

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

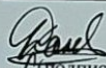
Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»
Магистерская программа: «Прикладной искусственный интеллект»

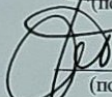
МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
на тему
**ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ДЛЯ
ОБРАБОТКИ, ХРАНЕНИЯ И ПЕРЕДАЧИ
ДАННЫХ**

Студент


(подпись, дата)

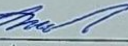
Коновалова Дарья Алексеевна
(ФИО полностью)

Руководитель


(подпись, дата)

Федюнин Р.Н.
(фамилия, инициалы)

Нормоконтролёр

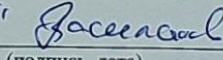

(подпись, дата)

Бычков А.С.
(фамилия, инициалы)

Рецензент

21.06.2024

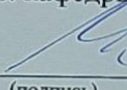
доцент кафедры
(должность, место работы)


(подпись, дата)

Васильев А.В.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 27.05.2024 г. № 11)

Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от

25.06.2024, № 46)
Секретарь ГЭК 
(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 13 » сентября 2022 г.

**ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ**

1. Студент Коновалова Дарья Алексеевна гр. 22ВВИм1 факультета вычислительной техники
2. Тема работы «Интеллектуальная система для обработки, хранения и передачи данных»
Тема утверждена приказом ПГУ №765/о от "13" сентября 2022 г.
3. Руководитель магистерской диссертации к.т.н., доцент Федюнин Р.Н.
4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать программу для работы книжного интернет-магазина. Приложение должно реализовывать следующие требования: авторизация пользователя в программе, интерфейс для покупателей и администратора, редактирование каталога товаров, отображать данные о продажах в виде таблиц и графиков.

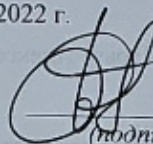
5. Перечень подлежащих разработке вопросов:

1. Теоретический анализ предметной области.
 2. Обзор и сравнительный анализ необходимых средств и инструментов для разработки программы.
 3. Создание базы данных для хранения информации. Разработка приложения для персонального компьютера.
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Теоретический анализ предметной области	10%	21.09.2022 – 21.11.2022	
2.	Обзор и сравнительный анализ средств для разработки	10%	22.11.2022 – 22.01.2023	
3.	Разработка базы данных	40%	23.01.2023 – 23.10.2023	
4.	Разработка приложения	30%	24.10.2024 – 30.03.2024	
5.	Оформление выпускной работы	10%	31.03.2024 – 31.05.2024	

Дата выдачи задания " 13 " сентября 2022 г.

Руководитель магистерской диссертации



(подпись, дата)

Федонин

Задание к исполнению принял студент



(подпись, дата)

Коновалова

Работу к защите допустить
Декан факультета



25.06.2024

(подпись, дата)

Фионов

РЕФЕРАТ

Магистерская диссертация содержит 87 листов, 40 рисунков, 1 таблицу, 21 источник, 2 приложения

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, ВИЗУАЛИЗАЦИЯ, АНАЛИЗ, ОБРАБОТКА ИНФОРМАЦИИ, БАЗЫ ДАННЫХ

Объектом исследования является извлечение данных о продажах книжного интернет-магазина, программное обеспечение, позволяющее решать практические и интеллектуальные задачи.

Предмет исследования – статистика продаж товаров, купленных в книжном интернет-магазине.

Цель работы – повышение эффективности работы специалистов, ответственных за аналитику продаж товаров. Проектирование и разработка приложения для извлечения данных с возможностью его дальнейшего включения в интеллектуальную систему. Данная цель обусловлена необходимостью иметь структурированные и подготовленные для анализа данные. Проблема сложности анализа информации человеком без подобных систем актуальна для продавцов малого бизнеса.

В ходе работы были выработаны требования к разрабатываемому приложению, спроектирована архитектура и разработано приложение для книжного интернет-магазина, а также определено направление дальнейшей работы над проектом.

					ПГУ 1.090401.10.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Интеллектуальная система для обработки, хранения и передачи данных Пояснительная записка	Лит.	Лист	Листов	
Разраб.		Коновалова Д.А.		24.06.24					
Провер.		Бедюнин Р.Н.		24.06.24			2	87	
И. Контр		Бычков А.С.		24.06.24		Кафедра ВТ, гр. 22ВВИм1			
Утверд.		Митрохин М.А.		24.06.24					

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ.....	5
ВВЕДЕНИЕ.....	6
1 Анализ предметной области.....	8
1.1 Информационные системы.....	8
1.2 Интеллектуализация информационных систем.....	10
1.3 Архитектура интеллектуальных систем.....	12
1.3.1 Структура интеллектуальных информационных систем.....	13
1.3.2 Интеллектуальный интерфейс.....	14
1.3.3 Разработка и проектирование интеллектуальных систем.....	17
1.4 Анализ существующих аналогов разрабатываемого.....	19
1.4.1 1С: Розница.....	20
1.4.2 «Управление ассортиментом» Эвотора.....	22
1.4.3 SUBTOTAL.....	24
1.4.4 МойСклад.....	25
1.4.5 LiteBox.....	26
1.4.6 Антисклад.....	27
1.4.7 Вывод.....	29
1.5 Анализ баз данных.....	29
1.6 Выводы.....	34
2 Обоснование использования заданных средств разработки.....	35
2.1 Анализ ОС для функционирования приложения.....	35
2.2 Анализ платформ для разработки приложения.....	36
2.2.1 QT Creator.....	38
2.2.2 Eclipse.....	39
2.2.3 Microsoft Visual Studio.....	40
2.2.4 IntelliJ IDEA.....	41
2.2.5 NetBeans.....	42
2.3 Вывод.....	43
3 Техническое задание.....	44
3.1 Назначение разработки.....	44
3.2 Требования к программе.....	44

3.3 Стадии и этапы разработки.....	46
3.4 Выводы.....	46
4 Описание программы.....	47
4.1 Общие сведения.....	47
4.2 Функциональное назначение.....	47
4.3 Проектирование взаимодействия между клиентом и приложением.....	47
4.4 Архитектура программы.....	48
4.5 Архитектура базы данных.....	49
4.6 Структура хранения данных.....	51
4.7 Отладка и тестирование.....	53
4.8 Руководство пользователя по использованию программы.....	54
4.9 Вывод.....	61
ЗАКЛЮЧЕНИЕ.....	62
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	63
ПРИЛОЖЕНИЕ А Листинг программы.....	65
ПРИЛОЖЕНИЕ Б Презентация.....	80

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

ИС – интеллектуальная система

ОС – операционная система

ПК – персональный компьютер

БД – база данных

SQL (Structured Query Language) – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных

URL (Uniform Resource Locator) – адрес ресурса в сети Интернет

ВВЕДЕНИЕ

Интеллектуальные информационные системы проникают во все сферы нашей жизни, поэтому становятся неотъемлемым элементом при решении задач автоматизации и управления сложными объектами и процессами.

Современное понятие интеллектуальных систем сформировалось в процессе развития теоретических основ кибернетики, современной теории управления, теории алгоритмов, развития современных информационных технологий и обобщения накопленных научных знаний, методов и средств в области искусственного интеллекта.

Одним из направлений в области искусственного интеллекта являются интеллектуальные информационные системы. Интеллектуальные информационные системы — это естественный результат развития обычных информационных систем. Они сосредоточили в себе наиболее наукоемкие технологии с высоким уровнем автоматизации не только процессов подготовки информации для принятия решений, но и самих процессов выработки вариантов решений, опирающихся на полученные информационной системой данные.

Современный этап развития информационных систем можно обозначить как этап их интеллектуализации. Разработка информационных систем без интеллектуализации уходит в прошлое. Какими интеллектуальными свойствами должна обладать информационная система, какие структурные части должны быть интеллектуальными — это современные актуальные проблемы.

Помимо упорядочивания данных, визуализация также помогает в обнаружении пропущенных значений и выявлении скрытых паттернов в данных. Этот подход становится важным средством для анализа информации и принятия обоснованных

решений [1].

Таким образом, анализ данных через визуализацию и работу с графикой играет ключевую роль как в научной сфере, так и в бизнесе, способствуя более эффективному и качественному использованию информации для прогнозирования и принятия стратегических решений.

На сегодняшний день на рынке программного обеспечения существует множество продуктов, предназначенных для обработки данных, а также для разработки и визуализации информации [2].

Рассмотрим стандартные возможности таких программ:

- наличие непосредственной связи с базами данных или возможности выполнения разовых процедур импорта данных из формата хранения популярных баз данных и табличных редакторов,
- наличие возможности редактирования и/или просмотра в табличном режиме данных, поступающих из подключенной базы данных, введенных в ручном режиме или импортированных извне,
- возможность выбора типа диаграммы или графика, используемого для отображения данных, настройки цветов линий и плоскостей на графике, ориентации пределов измерений шкал, подписей и т. д.,
- возможность манипуляции шкалами: нормирование отображаемых величин, установление масштаба отображения, изменение закона распределения делений на шкалах,
- возможность вывода на печать и экспорта в другие программы и сохранения представления в иных форматах.

Чтобы эффективно решить проблему отображения данных, разработчики программного обеспечения часто прибегают к сохранению результатов обработки в форматах, совместимых с распространенными программами для просмотра данных. Это делает возможным удобный просмотр и использование данных с

помощью встроенных средств отображения в популярных программных продуктах [3].

1 Анализ предметной области

Для работы эффективной торговли необходим анализ продаж. Анализ продаж — это структурированный разбор данных о продажах компании, который позволяет выявить, что работает, что не работает, и определить возможности для улучшения. Анализ помогает оценить эффективность работы команды продаж и выявить области для улучшения, выявить тенденции и закономерности в продажах, чтобы разработать стратегии для улучшения результатов.

Проанализировать финансовые показатели продаж, такие как выручка, рентабельность, стоимость привлечения клиентов могут быть одними из ключевых факторов для развития книжного интернет-магазина.

Входными данными для данной программы является информация о покупках пользователей книжного интернет-магазина.

Выходными данными являются визуальное представление статистики продаж за выбранный период, представленное в виде таблицы и графиков.

Основные задачи, которые необходимо решить за счёт данного приложения:

- ☒ получение данных по продажам товара,
- ☒ визуализация данных в виде графиков и таблиц,
- ☒ создание приложения для книжного интернет-магазина.

Таким образом, разработка программы для обработки, хранения и визуализации информации актуальна и востребована.

1.1 Информационные системы

Современное состояние информационных систем и технологий можно охарактеризовать следующими тенденциями:

– наличие большого количества промышленно функционирующих баз данных большого объема, содержащих

информацию практически по всем видам деятельности общества,

- создание технологий, обеспечивающих интерактивный доступ массового пользователя к этим информационным ресурсам. Технической основой данной тенденции появились государственные и частные системы связи и передачи данных общего назначения и специализированные, объединенные в национальные, региональные и глобальные информационно-вычислительные сети,

- расширение функциональных возможностей информационных систем, обеспечивающих параллельную одновременную обработку баз данных с разнообразной структурой данных, мульти объектных документов, гиперсреды, в том числе реализует технологии создания и ведения гипертекстовых баз данных. Создание локальных, многофункциональных проблемно ориентированных информационных систем различного назначения на основе мощных персональных компьютеров и локальных вычислительных сетей,

- включение в информационные системы элементов интеллектуализации интерфейса пользователя, экспертных систем, систем машинного перевода, автоиндексирования и других технологических средств.

В результате чего, выделяют пять основных тенденций в развитии информационных систем и технологий.

Осложнения информационных продуктов (услуг). Информационный продукт в виде программных средств, баз данных и служб экспертного обеспечения приобретает стратегическое значение.

Способность к взаимодействию. С ростом значимости информационного продукта возможность провести идеальный обмен этим продуктом между компьютером и человеком или между информационными системами приобретает значение ведущей

технологической проблемы. Также эта проблема касается совместимости технических и программных средств. Все проблемы обработки и передачи информационного продукта находились в полном соответствии совместимости и быстродействия.

Ликвидация промежуточных звеньев. Развитие способности к взаимодействию ведет к совершенствованию процесса обмена информационным продуктом, а значит, при взаимоотношении поставщиков и потребителей в этой области ликвидируются промежуточные звенья. Не нужны посредники, если есть возможность размещать заказы непосредственно с помощью информационных технологий.

