальных будет изложено далее.

Для роли клиента необходимо реализовать и сделать доступным модули магазина, библиотеки и друзей.

Для роли издателя необходимо реализовать и сделать доступным модули панели издателя и учёта продаж, а также сделать частично доступным модуль магазина (только просмотр).

Для роли товароведа необходимо реализовать и сделать доступным модули панели товароведа, учёта продаж и движения денежных средств, а также сделать сверх доступным модуль магазина (возможность управления каталогом).

Для роли сотрудник тех. поддержки необходимо реализовать и сделать доступным модуль панели сотрудника тех. поддержки.

Для роли администратор необходимо реализовать и сделать доступным все модули данной системы.

3.3 Функциональные требования

В ИС «СРЦЛК» должны быть реализованы следующие функции:

- публикация и продажа цифрового контента;
- покупка и использование контента;
- общение между клиентами;

- техническая поддержка пользователей;
- учёт продаж;
- учёт движения денежных средств.

Для реализации выше перечисленных функций в системе должны присутствовать следующие модули:

- магазин;
- библиотека;
- друзья;
- личный кабинет;
- тех. поддержка;
- учёт продаж;
- учёт движения денежных средств;
- панель издателя;
- панель товароведа;
- панель сотрудника тех. поддержки;
- панель администратора.

4 Нефункциональные требования

Главными нефункциональными требованиями к разрабатываемой системе являются требованиями к интерфейсу. Собственно, как и любая другая система, ИС «СРЦЛК» должна обладать достаточно простым, интуитивным и лёгким в восприятии интерфейсом.

Интерфейсы для каждой из ролей должны соответствовать выполняемым ими функциями и содержать только те объекты системы, которые им необходимы.

Немного подробнее про интерфейс для каждой из ролей:

1. Клиент. Интерфейс клиента должен позволять ему просматривать всё содержимое в магазине, покупать и использовать его. При возникновении каких-либо (как в системе, так и в приобретённом товаре) проблем, интерфейс

должен позволить клиенту связаться со службой технической поддержки для их решения. Помимо этого, интерфейс предоставляет социальный модуль для внутренней коммуникации между пользователями.

2. Издатель. Интерфейс издателя должен предоставлять ему доступ к кабинету издателя, где он сможет публиковать свои товары в системе, управлять ими (например, контроль версии), видеть статистику продаж, а также отслеживать движение денежных средств. Помимо этого, издатель может просматривать каталог магазина и обращаться в службу поддержки для устранения возникших проблем.

3. Товаровед. Интерфейс товароведа должен предоставить соответствующий его обязанностям личный кабинет, в котором он сможет управлять как всем имеющимся в системе продаваемым контентом, так и вновь публикуемым.

4. Сотрудник технической поддержки. Интерфейс сотрудника тех. поддержки должен позволять ему получать запросы остальных пользователей системы и иметь возможности их решения.

5. Администратор. Интерфейс администратора должен предоставить ему полный доступ ко всем ресурсам системы, а также вести полную статистику работы всей системы. Кроме того, на него ложится бухгалтерский учёт.

5 Системные ограничения

Разрабатываемая информационная система не должна требовать от заказчика большой вычислительной мощности аппаратного обеспечения. За исключением продаваемого контента, который может быть более требовательным.

Системные требования:

- операционная система: Windows XP/Vista/7/8/10;
- процессор: Intel Pentium IV и выше;

- оперативная память: 1 Гб;
- жесткий диск: ~ 100 Мб;
- SVGA дисплей.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий

институт

Информационные системы

кафедра

УТВЕРЖДАЮ

Зав. кафедрой ИС

_____ С. А. Виденин

подпись инициалы, фамилия

« ____ » _____ 2016 г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы

Студенту Касперович Валерий Константиновичу

Группа КИ12-13Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: ИС «Сервис распространения цифрового лицензионного контента».

Утверждена приказом по университету №4729/с от 05.04.2016.

Руководитель ВКР: С. А. Виденин, заведующий, доцент кафедры «Информационные системы» ИКИТ СФУ.

Исходные данные для ВКР: список требований к разрабатываемой системе, методические указания научного руководителя, учебные пособия, электронные ресурсы.

Перечень разделов ВКР: введение, общие сведения, аналитическая часть, проектная часть и часть реализации проекта.

Перечень графического материала: презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР

подпись

С. А. Виденин

инициалы и фамилия

Задание принял к исполнению

подпись

В. К. Касперович

инициалы и фамилия студента

« ____ » _____ 20__ г.