Комитет по образованию Правительства Санкт-Петербурга

**САНКТ- ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**О Т Ч Ё Т**

**о прохождении преддипломной практики**

**Тема диплома** «Разработка информационной системы

для музыкального магазина»

#### **Специальность 09.02.07 «Информационные системы и программирование»**

**Квалификация** Программист

#### **Студентки** Сидорова Антон Дмитриевич

#### **Группа** 493

##### Руководитель практики Матысик И.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_подпись

Преподаватель спецдисциплин колледжа информационных технологий

Место прохождения практики СПБ ГБПОУ КИТ

Сроки прохождения практики с 20.04.23 г. по 17.05.23 г.

Итоговая оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc135071570)

[1 Цель разработки 5](#_Toc135071571)

[2 Исследование предметной области 5](#_Toc135071572)

[3 Обзор и анализ аналогичных решений 8](#_Toc135071573)

[4 Анализ средств разработки приложения 10](#_Toc135071574)

[4.1 Обзор методов решения 10](#_Toc135071575)

[4.2 Обоснование выбора средств разработка 28](#_Toc135071576)

[ЗАКЛЮЧЕНИЕ 34](#_Toc135071577)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 35](#_Toc135071578)

[ПРИЛОЖЕНИЕ A. Структура репозитория на Git Hub 36](#_Toc135071579)

[ПРИЛОЖЕНИЕ Б. Исходный текст программы 37](#_Toc135071580)

[ПРИЛОЖЕНИЕ В. Работа программ 42](#_Toc135071581)

# ВВЕДЕНИЕ

Цель данной дипломной работы – разработать информационную систему для музыкального магазина.

На предприятии продаются музыкальные инструменты, музыкальное оборудование, запчасти для музыкальных инструментов и оборудывания, и прочие товары. Само предприятие представляет собой торговую сеть их головного офиса, нескольких складов в разных городах, и нескольких магазинов и пунктов выдачи, за которыми закреплён склад (с которого поставляются товары), и которые подчинены головному офису.

На предприятии работают следующие работники:

* менеджер по складу – проводит учёт товаров на складе и в магазине, обновляет информацию о товарах и их количестве;
* менеджер по заказам – формирует заказы на пункте выдачи, обновляет информацию о них;
* продавец – формирует заказы в магазине, и продаёт заказанные товары в них;
* оператор – формирует заказы, о которых им клиент сообщает по средствам связи;
* директор – возглавляет предприятие. Он нанимает работников, увольняет их, отправляет их в отпуск, и т. д;
* системный администратор – работает с техническим оборудыванием, имеет права на все ресурсы компьютеров на предприятия.

Помимо того, предприятие работает с клиентами, которые могут заказывать товары, и получать заказы. В магазине заказы формирует и выдаёт продавец. На пункте выдаче заказы формирует и выдаёт менеджер по заказам. Также, как в магазине, так и на пункте выдачи, заказывать товары, используя средства связи, где за формирование заказа отвечает оператор.

При работе над проектом были применены навыки, приобретённые во время изучения профессиональных модулей.

ПМ 01. «Разработка программных модулей программного обеспечения для компьютерных сетей»:

* ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;
* ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля;
* ПК 1.4. Выполнять тестирование программных модулей;
* ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

ПМ 02 «Осуществление интеграции программных модулей»

* ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент;
* ПК 2.2 Выполнять интеграцию модулей в программное обеспечение.

ПМ 04 «Сопровождение и обслуживание программного обеспечения компьютерных систем»

* ПК 4.1 Осуществлять инсталляцию, настройку и обслуживание программного обеспечения компьютерных систем.

ПМ 11 «Разработка, администрирование и защита баз данных»

* ПК 11.2 Проектировать базу данных на основе анализа предметной области;
* ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области;
* ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных.

# Цель разработки

Целью дипломного проекта является создание и разработка информационной системы для музыкального магазина, которая позволит автоматизировать бизнес-процессы, протекающие в компании. А именно: вести данные о клиенте-покупателе, консультироваться с ними, подбирать подходящие под условия клиента товары, вносить изменения о товарах, заказывать товары, продавать товары.

Автоматизированная система должна выполнять расчеты отчисляемых сумм продавцу/компании с учётом действующих скидок. Также, система должна автоматизировать заказы и продажи музыкальных инструментов, музыкального оборудывания, аксессуаров и иных товаров, связанных с музыкой, а также внесение изменений сведений о товарах. Причём, система должна автоматизировать данные процессы в на складах, а также в магазинах и пунктах выдачи нескольких организаций.

Разрабатываемый проект предполагает информационную систему, состоящую из базы данных, взаимодействующими с ней десктопным приложением и Web API, а также мобильное приложение, взаимодействующее с Web API. При этом, это предполагает, что будут работать десктопные и мобильные приложения, на многих устройствах, взаимодействуя при этом с одной базой данных.

Автоматизируется бизнес-процесс заказов товаров в магазинах, и в пунктах выдачи со складов. При формировании заказов, автоматически рассчитывается полная стоимость заказов. Таким образом в будущем не придется самому рассчитывать стоимость заказов, а также, можно будет просматривать наличие.

# Исследование предметной области

Информационная система «Музыкальный магазин» занимается продажей и заказами музыкальных инструментов, музыкального оборудывания, запчастей для музыкальных инструментов и оборудывания, и прочих товаров. Система должна предусмотреть возможность просмотра товаров, а также их количества на складах и в магазинах, а также возможность заказа товаров на пункт выдачи, и заказа товара в магазине. Система должна иметь базу данных, а также десктопное и мобильное приложение. Десктопное приложение взаимодействует с базой данных напрямую, а мобильное приложение через Web API. При этом, десктопные и мобильные приложения на всех устройствах работают с одной базой данных. Web API работает с базой данных, также, напрямую. Web API должно иметь запросы для функционала клиента, поскольку оно создано для взаимодействия мобильного приложения с базой данных.

Гость (неавторизированный пользователь) просматривать товары и их список. Для отбора товаров с указанием их количества, используются магазины (для заказов в магазинах) и склады (для заказов в пункт выдачи). Магазины и пункты выдачи, это одним словом – торговые пункты. Каждый торговый пункт имеет название, адрес расположения, контактные данные. Он принадлежит определённой организации, и закреплён за определённым складом. Каждая организация имеет название и головной офис, а значит, также адрес расположения, контактые данные, и свой сайт. Склад имеет название и контактые данные. Также, склад находится в определённом городе и имеет адрес расположения в нём.

Под контактными данными выше понимается телефон и Email-адрес.

Каждый авторизированный пользователь имеет полный функционал гостя, а также может редактировать свои данные: пароль, номера своих телефонов, Email-адреса. Также, авторизированный пользователь может просматривать историю входов.

Весь выше перечисленный функционал имеется в десктопном приложении, Web API и мобильном приложении.

Каждый пользователь имеет одну или несколько ролей в системе. Дополнительные возможности пользователя даются набором его ролей в системе. Но, полный функционал, только в десктопном приложении, а в Web API и мобильном приложении, полный функционал, только у клиента, а у остальных, только вышеперечисленный.