Глобализация. Фирмы могут с помощью информационных технологий вести дела где угодно, получая исчерпывающую информацию. Глобализация рынка информационного продукта нацелена на получение преимуществ за счет распределения постоянных и полупостоянных расходов на более широкий географический регион.

Конвергенция. Исчезают различия между изделиями и услугами, информационным продуктом и средствами, использованием в быту и для деловых целей, информацией и развлечением, а также среди различных режимов работы, таких как передача звуковых, цифровых и видеосигналов.

Данные тенденции приводят к осуществлению распределенных персональных вычислений, когда на каждом рабочем месте достаточно ресурсов для обработки информации в местах ее возникновения; созданию развитых систем коммуникаций, когда рабочие места соединены для пересылки сообщений; гибким глобальным коммуникациям, когда предприятие включается в мировой информационный поток; созданию и развитию систем электронной торговли; устранению промежуточных звеньев в

системе интеграции организация - внешняя среда.

1.2 Интеллектуализация информационных систем

Реализация интеллекта в информационных системах (ИС) имеет свою историю. Сначала это была реализация стратегических выигрышных алгоритмов, такие стратегии использовались для реализации различных компьютерных игр. Далее происходило внедрение основ логики и элементов логического вывода. Начали появляться так называемые экспертные системы.

Сегодня ставится проблема интеллектуализации компьютера. А это требует совсем другого подхода, который включает в себя интеллектуализацию интерфейса, возможность использования и накопления экспертных знаний (накопление опыта), вопросы обучения и самообучения, которые основаны на идеях нейроинформатики, накопление не только знаний из узкой предметной области, но и вообще метапредметных знаний. Необходимо отдельно рассматривать следующие методы интеллектуализации ИС:

- интеллектуализация интерфейса ИС,
- ИС управления предприятием,
- ИС управления технологическими процессами, производством,
- информационные технологии защиты информации и безопасности сетей и коммуникаций,
- ИС распознавания символов, образов, речи, перевода,
- интеллектуализация геоинформационных систем,
- интеллектуальные ИС для медицины.

Решение методов интеллектуализации всех перечисленных ИС и их структурных частей позволит приблизиться к созданию интеллектуального компьютера.

Рассмотрим особенности реализации интеллектуализации информационных систем. Для интеллектуализации ИС управления

предприятием необходимо реализовать цикл управления: накопление информации – анализ информации – принятие решения – доведение решения до исполнения.

Главным этапами интеллектуализации, конечно, являются анализ и принятие решения. Следовательно, данная информационная система должна содержать как экспертные знания, так и знания законов эффективности управления предприятием, должна уметь построить различные ситуативные модели и предложить наиболее оптимальные. На этапе накопления информации могут использоваться интеллектуальные речевые технологии.

Для интеллектуализации управления технологическими процессами должен быть автоматизирован и ввод актуальной информации, то есть должны использоваться датчики различных параметров. Причем такие системы должны быть обучаемы и самообучаемы.

В информационных технологиях защиты информации и безопасности сетей и коммуникаций должны использоваться так называемые интеллектуальные агенты, которые могут распознавать своих и не допускать в систему чужих агентов и пользователей. Агентный подход удобно реализовать и для разработки интеллектуальных операционных систем.

Для интеллектуализации геоинформационных систем требуется возможность детализации в системе картографической информации до конкретных реальных объектов: карта – масштабирование карты – детализация объектов. Современные спутниковые технологии уже позволяют реализовать первые два пункта данной цепочки.

Интеллектуальные медицинские системы требуют визуализации проникновения специалистов в самые сложные органы человеческого организма.

Таким образом, для формирования современных интеллектуальных информационных систем требуется решение следующих научных проблем:

- разработка оптимальных методов визуализации различных объектов и процессов,
- методология обучения и самообучения информационной системы,
- снабжение инструментария разработки информационных систем различными готовыми интеллектуальными агентами (сервисами), в том числе и по защите информации и целостности баз данных,
- снабжение разработчиков инструментами оптимизации баз знаний и банков данных,
- создание адаптируемых (абстрактных) интеллектуальных информационных систем.

1.3 Архитектура интеллектуальных систем

Интеллектуальная информационная система (ИИС) – это взаимосвязанная совокупность средств, методов и персонала, имеющая возможность хранения, обработки и выдачи информации, а также самостоятельной настройки своих параметров в зависимости от состояния внешней среды (исходных данных) и специфики решаемой задачи.

1.3.1 Структура интеллектуальных информационных систем

Говоря об архитектуре ИИС, прежде всего, понимают организацию структуры, в рамках которой происходило бы применение знаний и решение проблем в конкретной предметной области. Выбор соответствующей структуры, свойства и функции компонентов систем ИИС, определяется и направляется формулируемыми принципами инженерии знаний.

В зависимости от характера выполняемых функций и области

действий, эксперты выполняют несколько характерных задач, которые являются типичными. Их анализ служит ориентиром при рассмотрении архитектуры производственных систем, основанных на знаниях. Эти задачи, следующие: интерпретация, планирование, управление, проектирование, прогнозирование, диспетчирование, мониторинг и диагностика. А главное — эксперт способен обновлять свои знания, обучаться, объяснять действия, обосновывать решения, прогнозировать развитие ситуаций, активно взаимодействовать с внешней средой и воспринимать информацию различного характера, получать решения на основе имеющихся знаний, хранить в памяти необходимую информацию и фактографические данные.

Таким образом, чтобы создать систему, работающую со знаниями и способную в какой-то мере заменить эксперта или помочь ему в принятии решений при управлении производством, необходимо стремиться заложить в архитектуру нашей системы возможности по реализации названных функций.

Производственная система ИИС организует и направляет свое функционирование в соответствии с воплощенными в нее архитектурными проектными принципами. Для каждой управляющей схемы могут потребоваться своя база знаний и соответствующий механизм вывода, который работает со знаниями. Обычно интеллектуальные системы взаимодействуют с конечными пользователями, экспертом, инженером знаний, внешними базами данных, прикладным программным обеспечением.

Пользовательский интерфейс обеспечивает связь на ограниченном естественном языке, речевой ввод, а также визуальные представления (графику, техническое зрение). В качестве пользователя может выступать либо человек-оператор, либо сам производственный процесс, в случае закрытых циклических операций. Для некоторых производственных процессов,

бывают, необходимы средства для автоматического получения данных и их обработки, а также обратной связи по управлению. С инженером знаний интеллектуальная система обычно связывается с помощью структурных редакторов, которые позволяют ему получать и модифицировать компоненты базы знаний.

Взаимодействие интеллектуальной системы с прикладным программным обеспечением осуществляется при выполнении специальных вычислений, так как часто возникает необходимость использовать в качестве подзадач стандартные операции по обработке данных.

Связь с распределенной базой данных интегрированной системы управления производством и Интернет используется интеллектуальными системами для получения данных и знаний, рассредоточенных на различных уровнях иерархии управления. Кроме того, организуется взаимодействие с внешними базами данных и Интернет.

При проектировании интеллектуальных систем и выборе их архитектуры следует разрабатывать не просто независимое программное обеспечение, которое оценивает существующий производственный объект. Надо стремиться к тому, чтобы интеллектуальная система была приближена к различным элементам процесса, выступала в качестве одного из основных звеньев технологической цепочки управления и организации.

1.3.2 Интеллектуальный интерфейс

Производственные системы интеллектуального интерфейса функционируют в подавляющем большинстве реализаций в интерактивном режиме с пользователями, поэтому они должны обладать дружелюбным интерфейсом, позволяющим человеку легко и в удобной для него форме взаимодействовать с компонентами. Общение человека и ИС могут обеспечивать и реализовывать

различные программные и технические средства ввода и вывода информации.

Термин «пользовательский интерфейс» охватывает все аспекты взаимодействия между пользователем и интеллектуальной системой. Он включает не только техническое и программное обеспечение, но также факторы, которые связаны с обеспечением использования, доступности и человеко-машинного взаимодействия. Многие специалисты определяют пользовательский интерфейс как наиболее важный компонент системы, так как большинство характеристик, таких как эффективность, гибкость, легкость использования являются производными этого компонента.

Развитие способностей и возможностей комфортного и качественного взаимодействия пользователя с системой, которая организует, предоставляет этот компонент, позволяет говорить об интеллектуальном интерфейсе.

Подсистема интеллектуального интерфейса управляется программным обеспечением, называемым управляющая система интеллектуального интерфейса. Эта управляющая система состоит из нескольких программ, которые обеспечивают способности, некоторые из которых перечислены ниже:

- обеспечение графического пользовательского интерфейса,
- организация взаимодействия пользователя с различными входными устройствами,
- представление данных с различными форматами и на разные входные устройства,
- представление пользователю помощи, подсказок, советов, диагностического режима работы или другой гибкой поддержки,
- обеспечение взаимодействия с базой данных и базой моделей,
- хранение входных и выходных данных,

- обеспечение цветной графики, трехмерной графики,
- окна, позволяющие отображать множество функций одновременно,
- поддержка взаимодействия между пользователями и разработчиками системы,
- обеспечение обучения на примерах,
- обеспечение гибкости и адаптивности,
- взаимодействовать во многих различных стилях диалога.

Вид интерфейса определяет то, как информация введена и отображена. Он также определяет легкость и простоту обучения и использования системы.

Существуют следующие виды интерфейса. Взаимодействие на основе меню. При этом виде взаимодействия пользователь выбирает позицию или пункт из списка возможных выборов(меню) для того, чтобы функция была выполнена. Меню появляются в логическом порядке, начиная с главного меню и продвигаясь к локальным меню. Пункты меню могут включать команды, которые появляются в отдельных локальных меню или в меню с не командными пунктами. Меню может оказаться утомительным и продолжительным по времени, когда анализируются сложные ситуации, т.к. это может потребовать несколько меню для построения или использования системы и пользователь должен перемещаться назад и вперед меню.

Командный язык. При этом виде пользователь вводит команды. Многие команды включают комбинации «глагол-существительное». Некоторые команды могут исполняться с функциональными клавишами. Другим способом упрощения команд является использование макросов. Команды могут также вводиться голосом.

Вопросно-ответная форма интерфейса начинается с вопросов компьютера пользователю. Пользователь отвечает на вопросы

фразой или предложением (или выбором пункта меню). Компьютер может подсказывать пользователю для прояснения или дополнительного ввода информации. В некоторых применениях порядок вопросов может быть обратным: пользователь задает вопросы, а компьютер дает ответы.

Формирование взаимодействия. Пользователь вводит данные или команды в обозначенные формы (поля). Заголовки формы (или отчета, или таблицы) служат подсказками для ввода. Компьютер может представлять какой-то выход как результат, и пользователь может быть спрошен о продолжении интерактивного процесса.

Естественный язык. Взаимодействие человек – компьютер, которое подобно диалогу человека с человеком называется естественным языком. Сегодня диалог на естественном языке выполняется главным образом посредством клавиатуры. Такой диалог будет проводиться в будущем с использованием голоса для ввода и вывода информации. Главным ограничением использования естественного языка является по существу неспособность компьютера понимать естественный язык. Однако, достижения интеллектуального интерфейса все больше повышают уровень диалога на естественном языке.

Графический пользовательский интерфейс. В графическом пользовательском интерфейсе объекты обычно представляются как пиктограммы (или символы) и пользователь непосредственно ими манипулирует. Новейшие операционные системы компьютеров и их приложения исключительно основаны на графике.

1.3.3 Разработка и проектирование интеллектуальных систем

Проектирование интеллектуальных систем — это итеративный и эволюционный процесс, в котором участвуют несколько специалистов: эксперт, обладающий знаниями о предметной области

и желающий помочь работе по созданию системы, а также специалисты в области искусственного интеллекта -инженеры знаний, аналитики и программисты. В зависимости от объема и трудоемкости работ группа может состоять из трех - шести человек.

При оценке проблемной области на этапе проектирования интеллектуальных систем необходимо учитывать следующие факторы: легкость сбора данных, представимость данных, оправданность затрат на разработку интеллектуальных систем, наличие экспертов, наличие необходимых ресурсов (компьютеров, программистов, программного обеспечения).

После анализа проблемной области и определения целесообразности внедрения интеллектуальной системы в этой сфере приступают непосредственно к проектированию системы.

Существуют различные взгляды на определение числа этапов проектирования интеллектуальных систем. Это зависит от многих факторов, в частности от характера функций будущей интеллектуальной системы, области использования, наличия развитых инструментальных средств.

Процесс проектирования интеллектуальных информационных систем разделен на этапы.

Первый этап - идентификация определения задач и идентификация их характеристик. На этом этапе устанавливаются задачи, которые предполагается решать, их характеристики и особенности. Разрабатывается техническое задание на проектируемую систему. Далее очерчивается круг пользователей системы. Эти сведения помогут в дальнейшем правильно определить область знаний эксперта, определить функции системы и, как следствие, уровень необходимых знаний. В результате вырабатываются определенные требования.

Второй этап - выделение главных концепций предметной

области, которые отражают знания круга экспертов. Это дает возможность анализировать тип знаний, которыми оперирует эксперт в процессе принятия решений. Инженер по знаниям определяет формальные средства представления знаний и процедуры получения решений, которые в наибольшей степени соответствуют характеру рассуждений эксперта при выводе решения. Таким образом, в результате выполнения этого этапа выявляются и формулируются понятия, определяющие выбор характерной схемы представления знаний эксперта о предметной области.

Третий этап - выбор формализма представления знаний и определение механизма вывода решений. Эти компоненты моделирования в значительной степени влияют на успешное решение поставленной задачи по проектированию системы. Разработанная структура для представления знаний является основой для реализации следующего этапа — непосредственного построения базы знаний системы.

Четвертый этап - выбор или разработка языка представления знаний. После того как правила сформулированы и представлены на выбранном языке представления, они заносятся инженером знаний в БЗ.

Пятый этап - тестирование системы. Работоспособность системы определяется путем решения конкретных проверочных задач. При выявлении различных недостатков происходит обращение к тому или иному этапу разработки в зависимости от характера недостатков. При отсутствии каких-либо знаний в системе или их недостаточной определенности возвращаются к четвертому этапу и по возможности вносят поправки. В случае если какие-либо знания, представленные экспертом практически невозможно представить в пределах формализма выбранной модели представления знаний, то возвращаются к третьему этапу и

выбирают альтернативные модели или схемы представления знаний. Возможно, что причиной возврата может служить недостаточно адекватный базовый механизм логического вывода. Возникают ситуации, когда требуется переформулирование проблемы, так как исходная постановка задачи была неправильной.

1.4 Анализ существующих аналогов разрабатываемого

Во многих сферах жизни приходится обрабатывать и анализировать информацию. В основном ее собирают в табличном виде — это удобно для хранения, но не для восприятия. Чтобы упростить процесс, используют средства визуализации данных, инструменты для создания диаграмм и графиков. Они помогают быстро понять и проанализировать информацию, сравнить выполнение плана за разные периоды, увидеть динамику изменения показателей и сократить время принятия решений.

Рассмотрим сервисы визуального представления данных о продажах и их особенности [13].

Некоторые платформы работают исключительно в облачных пространствах, но есть и те, которые можно скачивать на компьютер или разворачивать на серверах компании. Одни предлагают свободный доступ, а другие — только по платной подписке. Также можно разделить эти инструменты по уровню компетенций, необходимых специалистам: одни работают по принципу no-code, а другие требуют знания языков программирования.

1.4.1 1С: Розница

Данная программа в значительной мере подойдет: розничным торговым сетям. Программа 1С для учета товара создана специально для розничной торговли. С ней можно вести товарный учёт в единичных магазинах, но учитывая объём опций, сложность внедрения и обучения, обычно её используют средние и крупные

торговые сети (рисунок 1).

Кроме товар учётной программы для розничной торговли в решении от 1С есть множество опций для магазинов:

- ☒ аналитика,
- ☒ управление складом и персоналом,
- ☒ управление товарооборотом и возвратами.
- ☒ работа с дисконтными картами.
- ☒ возможность подключать периферию,
- ☒ расчет зарплат,
- ☒ нишевые решения.

В программе учёта движения товара от 1С можно тонко настроить рабочее место кассира, оно имеет больше опций, чем в некоторых других программах. Также к ПО можно подключить любую периферию:

- ☒ онлайн-кассу, включая Эвоторы,
- ☒ сканер штрихкодов,
- ☒ принтер чеков,
- ☒ весы,
- ☒ считыватели карт, в том числе магнитных.

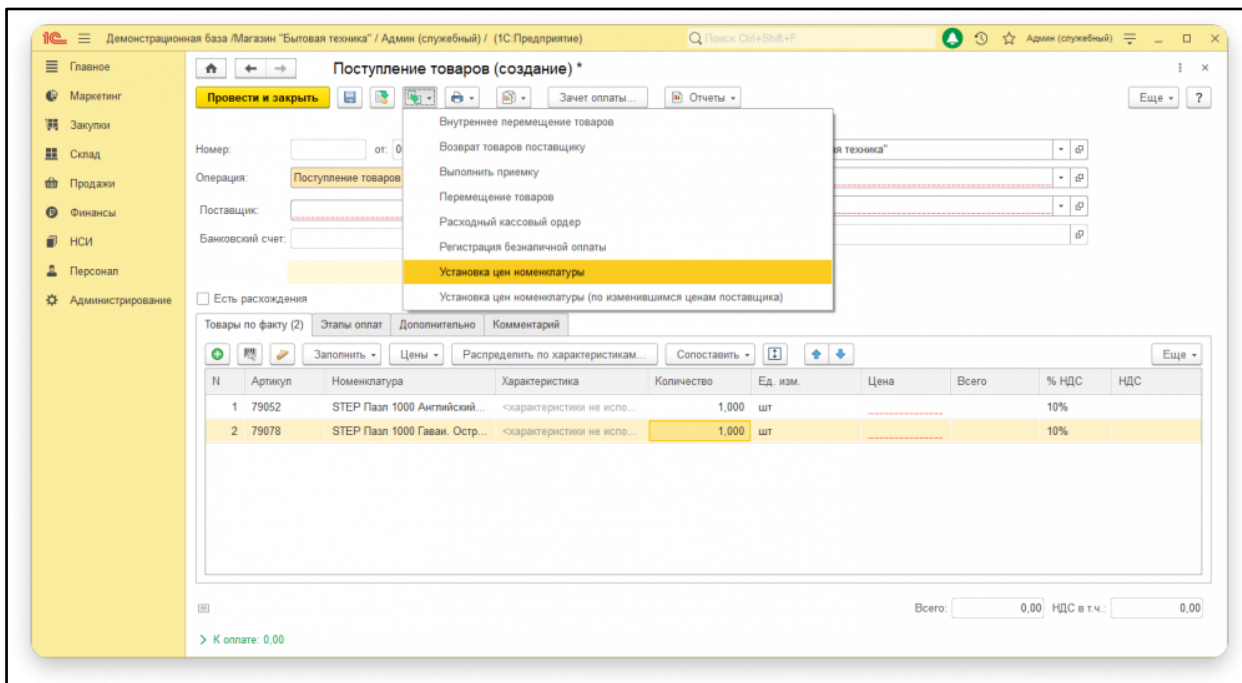


Рисунок 1 — Интерфейс 1С:Розница для создания товаров

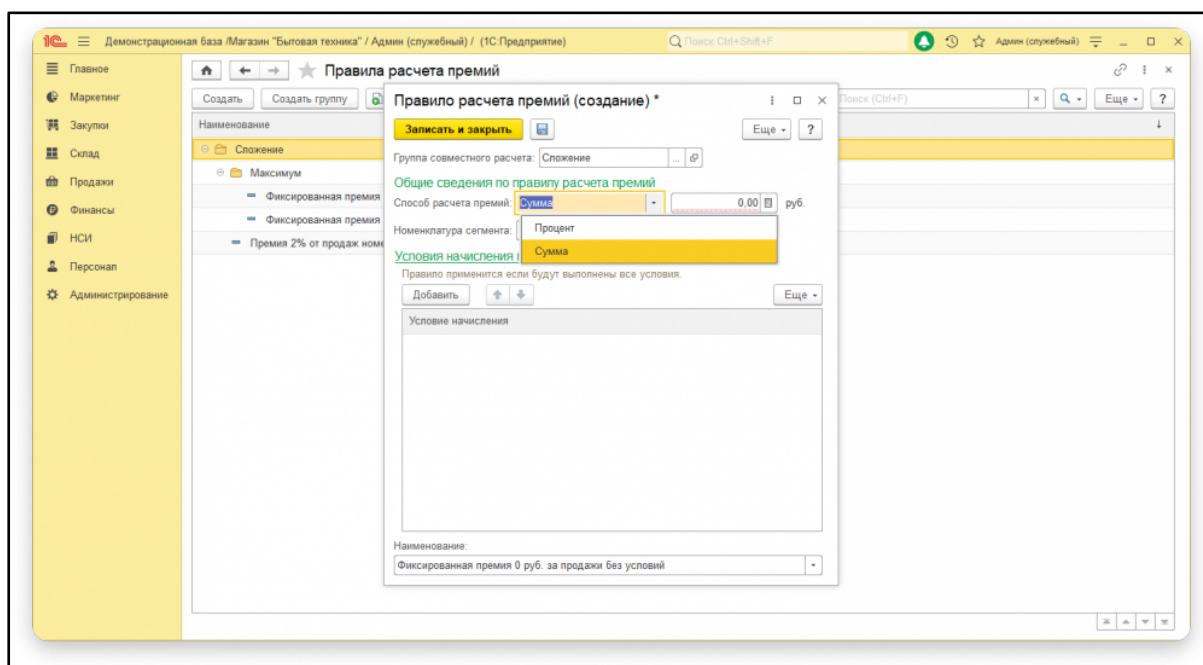


Рисунок 2 — Пример интеграции торговой оболочки и 1С
Предприятие.

Можно фасовать продукцию на развес и печатать для неё этикетки. «1С: Розница» умеет анализировать, какие товары кончаются быстро, а какие залеживаются, что позволяет эффективнее планировать закупки.

У программы для учёта товаров в розничной торговле глубокая интеграция с другими сервисами 1С — можно не переживать о выгрузке данных в бухгалтерию и подключить дополнительное ПО из экосистемы 1С (рисунок 2).

Аналогичным образом выглядят и другие подобные приложения, отличающиеся видом оболочки интерфейса, расположением приложений на панели и ориентацией на конкретные виды товара. Некоторые из них ориентированы на продажи одежды и обуви, другие на доставку продуктов питания и готовой еды через интернет заказы.

1.4.2 «Управление ассортиментом» Эвотора

На рисунке 3 представлена главная страница в сервисе «Управление ассортиментом». Даная программа подойдёт: продуктовой и не продуктовой рознице, малому торговому бизнесу.

Облачный сервис с простым и понятным интерфейсом, в котором можно разобраться без бухгалтера или программиста. Приложение «Управление ассортиментом» идеально подходит для онлайн-касс и интегрируется с другими продуктами, «Маркировкой», решением для ЕГАИС и электронной книгой учёта доходов и расходов. Предприниматель может создать единую экосистему управления бизнесом.