Роли в системе:

* клиент может заказывать товары, и просматривать историю своих заказов. Может заказывать товары в пункт выдачи, а также бронировать товары в магазине. Единственная роль, имеющая полный функционал, как в десктопном приложении, так и в Web API и мобильном приложении;
* директор может работать с пользователями в системе;
* менеджер по складу может добавлять, редактировать и удалять товары, а также их позиции на складах и в магазинах;
* менеджер по заказам может формировать заказы, редактировать и удалять их, а также назначать пункт выдачи для получения;
* продавец может формировать заказы в магазине, в котором он работает, а также продавать товары;
* оператор может формировать заказы и назначать пункт выдачи для их получения;
* администратор имеет полный функционал выше перечисленных пользователей, за исключением клиента. Также может просматривать историю входов и попыток входов в систему.

База данных системы должна иметь данные о товаре, пользователе, магазинах, и позициях товаров в них, складах и позициях товаров в них, заказах, пунктах выдачи и позициях заказов в них, пользователях, историю входа и т.п.

Данная информационная система разрабатывается для нескольких организаций (представленных головным офисом, и подчинёнными ему магазинами и пунктами выдачи), занимающихся продажей и заказами музыкальных инструментов, музыкального оборудывания, запчастей для музыкальных инструментов и оборудывания, и прочих товаров.

# Обзор и анализ аналогичных решений

В данном разделе представлен анализ существующих систем «музыкальный магазин», с целью выявления плюсов и минусов у разработанных решений. Проведя данный анализ можно разработать проект более качественным и быть конкурентно способным в данной предметной области.

**Музыкальный интернет-магазин Мир Музыки**

Сайт данного магазина находится в адресу <https://mirm.ru/shops/spb/>, как показано на рисунке 1.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 1 – Сайт «Музыкальный интернет-магазин Мир Музыки»

Данный сайт – сайт сети музыкальных салонов Мир Музыки представлена в 13-ти крупных городах России, где расположены розничные магазины. Данная сеть динамично развивается и является одним из лидеров рынка розничной торговли музыкальных инструментов и оборудования. Компания «Мир Музыки» – это сеть крупнейших в России магазинов музыкальных инструментов, световой техники, звукового, студийного и торгового оборудования для музыкальных магазинов.

Функционал сайта позволяет просматривать списки товаров, фильтруя по категориям, просматривать информацию о товаре, включая описание и характеристики. Можно просматривать магазины в разных городах, а также заказывать с доставкой в городах, где находятся эти магазины, и в другие города.

Сайт обеспечивает автоматизацию формирования заказов – от выбора товаров, до оформления заказов. Также, при выборе города и адреса магазина, можно просмотреть информацию о персонале.

Данный сайт можно запустить из браузера, что делает его независимым от параметров устройства: оборудывания и операционной системы.

**Музыкальный интернет-магазин Музторг**

Сайт данного магазина находится в адресу <https://www.muztorg.ru/>, как показано на рисунке 2.

Изображение выглядит как текст, снимок экрана, Веб-сайт, Реклама в Интернете

Автоматически созданное описание

Рисунок 2 – Сайт «Музыкальный интернет-магазин МузТорг»

Музторг — лидер рынка розничной продажи музыкальных инструментов и оборудования в России. Его магазины предлагают лучшие музыкальные инструменты, созданные мировыми производителями, а также весь спектр профессионального звукового, светового и студийного оборудования. В ассортименте более 40000 товаров. Данная сеть существует в большем количестве городов, чем предыдущая, но также, позволяет оформлять доставку в другие города. Даже если в Вашем городе нет магазина «Музторг», оперативно доставляется выбранный вами товар транспортной компанией в любой регион России (включая Крым), Беларуси или Казахстана.

Сайт обеспечивает возможность просмотра списка товаров с фильтром по категориям, и просматривать описание определённого товара. Можно выбрать город, в котором можно просматривать ассортимент.

Сайт обеспечивает автоматизацию формирования заказов – от выбора товаров, до оформления заказов. Также, при выборе города и адреса магазина, можно просмотреть информацию о персонале.

Данный сайт можно запустить из браузера, что делает его независимым от параметров устройства: оборудывания и операционной системы.

# Анализ средств разработки приложения

В данном разделе представлен анализ ПО, которое можно использовать для разработки данного проекта, а также, ПО, которое было выбрано для реализации программного решения дипломного проекта.

# Обзор методов решения

В данном разделе описываются средства для разработки программного обеспечения:

* среды программирования;
* средства для развёртывания баз данных;
* среды для работы с базами данных;
* средства для реализации инсталлятора;
* операционные системы;
* языки программирования.

**4.1.1 Среды программирования**

Для начала проведем подбор и анализ средств для создания и разработки приложений. Проанализируем программы такие как:

* Microsoft Visual Studio;
* JetBrains Rider;
* Android Studio.

**Microsoft Visual Studio**

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight. После покупки компании Xamarin корпорацией Microsoft появилась возможность разработки IOS и Android программ.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Плюсы:

* официальная. Так как и язык, и среда разработки созданы в Microsoft, логично предположить, что ничего более функционального вы не найдете во всем Интернете. В некоторых случаях без Visual Studio не обойтись — например, при использовании технологий UWP и WPF;
* бесплатная. Версии «Community edition» для рядового пользователя будет достаточно. Тем более, теперь можно подключать плагины (в отличие от старой версии Express);
* функциональная. В Visual Studio множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки;
* поддерживает платформы .NET. Visual Studio имеет широкие возможности по разработке приложений под Windows, в том числе в .NET-сегменте;
* облачные хранилища. Зарегистрируйтесь в сообществе Visual Studio — и получите доступ к облачному хранилищу, где сможете располагать файлы проектов;
* корпоративность. Технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки.

Минусы:

* баги при переходах с триал-версии. При переходе на платную версию могут теряться настройки и нарушаться работа корпоративного сервера;
* сложность. Самостоятельно освоить Visual Studio новичку будет непросто — слишком много доступных функций, спрятанных в подразделах меню.

**JetBrains Rider**

Rider — кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#.

Проект анонсирован в январе 2015 года. В его основе лежит другой продукт JetBrains — ReSharper. Среда поддерживает платформы .NET Framework, .NET и Mono. Работает на операционных системах Windows, macOS, Linux.

Плюсы:

* наличие плагина ReSharper, который изначально разработан для повышения производительности Visual Studio. Теперь на его основе выпущена IDE;
* поддержка полного цикла. Фирменная черта продуктов JetBrains, воплощенная и в Project Rider. С ним вы сможете организовать весь цикл создания ПО: от идеи до поддержки;
* функциональность. Project Rider позволяет подключить MSBuild и XBuild, работать с CLI-проектами и организовать отладку приложений .NET and Mono. Множество опций для быстрого создания кода улучшает производительность;
* Multiple runtime. Поддержка нескольких запущенных программ;
* кроссплатформенность. Project Rider работает с Windows, Linux и MacOS;
* контроль версий. Встроенный инструмент позволяет напрямую организовать работу с Git, Mercurial и TFS.

Минусы:

* молодость. Часть функциональности еще в разработке, не все стартовые баги исправлены;
* стоимость. Самая дешевая версия Project Rider обойдется в 139 долларов за первый год использования. Но есть триал-версия и специальные предложения для студентов и непрофильных организаций.

**Android Studio**

Android Studio — интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O. В последней версии Android Studio поддерживается Android 4.1 и выше.

Данная IDE находилась в свободном доступе начиная с версии 0.1, опубликованной в мае 2013, а затем перешла в стадию бета-тестирования, начиная с версии 0.8, которая была выпущена в июне 2014 года. Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android Development Tools (ADT) для Eclipse.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, — официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, macOS и GNU/Linux. 17 мая 2017, на ежегодной конференции Google I/O, Google анонсировал поддержку языка Kotlin, используемого в Android Studio, как официального языка программирования для платформы Android в дополнение к Java и C++.

Плюсы:

* среда разработки поддерживает работу с несколькими языками программирования, к которым относятся самые популярные – C/C++, Java;
* редактор кода, с которым удобно работать;
* позволяет разрабатывать приложения не только для смартфонов/планшетов, а и для портативных ПК, приставок для телевизоров Android TV, устройств Android Wear, новомодных мобильных устройств с необычным соотношением сторон экрана;
* тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе;
* рефакторинг уже готового кода;
* достаточно большая библиотека с готовыми шаблонами и компонентами для разработки ПО;
* разработка приложения для Android N – самой последней версии операционной системы;
* предварительная проверка уже созданного приложения на предмет ошибок в нем;
* большой набор средств инструментов для тестирования каждого элемента приложения, игры;
* для неопытных/начинающих разработчиков специально создано руководство по использованию Android Studio, размещенное на официальном сайте утилиты.

Минусы:

* несмотря на наличие встроенного Android-эмулятора в самой среде разработки, с тестированием новоразработанного приложения могут возникнуть трудности. Так, для его запуска необходима достаточно внушительная по производительности аппаратная основа ПК, на котором планируется тестирование;
* невозможность написать серверные проекты на языке Java для ПК, Android устройств.

**4.1.2 Средства для развёртывания баз данных**

Далее проведем анализ средств для работы с базой данных. Для этого берем следующие программы:

* Microsoft SQL Server;
* PostgreSQL.

**Microsoft SQL Server**

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Плюсы:

* масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов;
* размер страниц – до 8 Кб. Данные извлекаются быстро, а сложную информацию удобнее хранить. Система обрабатывает транзакции в интерактивном режиме, есть динамическая блокировка;
* автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей;
* удобный поиск. Его можно осуществлять по фразам, словам, тексту либо создавать ключевые индексы;
* поддержка работы с другими решениями Майкрософт, в том числе с Excel, Access.

Минусы:

* зависимость от ОС. Система работает только с Windows;
* высокая стоимость.

**PostgreSQL**

PostgreSQL (произносится «Пост-Грес-Кью-Эл») – свободная объектно-реляционная система управления базами данных (СУБД).

Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Плюсы:

* легко использовать;
* имеет пользовательский тип данных;
* открытый источник;
* большая поддержка сообщества;
* используйте хранимые процедуры;
* поддерживается ACID, т.е. атомарность, согласованность, изоляция, долговечность.

Минусы:

* архитектура – создаётся отдельный сервис для каждого клиента, который превращается в большое использование памяти;
* при сравнении с другими СУБД, PostgreSQL уступает в плане производительности;
* он не так популярен, как другие системы управления базами данных;
* нехватка квалифицированных специалистов;
* при сравнении с другими СУБД, PostgreSQL уступает по скорости;
* делать репликацию сложнее;
* установка не легка для новичка.

**4.1.3 Среды для работы с базами данных**

Далее проведем анализ средств для работы с базой данных. Для этого берем следующие программы:

* Microsoft SQL Server Management Studio;
* PgAdmin;
* Microsoft Access.

**Microsoft SQL Server Management Studio**

SQL Server Management Studio (SSMS) – утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

Также есть SQL Server Management Studio Express для Express версии сервера, которая является бесплатной. Однако в ней нет поддержки ряда компонентов (Analysis Services, Integration Services, Notification Services, Reporting Services) и SQL Server 2005 Mobile Edition.

Начиная с версии 16.5.3 пакет SSMS выделен в отдельный обновляемый продукт, доступный для скачивания на сайте Microsoft. Текущая доступная версия SSMS 18.4 (15.0.18206.0) (general availability) поддерживает MS SQL server начиная с версии 2008 по 2019.

Плюсы:

* масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов;
* размер страниц – до 8 Кб. Данные извлекаются быстро, а сложную информацию удобнее хранить. Система обрабатывает транзакции в интерактивном режиме, есть динамическая блокировка;
* автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей;
* удобный поиск. Его можно осуществлять по фразам, словам, тексту либо создавать ключевые индексы;
* поддержка работы с другими решениями Майкрософт, в том числе с Excel, Access.

Минусы:

* зависимость от ОС. Система работает только с Windows;
* высокая стоимость.

**PgAdmin**

pgAdmin – это ведущий инструмент управления с открытым исходным кодом для PostgreSQL, самой передовой в мире базы данных с открытым исходным кодом.

pgAdmin 4 предназначен для удовлетворения потребностей как начинающих, так и опытных пользователей PostgreSQL, предоставляя мощный графический интерфейс, который упрощает создание, обслуживание и использование объектов базы данных.

Плюсы:

* совместим с Linux, Windows, macOS;
* позволяет работать с несколькими серверами одновременно;
* экспорт в CSV;
* планирование запросов;
* возможность отслеживать ваши сеансы, блокировки БД с помощью панели мониторинга;
* ярлыки в редакторе SQL для более удобной работы;
* встроенный отладчик процедурного языка;
* тщательная документация и активное сообщество.

Минусы:

* медленный и не всегда интуитивно понятный пользовательский интерфейс по сравнению с платными конкурентами;
* тяжелый;
* высокий порог вхождения;
* для работы с несколькими базами данных одновременно потребуются продвинутые навыки.

**Microsoft Access**

Microsoft Office Access или просто Microsoft Access — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA в самом Access можно писать приложения, работающие с базами данных.

Плюсы:

* очень простой графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать приложения, используя встроенные средства;
* хранит все данные в одном файле, хотя и распределяет их по разным таблицам, как и положено реляционной СУБД. К этим данным относится не только информация в таблицах, но и другие объекты базы данных;
* предлагает большое количество Мастеров, которые выполняют основную работу за пользователя при работе с данными и разработке приложений, помогают избежать рутинных действий и облегчают работу неискушенному в программировании пользователю;
* распространенность, которая обусловлена тем, что Access является продуктом компании Microsoft;
* постоянно обновляется производителем, поддерживает множество языков;
* полностью совместим с операционной системой Windows;
* ориентированность на пользователя с разной профессиональной подготовкой, что выражается в наличии большого количества Мастеров, развитую систему справки и понятный интерфейс;
* широкие возможности по импорту/экспорту данных в различные форматы, от таблиц Excel и текстовых файлов, до практически любой серверной СУБД через механизм ODBC;
* наличие развитых встроенных средств разработки приложений. Большинство приложений, распространяемых среди пользователей, содержит тот или иной объем кода VBA (Visual Basic for Applications);
* наличие встроенного языка макрокоманд.