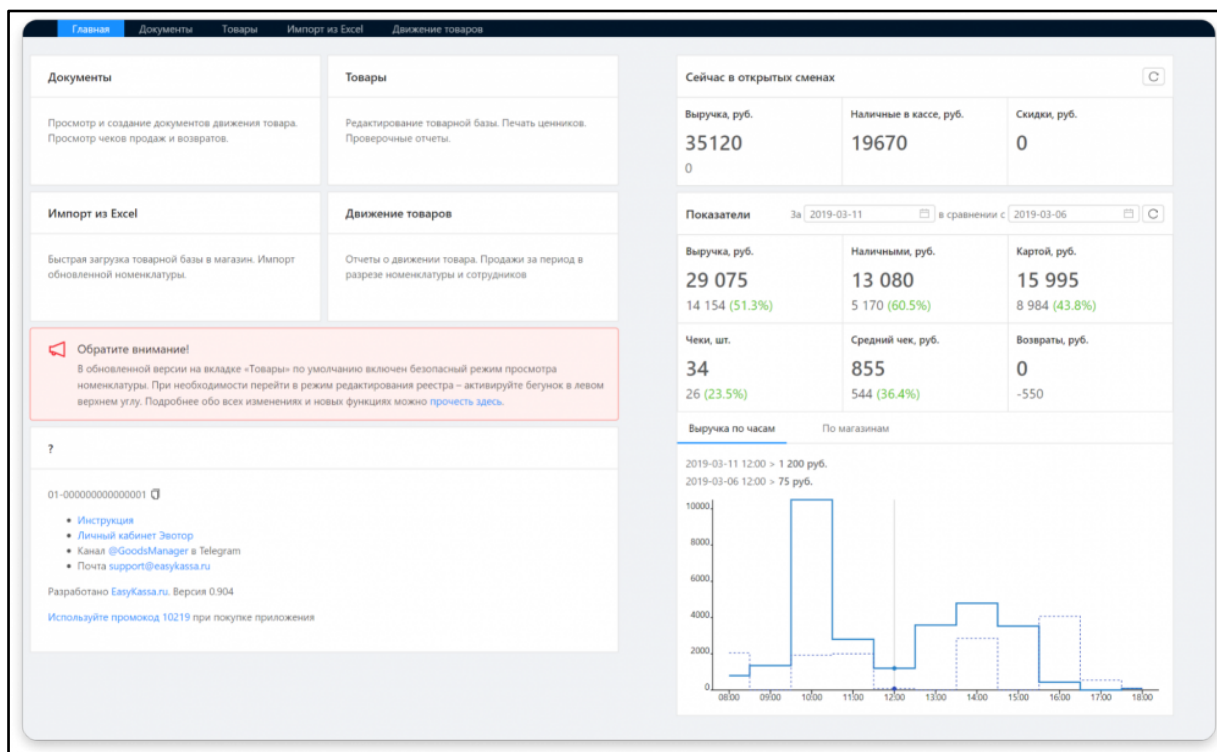


Рисунок 3 – Главная страница «Управление ассортиментом»

Программа учёта товара в розничной точке умеет создавать информативные отчёты о наличии товара, его количестве в магазинах и на складах, актуальном спросе на те или иные позиции. С ней легко понять — что стоит закупить впрок, а что не пользуется популярностью и занимает место на полках. ПО помогает выявить ошибки при учёте — найти отрицательные остатки и дубли, избежать пересортицы и просрочек. С помощью «Управления ассортиментом» можно быстро создавать карточки товаров и редактировать их — на компьютере или сразу в смарт-терминале. Все изменения автоматически синхронизируются с облачными данными. Владелец может заниматься управлением ассортиментом товаров

самостоятельно или выдать доступ сотрудникам.

Наименование	Код	Себестоимость	Цена	Остаток	Группа	Описание	Штрих-код
☐ Адрика кавказская	1	55.00	100.00	13	Услуги		2010061300214
☐ Адрика по-грузински	2	45.00	85.00	6	Услуги		2010009956626
☐ Азербайджанские помидоры	171	150.00	300.00	-2.6	Овощи		2010085629896
☐ Альпийские для сока	3	45.00	85.00	2	Фрукты		2010087977735
☐ Базилик	4	25.00	55.00	2	Овощи	Подмосковье	2010002992539
☐ Баклажаны	5	75.00	200.00	10	Овощи	Подмосковье	2010041808587
☐ Бананы	6	123.00	23.00	157	Фрукты		2010004711275
☐ Батон "Мента" в нарезке	12	27.51	32.00	100	Хлеб		4603797307770
☐ Батон нарезной классический	287	25.00	32.17	0	Хлеб		4607144171311
☐ Батон с изюмом к чаю, 400г	173	26.09	30.52	0	Сдоба		2000000000039
☐ Блины с малиновым джемом (3шт)	195	65.00	105.00	0	Пирожки	Хлебозавод №22	2010065150716
☐ Блины с мясом (3шт)	194	75.00	115.00	10	Пирожки	Хлебозавод №22	2010024724163
☐ Блины с творогом (3шт)	193	75.00	115.00	5	Пирожки	Хлебозавод №22	2010088281909
☐ Брокколи	7	123.00	250.00	2.7	Овощи		2010005653994
☐ Булка Ромашка "Приоскольская"	174	24.52	28.68	0	Сдоба		4603797307060
☐ Булочка с изюмом	8	25.00	30.00	20	Сдоба	Хлебозавод №22	2010062836194
☐ Булочка с маком	9	15.00	30.00	20	Сдоба	Хлебозавод №22	2010040405374
☐ Булочка с повидлом	10	25.00	30.00	11	Сдоба	Хлебозавод №22	2010002235544
☐ Ванилин	11	3.00	5.50	43	Услуги		2010044647060
☐ Ветчина	12	195.00	395.00	6.2	Услуги	Рублевский миксобиат	2010083090674
☐ Витрушка сдобная 0,350 кг	170	27.41	32.00	0	Сдоба		4603797307022
☐ Геркулес "Ясно солнышко"	13	25.00	45.00	19	Алкоголь		2010050027740
☐ Горох зеленый колотый Мистраль, 4...	14	65.00	91.00	8	Крупы		4607021399975
☐ Горошек консервированный	15	23.00	56.00	12	Услуги		2010035656903
☐ Доставка	16	150.00	300.00	1	Услуги		2010087008781
☐ Икра кабачковая традиционная	12900...	57.00	97.00	6	Бакалея		2010090480093
☐ Кабачки	17	20.00	50.00	19.5	Овощи	Подмосковье	2010059950865
☐ Капуста белокачанная	18	15.00	35.00	16.8	Овощи	Краснодарский край	2010039995909
☐ Капучино	2	0.00	50.00	0	Напитки		
☐ Картофель 2,5кг фасованный	19	30.00	122.00	213	Овощи	Израиль	2010023628668

Рисунок 4 — Главная страница «Управление ассортиментом»

В товароучётную систему можно загружать номенклатуру в формате Excel, после чего сервис её обработает — и нужные данные сразу появятся в смарт-терминале. Останется только сформировать и напечатать ценник — это тоже можно сделать в «Управлении ассортиментом» (рисунок 4). Ещё есть интеграция с 1С — программа учета товара умеет обмениваться данными с решением для учета финансов или ERP. Другие удобные приложения:

- ☒ модуль для работы с маркетплейсами,
- ☒ модуль для работы с франшизами,
- ☒ быстрая приёмка по ЭДО,
- ☒ модуль для печати ценников и штрихкодов.

1.4.3 SUBTOTAL

Приложение для онлайн-кассы в магазине розничной торговли или общепите. Позволяет контролировать сотрудников, продажи, наценку, товарные остатки. Автоматизирует действия:

- ☒ печать ценников — по шаблонам, со штрихкодом, любых размеров,
- ☒ контроль остатков по точкам продаж и складам,
- ☒ переоценка товаров по группам, с сохранением истории операций,
- ☒ документооборот — оприходования, списания, возвраты, перемещения товара и другие операции,
- ☒ инвентаризация по штрихкодам,
- ☒ ведение базы поставщиков — взаиморасчеты, история заказов, анализ по марже, ассортименту.

Есть тестовый период на 14 дней и пробный тариф. Работа программы проиллюстрирована на рисунке 5.

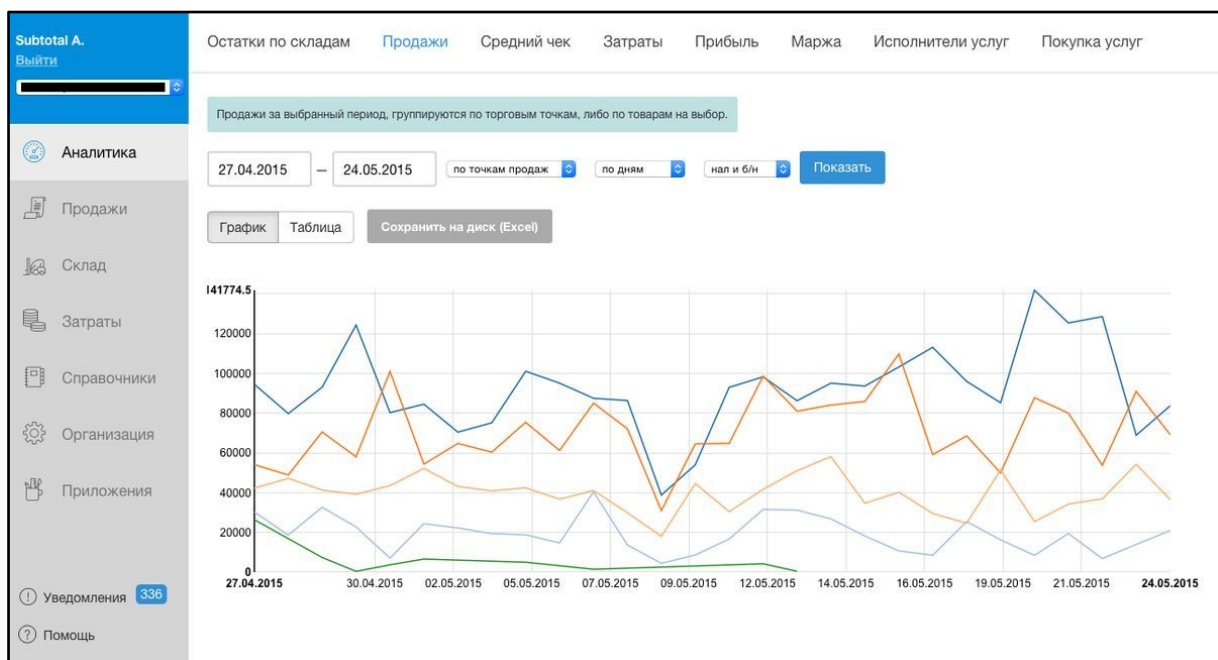


Рисунок 5 – Дашборд subtotal

1.4.4 МойСклад

Система управления торговлей объединяет продажи, закупки, склад, финансы, клиентов и поставщиков. Внутри есть собственная CRM, возможность подключить всё необходимое для работы, включая онлайн-кассу, фискальный регистратор, сканер и терминал (рисунок 6).

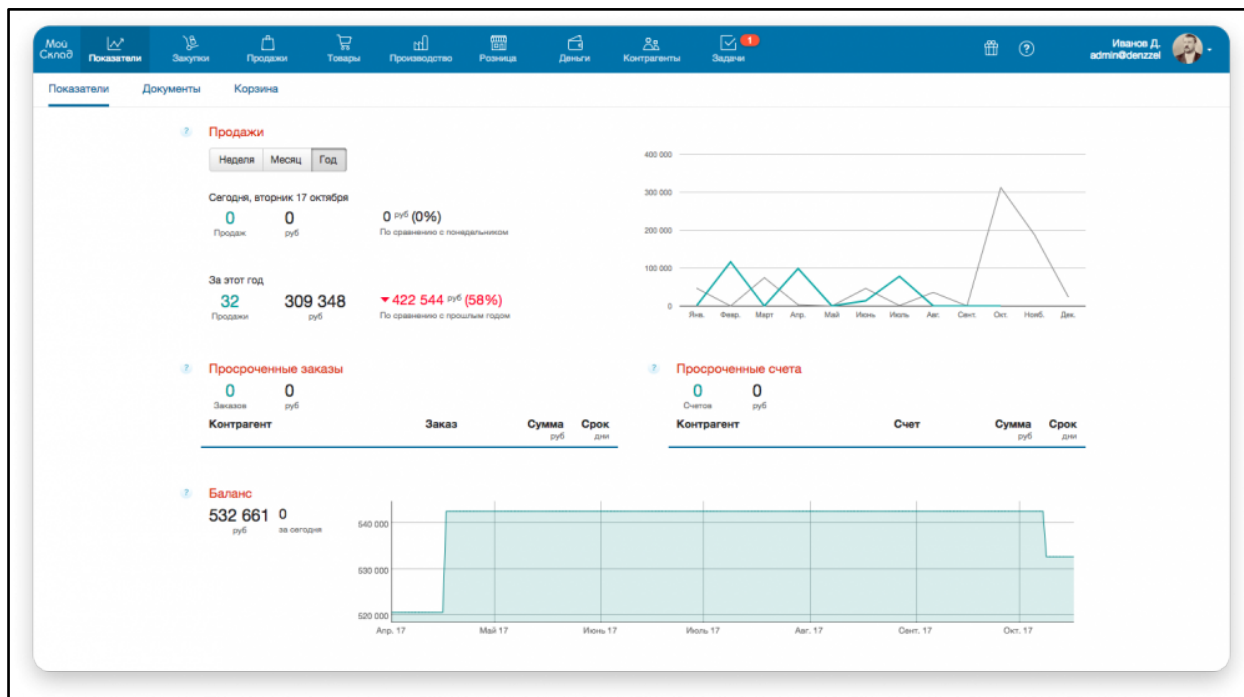


Рисунок 6 – Дашборд МойСклад

Программу учёта товаров можно подключить к системе ЕГАИС. Акт списания автоматически заполняется по ходу продажи бутылочных напитков за смену. Торговля маркированными товарами также поддерживается.