Минусы:

* ограничены возможности по обеспечению многопользовательской среды;
* в ранних версиях (до Access 2003) отсутствуют такие средства как триггеры и хранимые процедуры, что заставляет разработчиков возлагать поддержание бизнес логики БД на клиентскую программу или разрабатывать процедуры с помощью встроенного средства VBA;
* обладает несложными способами защиты с использованием пароля БД (возможно применения дополнительных мер по защите от несанкционированного доступа с использованием процедур VBA);
* в вопросах поддержки целостности данных отвечает только моделям БД небольшой и средней сложности;
* не распространяется бесплатно.

**4.1.4 Средства для реализации инсталлятора**

Теперь рассмотрим средства для реализации инсталлятора программного продукта:

* Inno Setup;
* Microsoft Visual Studio Installer Projects.

**Inno Setup**

Inno Setup — это бесплатное программное обеспечение, управляемая сценариями система установки, созданная в Delphi Джорданом Расселом. Первая версия была выпущена в 1997 году.

**Microsoft Visual Studio Installer Projects**

Microsoft Visual Studio Installer Projects – это плагин для создание проектов инсталлятора для созданного проекта.

**4.1.5 Операционные системы**

Далее проведем анализ операционных систем:

* Windows;
* Android;
* Linux.

**Windows**

Windows – группа семейств коммерческих проприетарных операционных систем корпорации Microsoft, ориентированных на управление с помощью графического интерфейса, пришедших на смену MS-DOS. В отличие от текстовой однозадачной MS-DOS, операционные системы Windows являются графическими и многозадачными. Каждое семейство обслуживает определённый сектор компьютерной индустрии. Активные семейства Microsoft Windows включают Windows NT и Windows IoT; они могут включать подсемейства (например, Windows Server или Windows Embedded Compact) (Windows CE). Устаревшие семейства Microsoft Windows включают Windows 9x, Windows Mobile и Windows Phone. Изначально Windows была всего лишь графической программой-надстройкой для распространённой в 1980-х и 1990-х годах операционной системы MS-DOS. Согласно данным ресурса Net Applications, по состоянию на август 2014 года под управлением операционных систем семейства Windows работает около 88 % персональных компьютеров. Windows работает на PC-совместимых архитектурах с процессорами x86, x86-64, а также на архитектуре ARM. Существовали также версии для DEC Alpha, MIPS, IA-64, PowerPC и SPARC. Последней на данный момент операционной системой Microsoft является Windows 11, представленная 24 июня 2021 года.

Плюсы:

* распространенность и популярность. Нельзя отрицать тот факт, что система является самой распространенной на рынке. Это обусловлено простотой освоения и доступностью ПО. Следовательно, вам не составит труда купить Windows для своего ПК, а также найти программное обеспечение под нее;
* удобный графический интерфейс. Меню системы адаптировано под пользователей любого уровня. Даже не имея навыков работы с ОС, вы быстро освоитесь в ее базовых функциях и сможете с успехом пользоваться;
* развлечение. Если вы любите играть в компьютерные игры, то без сомнений оцените возможности и преимущества Windows, так как эта ОС в наибольшей степени поддерживается разработчиками. Все последние игры обязательно будут доступны на ней;
* доступность современных технологий. Система активно развивается и дополняется всевозможными функциями в соответствии с современными технологиями. Свежие версии ПО, антивирусы, редакторы – все это поддерживается здесь.

Минусы:

* недостаточный уровень защищенности для узкоспециализированной работы (стандартное антивирусное ПО не способно обеспечить необходимую безопасность);
* сбои в работе (даже самые актуальные версии ОС могут работать с ошибками, но они устраняются посредством обновлений);
* решения в вопросах обновления функционала и меню не всегда удачные.

**Linux**

Linux – семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для сопровождения и обновлений, — и имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных.

Появившись как решения вокруг созданного в начале 1990-х годов ядра, уже с начала 2000-х годов системы Linux являются основными для суперкомпьютеров и серверов, расширяется применение их для встраиваемых систем и мобильных устройств, некоторое распространение системы получили и для персональных компьютеров.

Плюсы:

* бесплатность;
* настраиваемость;
* безопасность;
* нетребовательность к ресурсам;
* драйверы оборудования;
* удобная командная строка;
* большой выбор графических окружений;
* продуманная файловая система;
* удобная система хранения настроек;
* поддержка большого количества архитектур;
* отсутствие сбора данных;
* неплохая поддержка сообществом;
* большое количество бесплатных программ.

Минусы:

* сложность освоения;
* отсутствие версий популярных программ;
* отсутствие поддержки некоторого оборудования;
* недостаточная поддержка игр.

**Android**

Android – операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, нетбуков, смартбуков, очков Google Glass, телевизоров, проекторов и других устройств (в 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов).

Изначально разрабатывалась компанией Android, Inc., которую затем приобрела Google. Основана на ядре Linux и собственной реализации виртуальной машины Java компании Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (OHA), который занимается поддержкой и дальнейшим развитием платформы.

Плюсы:

* доступность системы для аппаратных платформ MIPS, x86, AR;.
* быстрая работа;
* большое количество дополнений и приложений, способствующих расширению функционала;
* способность системы подстраиваться под пользователей;
* оптимизация, отсутствие жёстких требований к самим устройствам;
* открытый исходный код, благодаря которому разработчики могут создавать любые решения.

Минусы:

* кража данных у пользователей без их ведома;
* лёгкий поиск уязвимостей, разработка вирусов;
* сильный расход энергии, если оптимизация пострадала;
* производители создают оболочки, которые не всегда могут похвастаться высокой производительностью

**4.1.6 Языки программирования**

И напоследок рассмотрим языки программирования для реализации программного продукта:

* С#;
* Java;
* Swift.

**C#**

C# – объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java – С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

Плюсы:

* независимость от аппаратного функционала;
* поддержка;
* управление памятью;
* строгая типизация;
* крупное сообщество;
* синтаксический сахар.

Минусы:

* невысокая скорость;
* слабое взаимодействие с железом.

**Java**

Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Разработка ведётся сообществом, организованным через Java Community Process; язык и основные реализующие его технологии распространяются по лицензии GPL. Права на торговую марку принадлежат корпорации Oracle.

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины. Дата официального выпуска – 23 мая 1995 года. Занимает высокие места в рейтингах популярности языков программирования (2-е место в рейтингах IEEE Spectrum (2020) и TIOBE (2021)).

Плюсы:

* концепция ООП (объектно-ориентированного программирования);
* язык высокого уровня, т.е., он больше похож на человеческую речь, а не на машинный код;
* безопасность;
* удобство для распределённого программирования;
* принцип «написать один раз и использовать везде»;
* стабильное и постоянно развивающееся сообщество.

Минусы:

* низкая скорость;
* многословие (verbosity);
* платность для коммерческого использования (с 2019 года).

**Swift**

Swift — открытый мультипарадигмальный компилируемый язык программирования общего назначения, разработанный и поддерживаемый компанией Apple. Первая версия была представлена в 2014 году.

Чаще всего Swift используется в разработке приложений для macOS, iOS, iPadOS, watchOS и tvOS, однако язык также доступен для Windows и Linux. На платформах Apple язык работает с фреймворками Cocoa и Cocoa Touch и совместим с основной кодовой базой Apple, написанной на более раннем языке Objective-C. Swift задумывался как более лёгкий для чтения и устойчивый к ошибкам программиста язык: существенная часть проблем, которые в случае с Objective-C обнаруживаются только при исполнении программ и приводят к крэшам, в Swift в принципе не позволят скомпилировать код.

Компилятор Swift работает на базе LLVM, в результате чего один и тот же код может быть скомпилирован для различных платформ: x86, ARM, WASM и других. Набор инструментов (toolchain) для работы с языком встроен в интегрированную среду разработки Xcode 6 и выше. Swift может использовать рантайм Objective-C, что делает возможным использование обоих языков (а также языка Си) в рамках одной программы.

Плюсы:

* легкий;
* безопасный;
* имеет автоматическое управление памятью;
* имеет открытый исходный код;
* имеет большой потенциал;
* функционально совместимый.

Минусы:

* ограниченный кадровый потенциал;
* малое количество библиотек;
* нестабильность.

# Обоснование выбора средств разработка

В данном подразделе проводится обоснование стека выбранных технологий и средство разработки, которые наилучшим образом подходят для разрабатываемого проекта.

**4.2.1 Среды программирования**

Для разработки десктопного приложения и Web API была выбрана Visual Studio, поскольку, эта среда разработки:

* имеет все необходимые средства для разработки проектов, обоих выше перечисленных типов;
* проекты обоих типов, можно, разработать, используя, для этого один языка программирования – C#;
* разработанное Web API на платформе NET 5 (Для Web API выбрана эта платформа), можно запустить в браузере, используя в нём Swagger, а также, предоставить к нему удалённый доступ, в том числе для разработки клиентского приложения;
* разработанное Web API запускается под адресов localhost, что позволяет с ним работать без подключения к сети;
* разработанное десктопное приложение на платформе NET Framework 4.7.2 (для десктопного приложения выбрана эта платформа) можно запустить на любом устройстве с операционной системой Windows, используя для этого созданный при компиляции exe-файл;
* имеется встроенные отладчик;
* имеет возможность создавать окна десктопного приложения с помощью конструктора.

Для разработки мобильного приложения, была выбрана Android Studio, поскольку:

* имеет встроенный эмулятор;
* имеет возможность создать APK-файл, из которого можно установить приложение на физическое устройство с операционной системой Android;
* имеет возможность создавать окна с помощью конструктора;
* для каждого приложения выбирается Min SDK (минимальная версия Android и уровень API), а само приложения можно установить на устройство, которое соответствует данным параметрам, и на любое устройство, с более поздней версией, и более высоким уровнем API.

Рейтинг сред разработки представлен на рисунке 3.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 3 – Рейтинг сред разработки

**4.2.2 Средства для развёртывания баз данных**

Для развёртывания базы данных был выбран Microsoft SQL Server, потому что:

* удобно работать;
* быстрый и мощный;
* имеется возможность работать с базами данных, используя код на языке C# в Visual Studio, вне зависимости от типа проекта;
* можно создавать несколько диаграмм одной базы данных.

Рейтинг сред развёртывания баз данных представлен на рисунке 4.

Изображение выглядит как линия, снимок экрана, График, текст

Автоматически созданное описание

Рисунок 4 – Рейтинг сред развёртывания баз данных

**4.2.3 Средства работы с базой данных**

Для работы с базой данных была выбрана среда MS SQL Server Management Studio, потому что:

* имеет встроенный язык Transact SQL;
* можно создавать скрипты с запросами на языке Transact SQL;
* простота создания баз данных и её объектов.

Рейтинг сред работы с базой данных представлен на рисунке 5.

Изображение выглядит как текст, снимок экрана, диаграмма, круг

Автоматически созданное описание

Рисунок 5 – Рейтинг сред работы с базами данных

**4.2.4 Средства для реализации инсталлятора**

Для реализации инсталлятора своего проекта было выбрано приложение Into Setup, поскольку оно:

* простое;
* бесплатное;
* интуитивно понятное.

**4.2.5 Операционная система**

Выбранной операционной системой, является Windows, поскольку она:

* простая в использовании;
* имеет графический интерфейс пользователя;
* имеет выход в Интернет;
* простая в установке программ.

При этом для тестирования мобильного приложения была выбрана операционная система Android:

* имеет возможность установки из API-файлов, следовательно есть возможность установить приложение, написанное в Android Studio;
* имеет выход в Интернет;
* простая в использовании;
* проста в установке программ.

Рейтинг операционных систем представлен на рисунке 6.

Изображение выглядит как снимок экрана, диаграмма, Красочность, текст

Автоматически созданное описание

Рисунок 6 – Рейтинг операционных систем

**4.2.5 Язык программирования**

Для разработки десктопного приложения и Web API был выбран язык C#, поскольку:

* интуитивно понятный;
* имеет возможность наследования и переопределения;
* поддерживается Visual Studio;
* имеет атрибуты;
* прост в написании кода;
* можно сократить код за счёт имеющихся возможностей.

Для разработки мобильного приложения был выбран язык Java, поскольку:

* поддерживается в Android Studio;
* имеет возможности наследования и переопределения;
* созданное мобильное приложение можно установить на любое устройство с операционной системой Android;
* прост в реализации многопоточности (Необходимо для работы с запросов Web API).

Рейтинг языков программирования представлен на рисунке 7.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 7 – Рейтинг языков программирования

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы по преддипломной практике, была разработана информационная система для музыкального магазина. Была изучена предметная область. Также, был проведён анализ аналогов.

Были изучены:

* создание десктопное приложение в Visual Studio;
* создание Web API в Visual Studio;
* создание мобильных приложений в Android Studio;
* создание баз данных на Microsoft SQL Server;
* языки программирования Java и C#.

Была полностью разработана база данных. Также, были частично разработаны десктопное приложение, Web API и мобильное приложение.

В перспективе стоят следующие задачи:

* доделать функционал мобильного приложения до функционала десктопного приложения (возможно, дополнив Web API);
* разработать формирование заказа и просмотр заказов в десктопном и мобильных приложениях (естественно, дополнив Web API);
* сделать функционал для остальных ролей в десктопном приложении;
* сделать инсталлятор для десктопного приложения;
* протестировать WebAPI на Postman.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Бен-Ган, Ицик Microsoft SQL Server 2012. Создание запросов. Учебный курс Microsoft (+ CD-ROM) / Ицик Бен-Ган , Деян Сарка , Рон Талмейдж. - Москва: Машиностроение, 2022. - 720 c.
2. Гончарова А.Г., Леонтьева Т.В. Основные проблемы при разработке графической составляющей мобильного приложения
3. Никифоров, И. В. Курсовое проектирование по учебной дисциплине "Наука о данных и аналитика больших объемов информации": Учебное пособие / И. В. Никифоров. – Санкт-Петербург: Федеральное государственное автономное образовательное учреждение высшего образования "Санкт-Петербургский политехнический университет Петра Великого", 2017. – 62 с. – ISBN 978-5-7422-5638-0
4. Ицик, Бен-Ган Microsoft SQL Server 2012. Высокопроизводительный код T-SQL. Оконные функции / Бен-Ган Ицик. - М.: Русская Редакция, 2022. - 788 c.
5. Скрипчук А.В., Воинов Н.В., Каплан Е.В. Клиент-серверное приложение для управления паролями
6. developer.android.com – руководство по Android Studio
7. habr.com – хабр
8. learn.microsoft.com – сайт описания продуктов Microsoft
9. metanit.com – метанит
10. ru.wikipedia.org – википедия

**ПРИЛОЖЕНИЕ A. Структура репозитория на Git Hub**

Исходный текст программы находится в репозитории на Git Hub. Ссылка на репозиторий – <https://github.com/AntonSidorov1/DiplomProject>.