«МойСклад» интегрирован со многими сервисами, банками,

службами доставки и маркетплейсами, включая Яндекс.Маркет и Озон, что позволит эффективно торговать в онлайн. Есть продвинутый сервис аналитики клиентов, который поможет сделать маркетинговые кампании эффективнее. Приложение для учёта товаров и продаж «Мойсклад» также удобно для производств – потому что тут можно:

- ☒ оформлять операции сборки-разборки;
- ☒ планировать закупку материала;
- ☒ оформлять передвижение комплектующих к разным товарам.

Работа программы проиллюстрирована на рисунке 7.

№	Пров.	Время	Контрагент	Организация	Сумма	Валюта	Выставлено с...	Оплачено	Отгружено	Зарезервиро...	Статус	Изменен	Изменен	И -
79483	✓	17.10.2017 13:22	Новый клиент, истощ...	ООО "РОМАШКА"	0,00	руб	0,00	0,00	0,00	0,00	Новый	17.10.2017 17:12	Иванов Д.	
79482	✓	17.10.2017 13:21	Новый клиент, истощ...	ООО "РОМАШКА"	0,00	руб	0,00	0,00	0,00	0,00	Новый	17.10.2017 17:12	Иванов Д.	
79481	✓	17.10.2017 13:21	Новый клиент, истощ...	ООО "РОМАШКА"	0,00	руб	0,00	0,00	0,00	0,00	Новый	17.10.2017 17:12	Иванов Д.	
BB1	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	5 387,10	руб	0,00	0,00	0,00	0,00	Отгружен	17.10.2017 17:11	Иванов Д.	
BB2	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	6 156,20	руб	0,00	0,00	0,00	0,00	Собран	17.10.2017 17:11	Иванов Д.	
BB3	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	2 302,20	руб	0,00	2 302,20	2 302,20	0,00	Собран	17.10.2017 17:11	Иванов Д.	
BB4	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	1 899,00	руб	0,00	0,00	0,00	0,00	Отгружен	17.10.2017 17:12	Иванов Д.	
BB5	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	1 999,00	руб	0,00	1 999,00	1 999,00	0,00	Подтвержден	17.10.2017 17:11	Иванов Д.	
BB6	✓	06.02.2017 09:43	Иванов Петр	ООО "РОМАШКА"	1 500,00	руб	0,00	0,00	0,00	0,00	Возврат	17.10.2017 17:11	Иванов Д.	
сск_добр...	✓	17.01.2017 10:47	Иванов	ООО "РОМАШКА"	957,51	руб	0,00	0,00	0,00	0,00	Отменен	17.10.2017 17:11	Иванов Д.	
сск_добр...	✓	13.01.2017 14:03	Иванов	ООО "РОМАШКА"	50,00	руб	0,00	0,00	0,00	0,00	Доставлен	17.10.2017 17:11	Иванов Д.	
BB7	✓	22.11.2016 18:17	ООО "Покупатель"	ООО "РОМАШКА"	2 250,00	руб	0,00	0,00	0,00	0,00	Доставлен	17.10.2017 17:10	Иванов Д.	
00001	✓	15.04.2016 12:05	Розничный покупатель	ООО "РОМАШКА"	11 111,00	руб	0,00	0,00	0,00	0,00	Доставлен	17.10.2017 17:10	Иванов Д.	
1-13 из 13					33 612,01		0,00	4 301,20	4 301,20	0,00				

Рисунок 7 – модуль «Продажи»

1.4.5 LiteBox

Это российская платформа, которая, как считают эксперты, может составить достойную конкуренцию зарубежным аналогам

(рисунок 8).

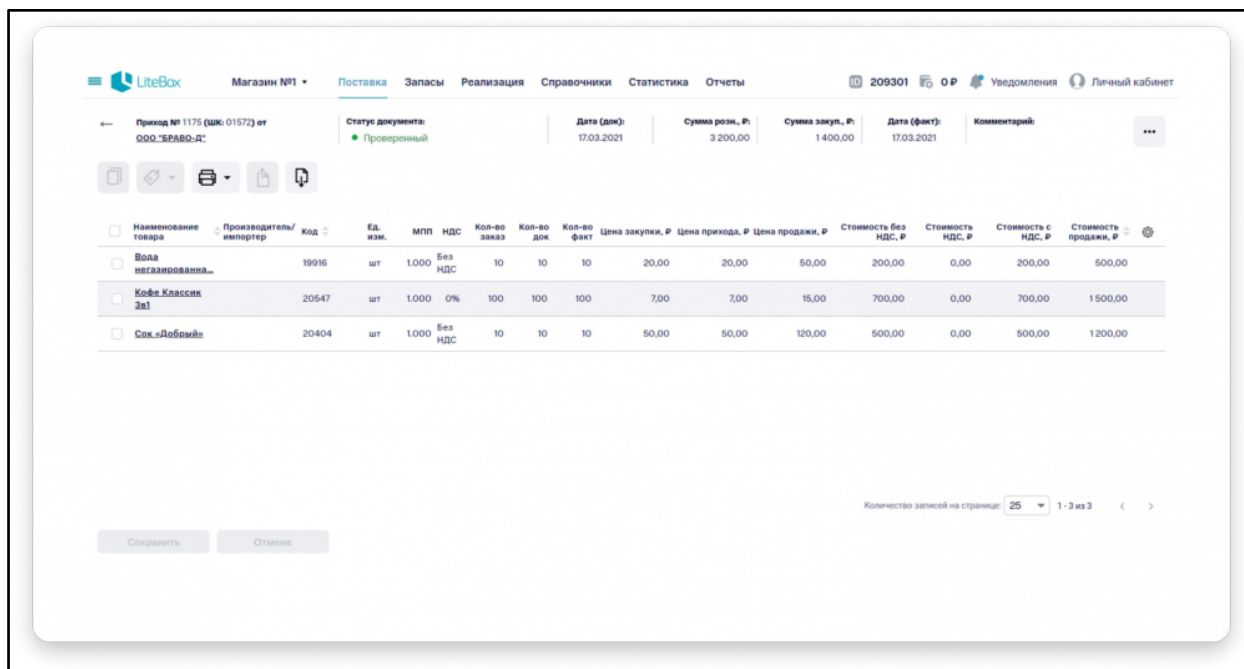


Рисунок 8 – Дашборд LiteBox

Одна из популярных и простых программ для учёта товаров. Сервис позволяет работать с любого устройства, включая планшеты и смартфоны — главное, чтобы был интернет. Вместе с сервисом можно использовать периферию:

- ☒ онлайн-кассы;
- ☒ сканеры штрихкодов;
- ☒ принтеры чеков и этикеток;
- ☒ весы;
- ☒ денежные ящики и т.д.

С «Лайтбоксом» доступны все необходимые операции по товарообороту: от приёмки до списания и возвратов. В том числе можно работать с подакцизными и маркированными товарами.

Помимо этого, программа для ведения учёта товаров в магазине умеет создавать аналитику по продажам с 40 видами отчетов. ПО позволяет оформлять перемещение продукции со склада на склад, накладные, счета-фактуры и другие документы.

Работа программы проиллюстрирована на рисунке 9.

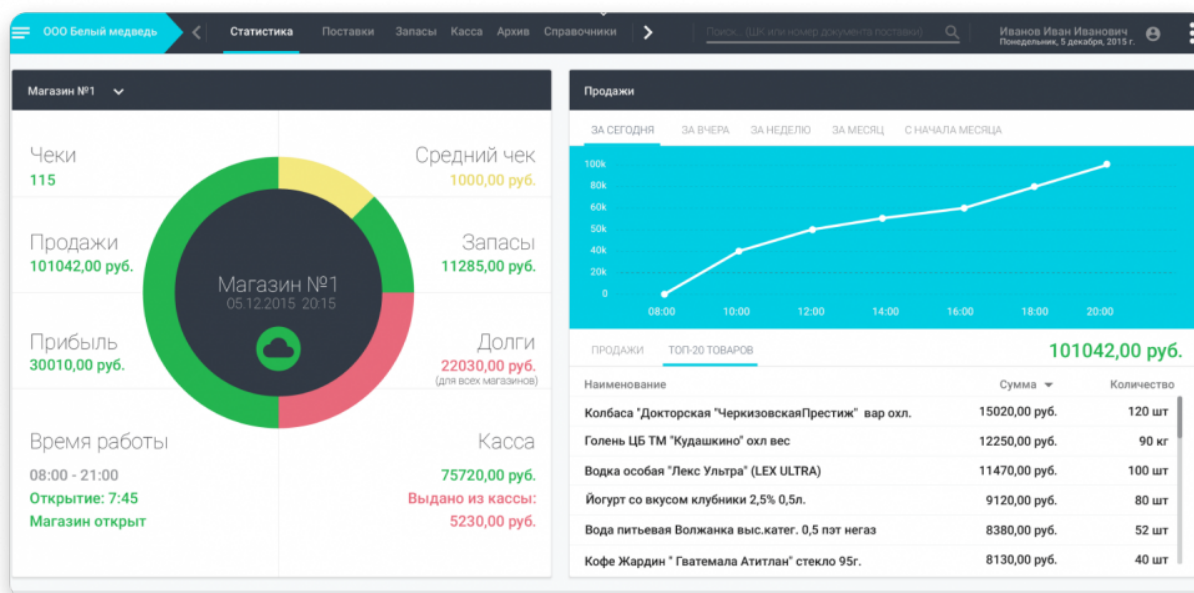


Рисунок 9 — Статистика по показателям

1.4.6 Антисклад

Достаточно простая и понятная программа для розничной торговли (рисунок 10). В ней есть всё необходимое для учёта и продаж любых товаров, включая маркированные. Сервис заточен под работу с размерными товарами: удобно создавать сетки с цветами и размерами, редактировать их под разные виды одежды.

АНТИСКЛАД
программа учета

Настройки ПомощьАдминистраторВыход

Главная

Склады

Заказы

Магазин

Анализ

Архив операций

Справочники

ФИЛЬТР ПО СКЛАДАМ

- все склады -

ФИЛЬТР ПО ТОВАРУ

Категория:

Торговая марка:

☐ С учетом резервов

ФИЛЬТР ПО ПОСТАВЩИКАМ

Период закупок:

Весь период

ВСЕ СКЛАДЫ - 58

ОстатокПоступленияРезервыЭтикеткиДокументыЕще

	Фото	Название	Категория	Марка	Кол-во	Склады		Закупочная			Наценка		Отпускная			
						Главный склад	Склад магазина	Цена	Курс	Цена Р	Сумма	Наценка	%	Марка	Цена	Сумма
1.		Intel Набор картidgeй для фильгра 2 шт. Z9902	Бассейны		11	1	10	230.00 Р	1.00	230.00	2 530.00	250.00	109%	2 750.00	480.00	5 280.00
2.		MadWave Секундомер Stopwatch 500 memory	Секундомер		12		12	1 490.00 Р	1.00	1 490.00	17 880.00	1 490.00	100%	17 880.00	2 980.00	35 760.00
3.		Апперкотная подушка DFC TR1 кожа	Бокс	DFC	5		5	9 048.00 Р	1.00	9 048.00	45 240.00	10 012.00	111%	50 060.00	19 060.00	95 300.00
4.		BATUT DFC TRAMPOLINE FITNESS 32INCH TR 32 двойка б/сетка (81cm)	Гимнастические снаряды	DFC	3	1	2	1 120.00 Р	1.00	1 120.00	3 360.00	1 270.00	113%	3 810.00	2 390.00	7 170.00
5.		Баскетбольная мобильная стойка DFC STAND4PVC3 110x75cm ПВХ раздвиж.регулировка	Баскетбол	DFC	5		5	6 980.00 Р	1.00	6 980.00	34 900.00	8 010.00	115%	40 050.00	14 990.00	74 950.00
6.		Бассейн каркасный BestWay 56088 366*122 см	Бассейны		3		3	9 560.00 Р	1.00	9 560.00	28 680.00	11 560.00	121%	34 680.00	21 120.00	63 360.00
7.		Бассейн каркасный BestWay Steel Pro Frame 56415/56030	Бассейны		4	2	2	3 780.00 Р	1.00	3 780.00	15 120.00	2 570.00	68%	10 280.00	6 350.00	25 400.00

Рисунок 10 – Дашборд Антисклад

Антисклад – хорошая программа для учёта товара, но в ней нельзя работать с сыпучими и жидкими товарами на развес и розлив – она пригодна только для поштучной продажи. Антисклад интегрируется с Эвотором, InSales и Мегапланом. Также можно автоматически выгружать товары сразу на Яндекс.Маркет

Работа программы проиллюстрирована на рисунке 11.