Скрин данного репозитория представлен на рисунке 8.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 8 – Репозиторий на Git Hub

**ПРИЛОЖЕНИЕ Б. Исходный текст программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace MusicShopDesktopApp

{

public partial class Authorize : Form

{

public Authorize(bool registrate = false)

{

this.registrate = registrate;

InitializeComponent();

}

bool registrate = false;

private void Model\_Load(object sender, EventArgs e)

{

if(registrate)

{

Text = "Регистрация";

labelTitle.Text = "Регистрация";

buttonSignIn.Text = "Зарегистрироваться";

checkBoxLogIn.Visible = true;

}

this.Text += " - " + Application.ProductName + " - " +

Application.ProductVersion;

notifyIconApp.Text = Text;

}

private void buttonExit\_Click(object sender, EventArgs e)

{

Close();

}

private void timerDateTime\_Tick(object sender, EventArgs e)

{

DateTime now = DateTime.Now;

string date = now.ToLongDateString();

string time = now.ToLongTimeString();

toolStripStatusLabelDate.Text = date;

toolStripStatusLabelTime.Text = time;

}

private void checkBoxShowPassword\_CheckedChanged(object sender,

EventArgs e)

{

textInputPassword.UseSystemPasswordChar = !(sender as

CheckBox).Checked;

textInputPassword.PasswordChar = '\0';

}

private void buttonCancel\_Click(object sender, EventArgs e)

{

Close();

}

private void buttonSignIn\_Click(object sender, EventArgs e)

{

if(!DataBaseConfiguration.Check())

{

MessageBox.Show("Не удаётся подключиться к базе данных",

"Авторизация", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

DataBaseConnectionEdit connectionEdit = new DataBaseConnectionEdit();

try

{

Helper.GetEnvirontmentByUser();

}

catch

{

if (!connectionEdit.Check())

{

MessageBox.Show("Не удаётся подключиться к базе данных",

"Авторизация", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

string login = textInputLogin.Value;

string password = textInputPassword.Value;

Account account = new Account

{

Login = login,

Password = password

};

if (!registrate)

{

try

{

Helper.Session = SessionsController.

GetController().

SignIn(Helper.EnvirontmentSession, account);

MessageBox.Show("Вы успешно авторизировались",

"Авторизация", MessageBoxButtons.OK, MessageBoxIcon.Information);

Close();

return;

}

catch(UserException ex)

{

MessageBox.Show(ex.Message,

$"Ошибка авторизации ({ex.Code})",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch(Exception ex)

{

MessageBox.Show("Неверный логин или пароль,

или пользователь заблокирован \n" +

"Или, пользователь не имеет ролей в системе \n" +

"Или, подключиться к базе данных не удаётся", "Авторизация",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

if (login == "")

{

MessageBox.Show("Введите логин", "Регистрация",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (password == "")

{

DialogResult result = MessageBox.Show("Пользователь без пароля

уязвим \n" +

"Вы согласны с этим?", "Регистрация",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if(result == DialogResult.No)

{

MessageBox.Show("Пользователь не был создан", "Регистрация",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

try

{

AccountsController controller = new AccountsController();

RegistrateSession session = controller.Registrate(account,

Helper.Session);

MessageBox.Show("Вы успешно зарегистрировались",

"Регистрация", MessageBoxButtons.OK,

MessageBoxIcon.Information);

if (!checkBoxLogIn.Checked)

{

Close();

return;

}

else

{

RegistrateEnvirontmentSession registrate = new

RegistrateEnvirontmentSession(session.RegistrateToken);

registrate.EnvirontmentToken =

SessionsController.GetController().

SignOut(Helper.Session).EnvirontmentToken;

Helper.Session = SessionsController.

GetController().

SignInByRegistrateToken(registrate);

MessageBox.Show("Вы успешно авторизировались",

"Авторизация",

MessageBoxButtons.OK, MessageBoxIcon.Information);

Close();

return;

}

}

catch

{

MessageBox.Show("Невозможно зарегистрироваться \n" +

" Возможно логин уже существует", "Регистрация",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}

**ПРИЛОЖЕНИЕ В. Работа программ**

Далее будут представлены окна десктопного и мобильного приложений, а также Web API.

**Примечания**

Окна приложений (особенно, где выводится списки и информация) могут незначительно отличаться от представленных в данном приложении.

**Десктопное приложение**

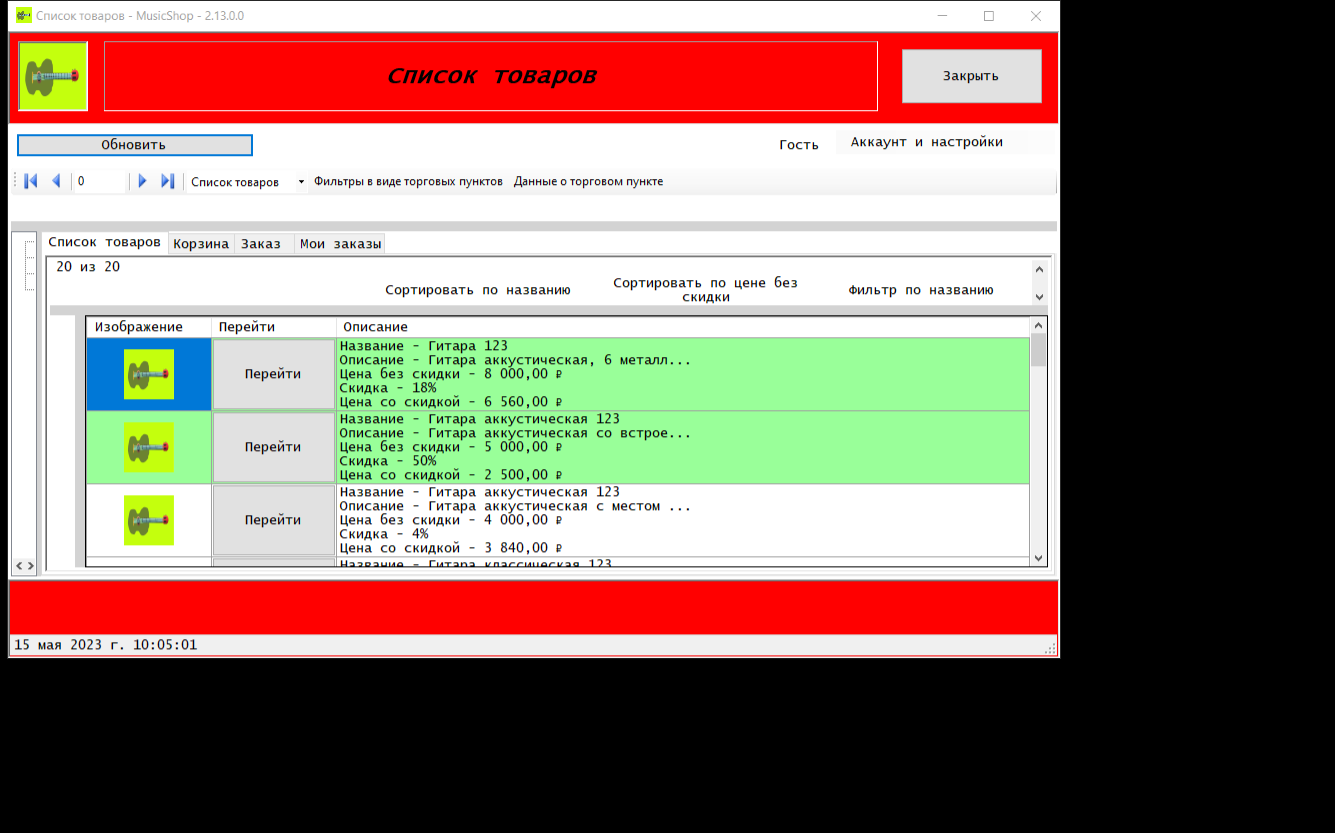


Рисунок 9 – Список товаров

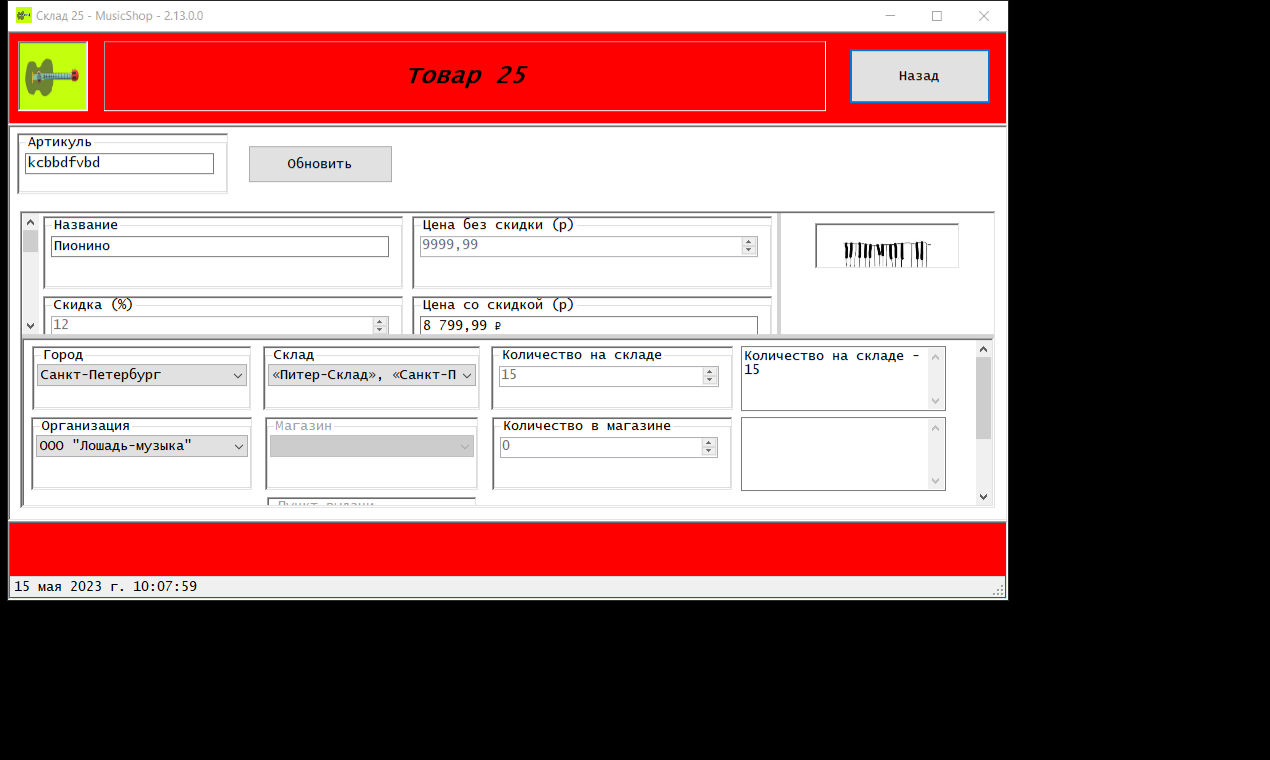


Рисунок 10 – Информация о товаре

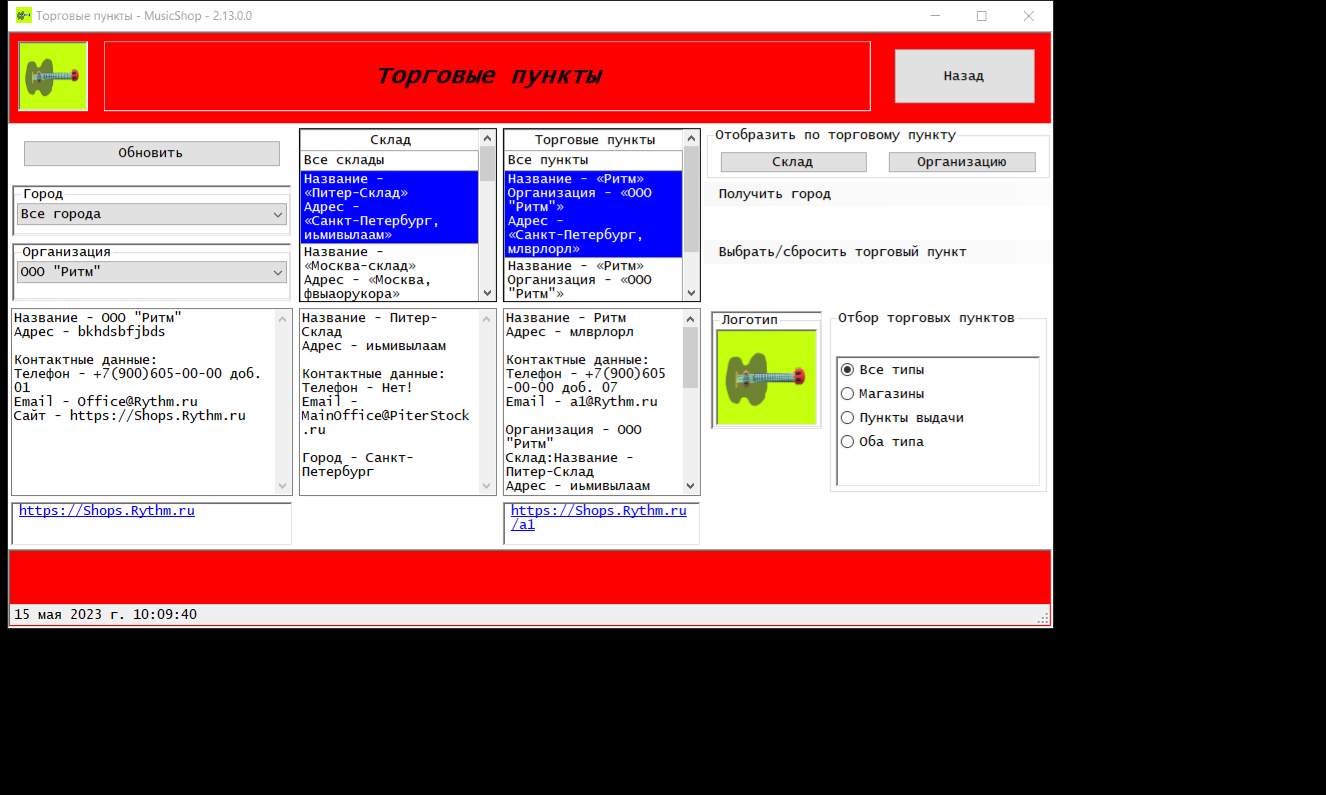


Рисунок 11 – Выбор торгового пункта для выдачи заказа, и для просмотра количества товаров