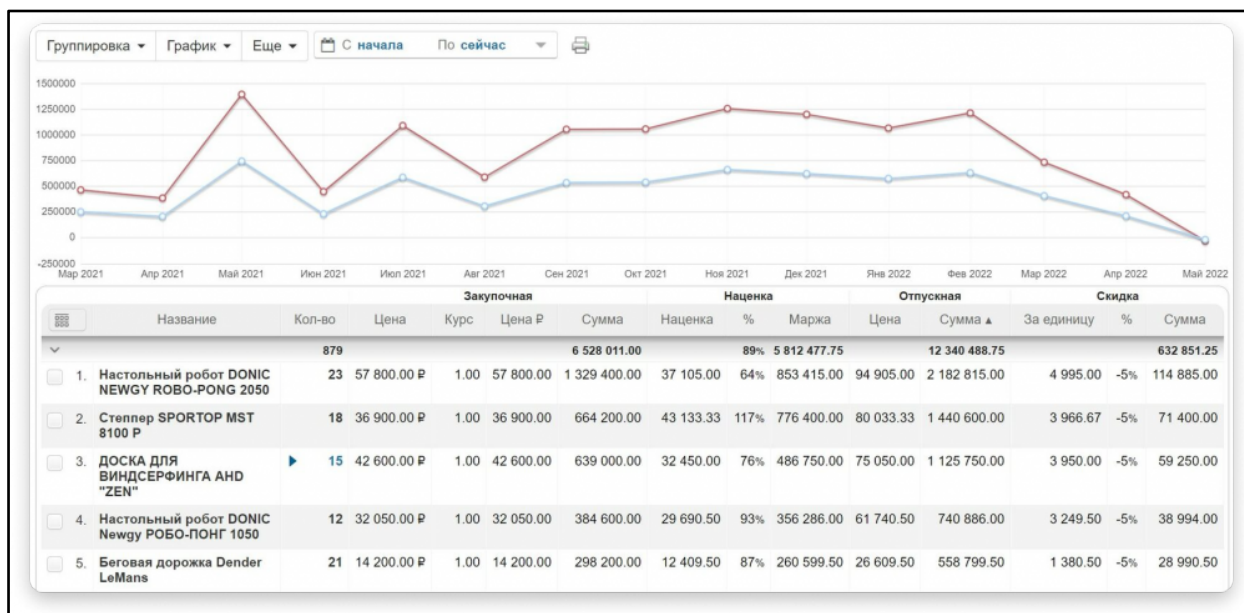


Рисунок 11 — Аналитика в программе

1.4.7 Вывод

Для эффективной подачи информации при помощи графиков и таблиц, необходимо обратить внимание на несколько важных аспектов: подбор правильных типов графиков и диаграмм, а также создание интуитивно понятного интерфейса играют ключевую роль в улучшении восприятия информации.

Поэтому, при разработке приложения, важно уделить внимание не только качеству визуализации данных, но и удобству интерфейса для пользователей, чтобы обеспечить оптимальный опыт работы и понимание сути данных без необходимости глубокого вникания в цифры.

В разрабатываемом приложении информация будет предоставляться в виде графиков и таблиц.

1.5 Анализ баз данных

У современного человека часто возникает потребность в доступе к большому объему информации. Иногда, для поиска нужных данных требуется большое количество времени. Анализировать обширные объемы информации нелегко, и всегда существует риск упустить важные детали [16].

База данных (БД) — это набор информации, которая хранится упорядоченно в электронном виде.

Базы данных позволяют обрабатывать, хранить и структурировать намного большие объёмы информации, чем таблицы.

Удалённый доступ и система запросов позволяет множеству людей одновременно использовать базы данных. С электронными таблицами тоже можно работать онлайн всей командой, но системы управления базами данных делают этот процесс организованнее, быстрее и безопаснее.

Объём информации в базах данных может быть огромным и не влиять на скорость работы. А в Google Таблицах уже после нескольких сотен строк или тысяч символов страница будет загружаться очень медленно.

Отличия таблиц от баз данных основаны на главном свойстве последних – информацию можно связать, даже если она находится в разных таблицах.

Управление базами данных выполняется с помощью языка запросов SQL.

Удобное использование баз данных основано на их свойствах:

Быстродействие. Современные БД проектируются по принципу «получить данные прямо сейчас», чтобы пользователь не ждал отклик на запрос.

Простота получения и обновления данных. Какой бы высокой ни была скорость, это бессмысленно, если нужно сделать много

сложных операций, чтобы получить, обновить или добавить данные в базу.

Независимость структуры. Изменения в любом количестве и качестве информации не должны влиять на структуру базы данных. Также изменения не должны касаться программного обеспечения и средств хранения, например жёсткого диска.

Стандартизация. Аналогично свойству независимости структуры: при обновлении программного обеспечения или СУБД (сокр. от «системы управления базами данных») база данных не должна менять свою структуру или свойства.

Безопасность данных. Для каждой категории пользователей делают список ограничений и доступов, согласно которым можно взаимодействовать с информацией из БД.

Интегрированность. Данные должны быть логически связаны. И эти связи должны прослеживаться по структуре таблиц.

Многопользовательский доступ. Удалённо вносить изменения и получать информацию из БД могут сразу несколько человек с разных устройств.

Чаще всего базы данных классифицируют в зависимости от того, как в них структурирована информация и как с ней взаимодействовать.

Иерархические. Простейшая структура, где записи, как ветви, отходят от «родителя». Информация связана по аналогии с папками на рабочем столе. У каждой записи есть физическая связь только с одной предыдущей, а отношение многих ко многим невозможно.

Сетевые. В отличие от иерархической структуры, у каждой записи может быть более одного родителя. Сетевые БД представляют собой не древовидный, а общий граф.

Объектно-ориентированные. Базы данных, где информация о реальных вещах представлена в виде объектов под уникальным

идентификатором. К ООБД обращаются на языке объектно-ориентированного программирования (ООП). Состояние объекта описывается атрибутами, а его поведение — набором методов. Объекты с одинаковыми атрибутами и методами образуют классы.

Объект в ООП создаётся как отдельная сущность со своими свойствами и методами работы. И как только объект создан, его можно вызвать по «имени», или коду, а не разрабатывать заново.

Например, при разработке руки главного героя в компьютерной игре программист один раз создаёт объект «рука». И когда нужно запрограммировать движения, достаточно сказать на языке программирования: «рука, сожми кулак» или «рука, поднимись вверх».

По такому же принципу работают и объектно-ориентированные базы данных. Данные формируются в объекты, и работать с ними можно как с рукой героя в примере.

Реляционные. Их также называют SQL — как язык программирования, с помощью которого создают, преобразовывают и управляют данными в реляционных БД. Записи и связи между ними организованы при помощи таблиц. В таблицах есть поле для внешнего ключа со ссылками на другие таблицы. Благодаря высокой организации и гибкости структуры реляционные БД применяются для многих типов данных.

Нереляционные (NoSQL). Эту группу называют также NoSQL, потому что к таким базам данных нужны отличные от SQL-запросы.

Базы данных «ключ-значение». В таких базах данные сохраняются под ключами. Если хотите получить объект, например, изображение или текст, нужно ввести ключ. Таким образом часто хранят информацию о состоянии объектов, представленную различными типами данных. Каждому хранилищу разрабатывают свою схему именования ключей в зависимости от форматов значений.

Графовые базы данных. Связи в графовых базах данных обозначены узлами, рёбрами и свойствами. Записи в этих БД могут иметь любое количество связанных с ними свойств.

Колоночные базы данных. Подобно реляционным, в этих базах данные хранятся в виде таблиц. Но структура колонок строго не регламентирована — они могут объединяться в колоночные семейства с определённым форматом. Строки колоночного семейства имеют уникальные идентификаторы.

NewSQL базы данных совмещают реляционную модель, язык запросов SQL и распределённые горизонтально масштабируемые базы данных NoSQL. Этот тип появился на рубеже 2000-х и 2010-х годов как ответ на потребности рынка, которые существующие БД не могли удовлетворить. Так, SQL-базы не поддерживали масштабируемость на уровне NoSQL, а эти, в свою очередь, не отвечали стандартам точного выполнения оперативных транзакций ACID (англ. atomicity, consistency, isolation, durability — «атомарность, непротиворечивость, изолированность, долговечность»).

Для баз данных NewSQL характерны:

- реляционная модель и транзакционность,
- язык SQL для доступа к данным,
- горизонтальная масштабируемость,
- более быстрая производительность за счёт новых «движков».

Развитие типов баз данных не останавливается на NewSQL. Появляются новые гибридные модели (англ. multi-model databases), так как использование базы данных может решить разные задачи.

Популярные системы управления базами данных

Системы управления базами данных (СУБД) — это инструменты, с помощью которых пользователь обращается к данным, изменяет их или создаёт. СУБД функционируют как менеджеры по работе с базами данных, которые «говорят» на их языке программирования. В

российских и зарубежных компаниях используют шесть популярных СУБД:

Oracle -объектно-реляционная СУБД, созданная одноимённой компанией-лидером на рынке. Преимущества Oracle: быстрая установка и настройка, возможность расширять функционал, практичность и надёжность. Но лицензия стоит дорого, поэтому Oracle обычно используют крупные корпорации.

MySQL - реляционная СУБД с открытым исходным кодом, то есть доступна для просмотра, исправления ошибок и создания новых версий программ. MySQL — бесплатная, быстрая и гибкая система, подходящая для таблиц разных типов.

Microsoft SQL Server оптимальная СУБД для операционных систем Windows, но совместима и с Linux. Легко интегрируется с другими продуктами Microsoft, удобна в использовании, но потребляет много ресурсов, а лицензия стоит дорого.

PostgreSQL - объектно-реляционная СУБД, которую используют для сайтов, сервисов и платформ. Бесплатный доступ и поддержка многих языков программирования делают эту СУБД одной из самых популярных. По её лицензии создано немало расширенных версий, в том числе для коммерческого использования.

Apache Cassandra в отличие от вышеназванных, Cassandra — не реляционная СУБД. Она разработана на языке Java и принадлежит фонду Apache Software Foundation. Система хранит данные по модели семейства столбцов и «ключ-значение», распределяет данные в несколько дата-центров и легко масштабируется при увеличении объёма информации.

Redis - NoSQL резидентная СУБД, которая использует модель «ключ-значение». Она написана на языках C и C++, а применяется для атомарных операций, например, записи и чтения быстро изменяющихся данных.

Управлять базами данных с помощью языка запросов SQL и различных СУБД можно научиться на курсе «SQL для работы с данными и аналитики». Этот навык пригодится в работе аналитикам, маркетологам, продакт-менеджерам и экономистам. Даже без опыта в IT студенты за период от 1,5 месяцев осваивают современные инструменты для работы с базами данных.

Вывод о базах данных.

Базы данных (БД) — это структурированная информация, которая хранится в связанных электронных таблицах. Базы можно встретить в банках, библиотеках, космических наблюдениях и вообще везде, где данных слишком много, чтобы обрабатывать их вручную.

Не каждая электронная таблица — это база данных. В базе между таблицами есть связи, которые организованы в строгой системе. Отсюда и свойства баз данных: скорость обработки, простота получения и независимость данных.

По структуре и способу связей основные базы данных делятся на типы: иерархические, сетевые, колоночные, реляционные, нереляционные и объектно-ориентированные.

Чтобы управлять данными в базе, используют СУБД (систему управления базами данных). Каждому типу баз подходят свои СУБД.

1.6 Выводы

Итак, в рамках этого раздела были достигнуты следующие результаты: произведен исчерпывающий анализ предметной области, в рамках которой планируется вести разработку приложения, было рассмотрено машинное обучение и нейронные сети, проанализированы аналоги разрабатываемого приложения, а так же проанализированы методы парсинга веб сайтов.

2 Обоснование использования заданных средств разработки

2.1 Анализ ОС для функционирования приложения

Для функционирования интеллектуальной системы для обработки, хранения и визуализации информации необходимо выбрать операционную систему (ОС).

В соответствии с ГОСТ 15971-90, ОС – это совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг [22].

При выборе ОС необходимо обратить внимание на такие особенности, как совместимость с различными ПО, надежность, простота установки, обслуживание и стоимость приобретения.

Самыми распространенными ОС на сегодняшний день являются Windows, Mac OS и Linux.

Windows является самой распространенной ОС. Она установлена на большую часть устройств, будь то планшеты, ноутбуки или компьютеры. Windows используется как дома, так и на предприятии. Основными достоинствами ОС Windows является отличная совместимость с различными программными средствами и широкая распространенность.

Однако с распространением мобильной ОС Android, Linux стала теснить

Windows, т. к. стала основой для Android. Linux принадлежит к семейству открытых и свободных систем семейства Unix. Основное достоинство данной ОС – это возможность доработки недочетов и неточностей благодаря открытому исходному коду.

Mac OS (Macintosh Operating System) – семейство ОС производства корпорации Apple. Эта система разработана для

линейки персональных компьютеров (ПК) Macintosh. Одной из последних версий данной ОС является

Mac OS X, которая в недавнем времени была переименована в macOS. Преимуществами MacOS являются стабильность и производительность.

По данным Яндекс.Метрики 92,58% пользователей ПК используют ОС Windows. Соотношение ОС у пользователей ПК представлено на рисунке 12.

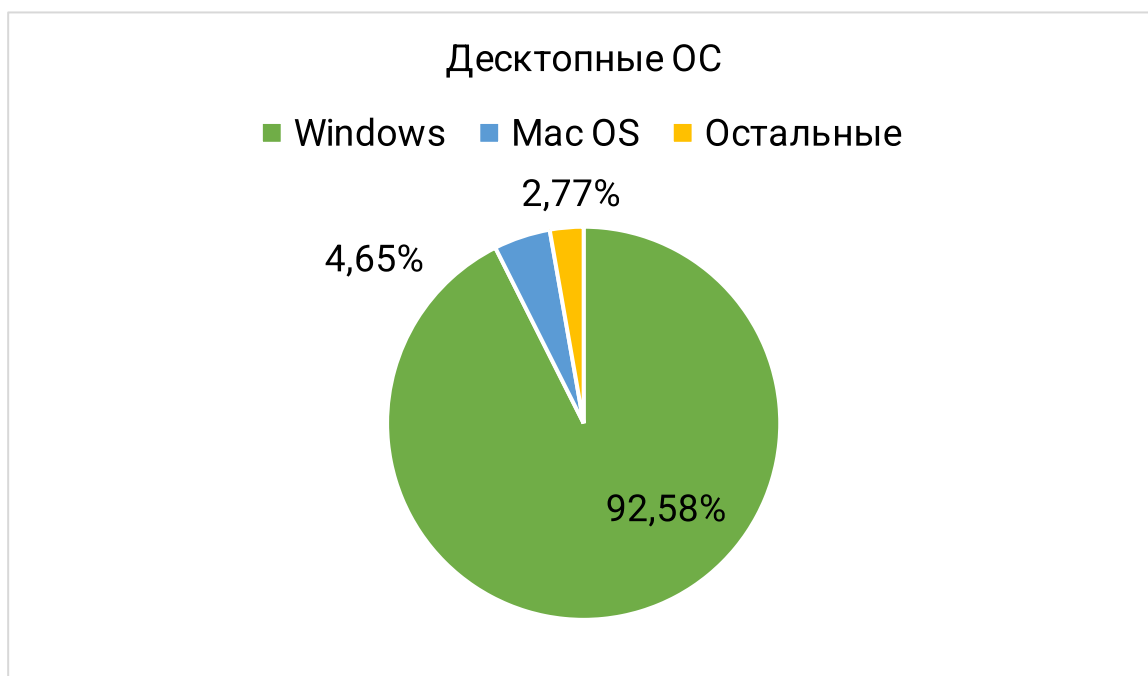


Рисунок 12 — Соотношение десктопных ОС

Целесообразность использования данной ОС подтверждается следующим рядом достоинств:

- ☒ возможность установки на многие компьютеры,
- ☒ высокий уровень надежности,
- ☒ возможность резервного копирования важных файлов, что

обеспечивает сохранность данных,

- ☒ возможность одновременной работы нескольких приложений, не теряя при этом производительности,

- ☒ совместимость с наиболее распространенными программами,

- ☒ и многие другие.

Таким образом, для работы программы для сбора, анализа, хранения и визуализации данных была выбрана ОС Windows.

2.2 Анализ платформ для разработки приложения

Для разработки программного обеспечения используются разнообразные инструменты. В их спектр включены компоновщики, ассемблеры, компиляторы, а также отладчики, приложения для анализа производительности, дизайнеры графических интерфейсов. Грамотно подобранные инструменты значительно повышают производительность, а также помогают поддерживать рабочий процесс имеющегося проекта.

Платформы для разработки приложений предоставляют разработчикам единую среду, в которой есть все инструменты, необходимые для создания программных приложений. Некоторые платформы обрабатывают множество различных типов приложений, в то время как другие ориентированы на конкретный вариант использования.

Компании, нуждающиеся в программном приложении, предоставляют своим разработчикам подобные платформы для написания, управления и развертывания приложений. Обычно этот процесс происходит полностью в отделе разработки или IT, но может происходить в рамках небольшой компании или на индивидуальной основе.

Платформы для разработки приложений обладают характеристиками, аналогичными интегрированным средам разработки (IDE), но они, как правило, более надежны в

предоставлении разработчикам полного набора инструментов разработки.

Чтобы претендовать на включение в категорию платформ для разработки приложений, продукт должен:

- ▣ предоставлять возможность разработки программного обеспечения с помощью текстового редактора или графического пользовательского интерфейса (GUI),

- ▣ предоставлять интерфейс прикладного программирования (API) платформы и позволять выполнять компиляцию, отладку, контроль версий, предложения кода для конкретной платформы или развертывание кода,

- ▣ обладать интерфейсными инструментами разработки для создания пользовательского интерфейса / UX, внутренними инструментами разработки для служб передачи данных,

- ▣ предоставлять набор разработчика программного обеспечения с дополнительными инструментами для таких действий, как совместная работа, отладка или анализ кода.

Рассмотрим существующие платформы для разработки, выясним преимущества и недостатки каждой из них.

2.2.1 QT Creator

Qt Creator – это кроссплатформенная интегрированная среда разработки C ++, JavaScript и QML, входящая в состав SDK для Qt Framework [23]. Внешний вид представлен на рисунке 13.

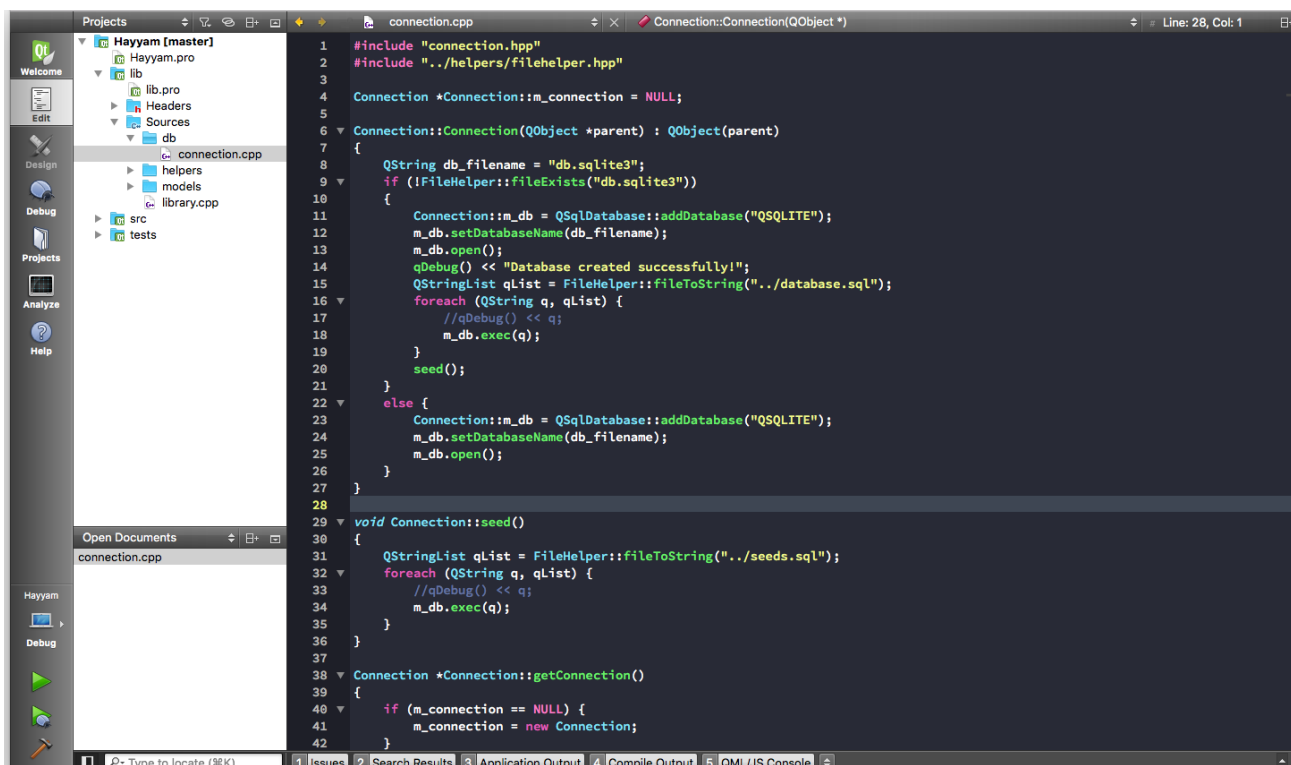


Рисунок 13 — Среда программирования Qt Creator

Он включает в себя визуальный отладчик, встроенный графический интерфейс и конструктор форм. Среди функций редактора - подсветка синтаксиса и автозаполнение.

Qt Creator использует компилятор C++ из коллекции компиляторов GNU для Linux и FreeBSD. Под Windows может работать с MinGW или MSVC с установкой по умолчанию, а также поддерживает отладчик консоли Microsoft при компиляции из исходного кода. Также есть поддержка Clang.

Qt Creator включает в себя менеджер проектов, который может использовать различные форматы проектов, такие как .pro, CMake, Autotools и другие. Файл проекта содержит информацию о

включенных файлах, пользовательских этапах сборки и настройках для запуска приложений.

Кроме того, Qt Creator включает редактор кода и интегрирует Qt Designer для проектирования и создания графических пользовательских интерфейсов (GUI) из виджетов Qt.

2.2.2 Eclipse

Eclipse представляет собой расширяемую платформу разработки со средами исполнения и прикладными платформами для создания, развертывания и управления программным обеспечением на всех этапах его жизненного цикла.

Внешний вид представлен на рисунке 14.