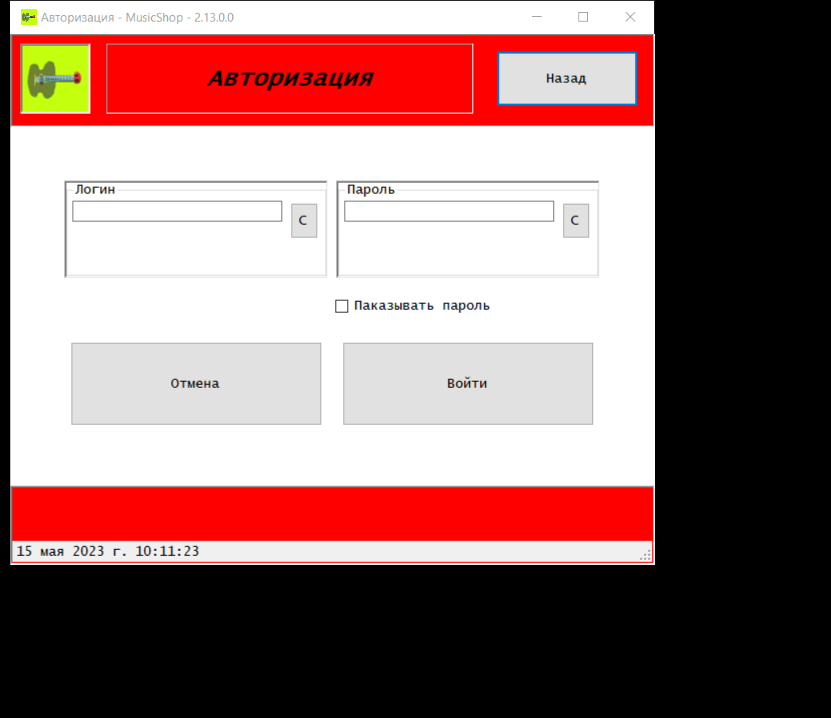


Рисунок 12 – Авторизация

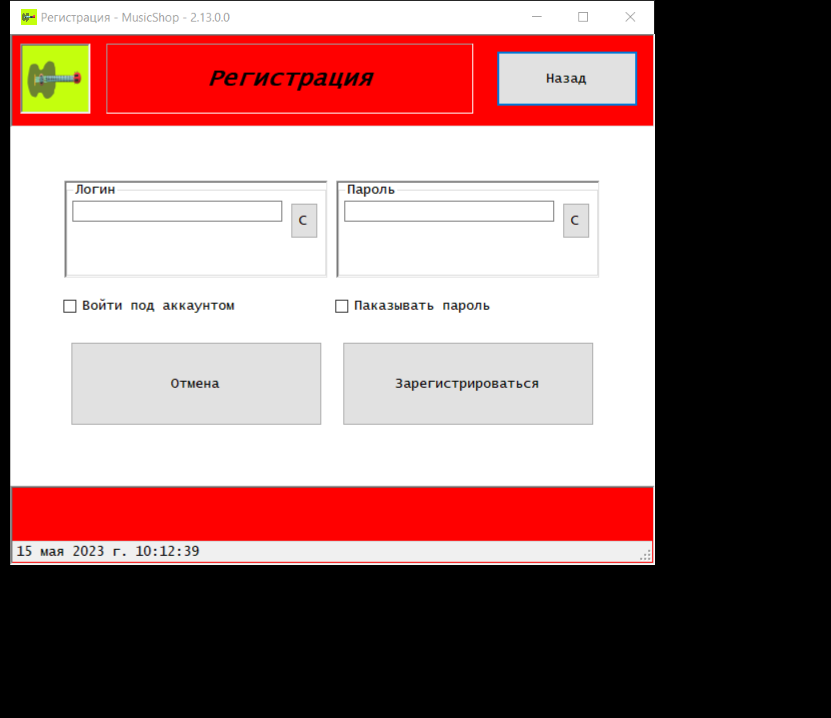


Рисунок 13 – Регистрация



Рисунок 14 – Редактирование аккаунта

**Web API**

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 15 – Аккаунты

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 16 – Категории товаров

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 17 – Фильтры категорий

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 18 – Email-адреса пользователя

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 19 – Организация (торговая сеть)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 20 – Товары

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 21 – товары в магазинах

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 22 – Товары на складах

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

Рисунок 23 – Сессии

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 24 – Города

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 25 – Параметры сортировки

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 26 – Склады

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 27 – Торговые пункты (магазины и пункты выдачи)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 28 – Роли пользователя

**Мобильное приложение**

Изображение выглядит как текст, электроника, Мобильный телефон, Устройство связи

Автоматически созданное описание

Рисунок 29 – Список товаров

Изображение выглядит как текст, Устройство связи, гаджет, Мобильное устройство

Автоматически созданное описание

Рисунок 30 – Просмотр информации о товаре

Изображение выглядит как гаджет, Устройство связи, текст, Мобильное устройство

Автоматически созданное описание

Рисунок 31 – Авторизация

Изображение выглядит как гаджет, Устройство связи, Мобильное устройство, Портативное устройство связи

Автоматически созданное описание

Рисунок 32 – Регистрация

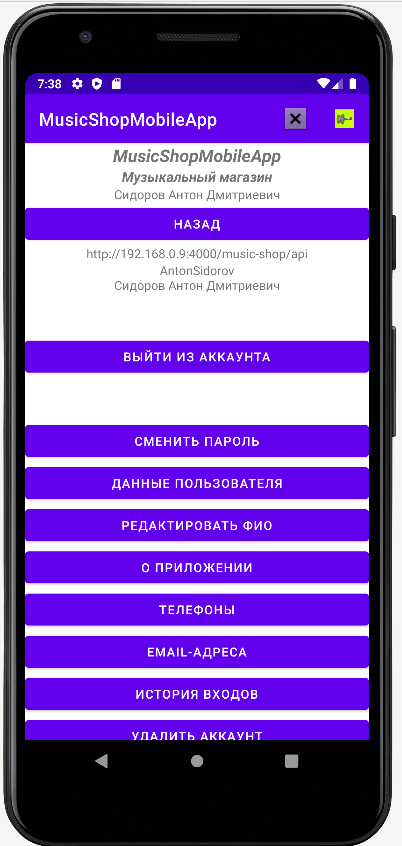


Рисунок 33 – Редактирование аккаунта