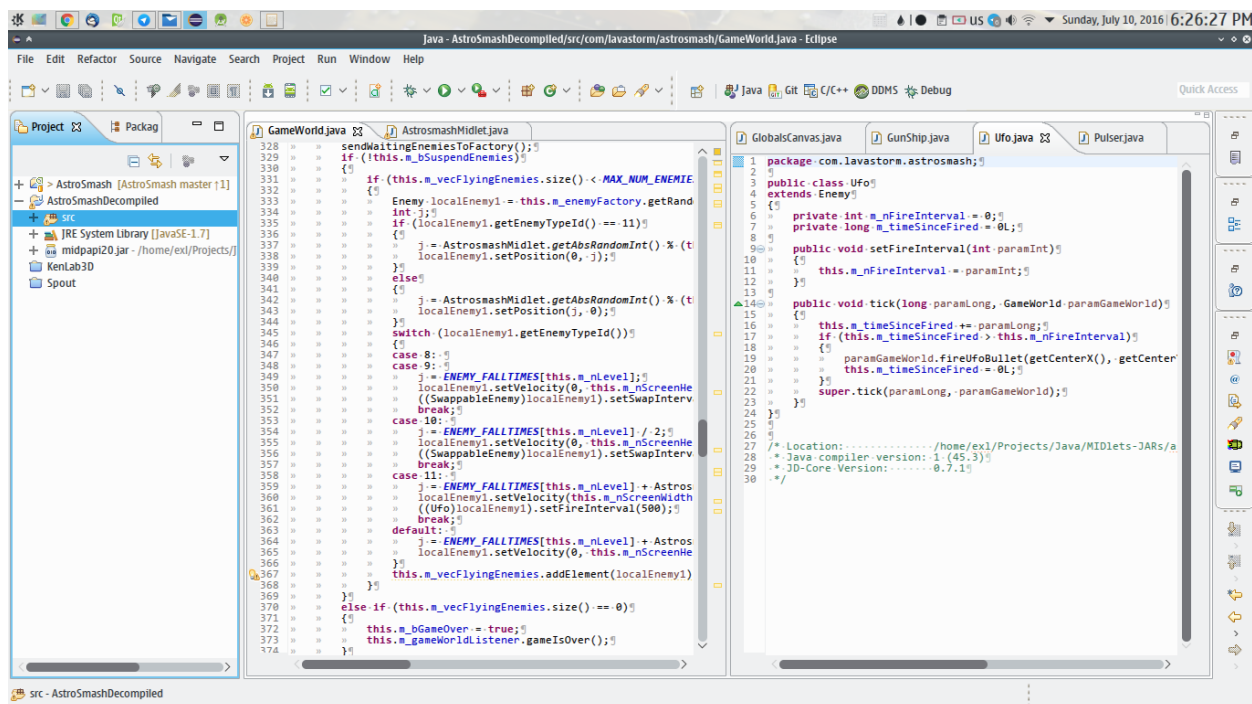


Рисунок 14 – Среда программирования Eclipse

Eclipse тесно связан с экосистемой Java, часто сопровождая

новые версии Java обновлениями своей платформы. Переход к регулярным квартальным обновлениям от ежегодных выпусков отражает стремление к более частым итерациям разработки. Одной из уникальных особенностей Eclipse является его модульность. Он представляет собой практически полностью модульную систему, состоящую из различных плагинов.

Некоторые плагины работают на уровне ядра и отвечают за базовые функции программы, такие как интерфейс, файловая система, система контроля версий и другие. Другие плагины могут быть добавлены к системе с помощью специальных точек расширения, позволяя разработчикам добавить функциональность, необходимую для их работы.

2.2.3 Microsoft Visual Studio

Microsoft Visual Studio — это интегрированная среда разработки (IDE) Она предназначена для создания компьютерных программ для операционной системы Microsoft Windows, а также веб-сайтов, веб-приложений и веб-сервисов. Visual Studio использует платформы разработки программного обеспечения от Microsoft, включая Windows API, Windows Forms, Windows Presentation Foundation, Windows Store и Microsoft Silverlight. Среда поддерживает как нативный код, так и управляемый. Внешний вид представлен на рисунке 15.

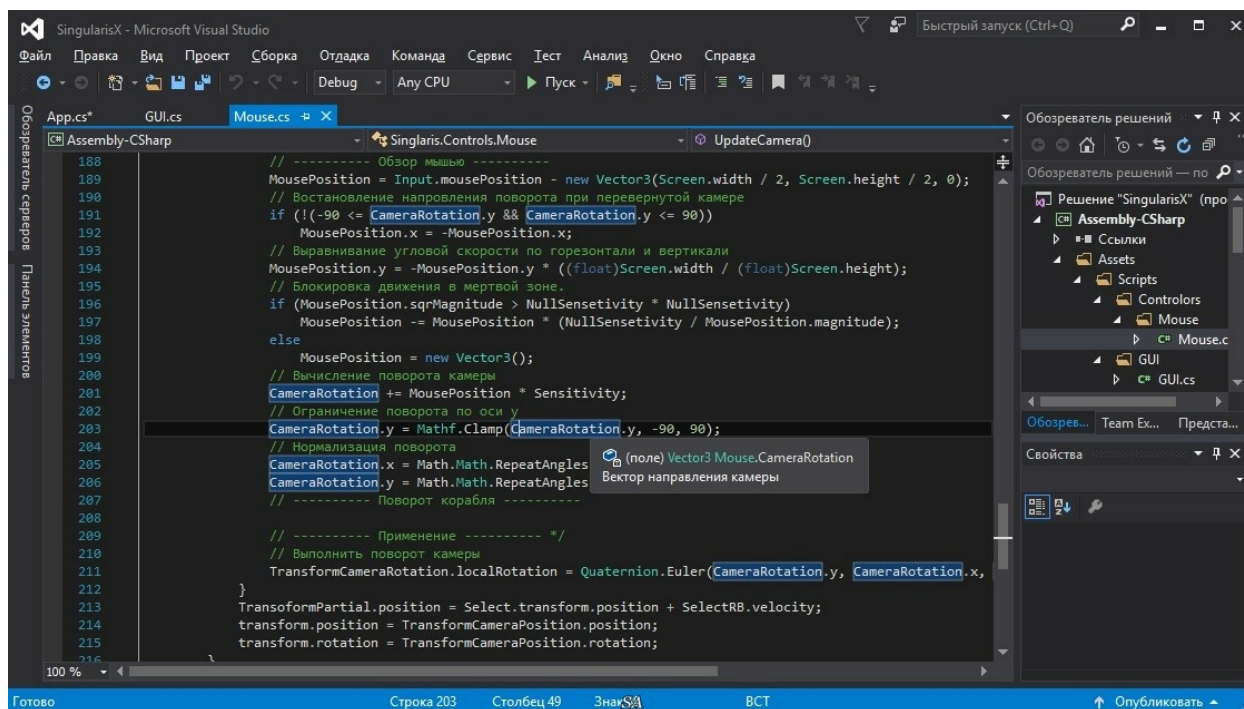


Рисунок 15 — Среда программирования Microsoft Visual Studio

Visual Studio включает в себя редактор кода, поддерживающий IntelliSense, а также рефакторинг кода. Интегрированный отладчик работает как отладчик уровня источника и отладчик уровня машины. Другие встроенные инструменты включают конструктор форм для создания приложений с графическим интерфейсом, веб-дизайнер, дизайнер классов и конструктор схем базы данных. Он принимает плагины, которые расширяют функциональные возможности практически на каждом уровне, включая добавление поддержки систем управления исходным кодом и добавление новых наборов инструментов, таких как редакторы и визуальные дизайнеры для языков, специфичных для предметной области, или наборов инструментов для других аспектов жизненного цикла разработки

программного обеспечения.

Visual Studio поддерживает разные языки программирования и позволяет редактору кода и отладчику поддерживать практически любой язык программирования при условии, что существует языковая служба. Встроенные языки включают C, C ++ и C ++ / CLI (через Visual C ++), VB.NET (через Visual Basic .NET), C # (через Visual C #) и F # (начиная с Visual Studio 2010). Поддержка других языков, таких как M, Python и Ruby и др., доступна через языковые службы, устанавливаемые отдельно. Он также поддерживает XML / XSLT, HTML / XHTML, JavaScript и CSS.

2.2.4 IntelliJ IDEA

IntelliJ IDEA — это бесплатная коммерческая среда Java IDE, разработанная компанией JetBrains. Его дизайн ориентирован на производительность программиста. Она ориентирована на повышение производительности программиста, предлагая ряд функций, направленных на ускорение разработки, в то время как IntelliJ IDEA выполняет рутинные задачи кодирования.

IntelliJ IDEA доступен в двух редакциях: Ultimate (платная) и Community (бесплатная и с открытым исходным кодом).

IDEA поддерживает различные языки программирования с расширенными функциями проверки кода и рефакторинга, такие как Groovy, Javascript, Actionscript, Ruby / JRuby, PHP и другие.

Версия Community Edition является открытым исходным кодом, но пропускает большинство функций для серверной и веб-разработки.

Внешний вид представлен на рисунке 16.

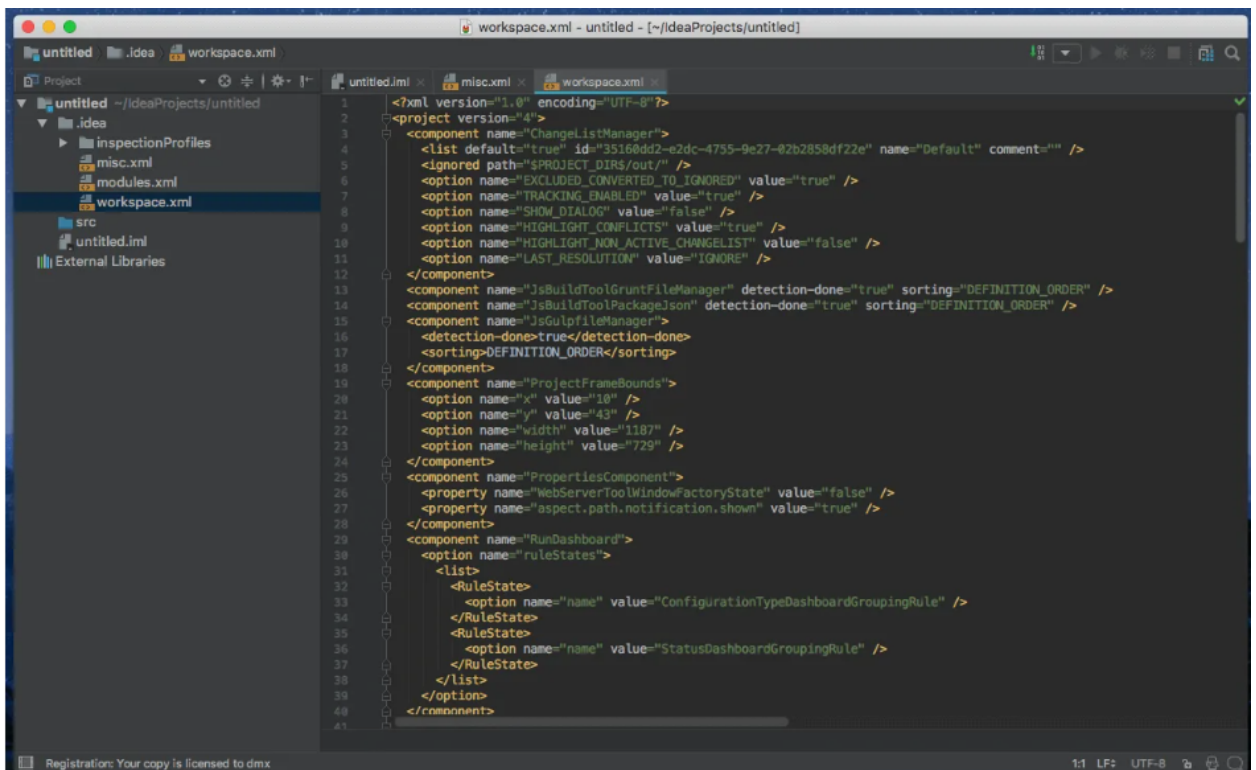


Рисунок 16 — Среда программирования IntelliJ IDEA

2.2.5 NetBeans

NetBeans IDE - это бесплатная интегрированная среда разработки с открытым исходным кодом, созданная для разработчиков программного обеспечения. Она обеспечивает все необходимые инструменты для создания профессиональных приложений: от настольных и корпоративных до веб-приложений и мобильных приложений на различных языках, таких как Java, C/C++, а также динамические языки программирования, включая PHP, JavaScript, Groovy и Ruby.

Среда IDE NetBeans проста в установке и использовании прямо из коробки и работает на многих платформах, включая Windows, Linux,

Mac OS X и Solaris.

В разделе загрузки удобно представлены предварительно упакованные пакеты/установщики для каждого поддерживаемого языка программирования, содержащие только необходимые плагины.

Внешний вид представлен на рисунке 17.

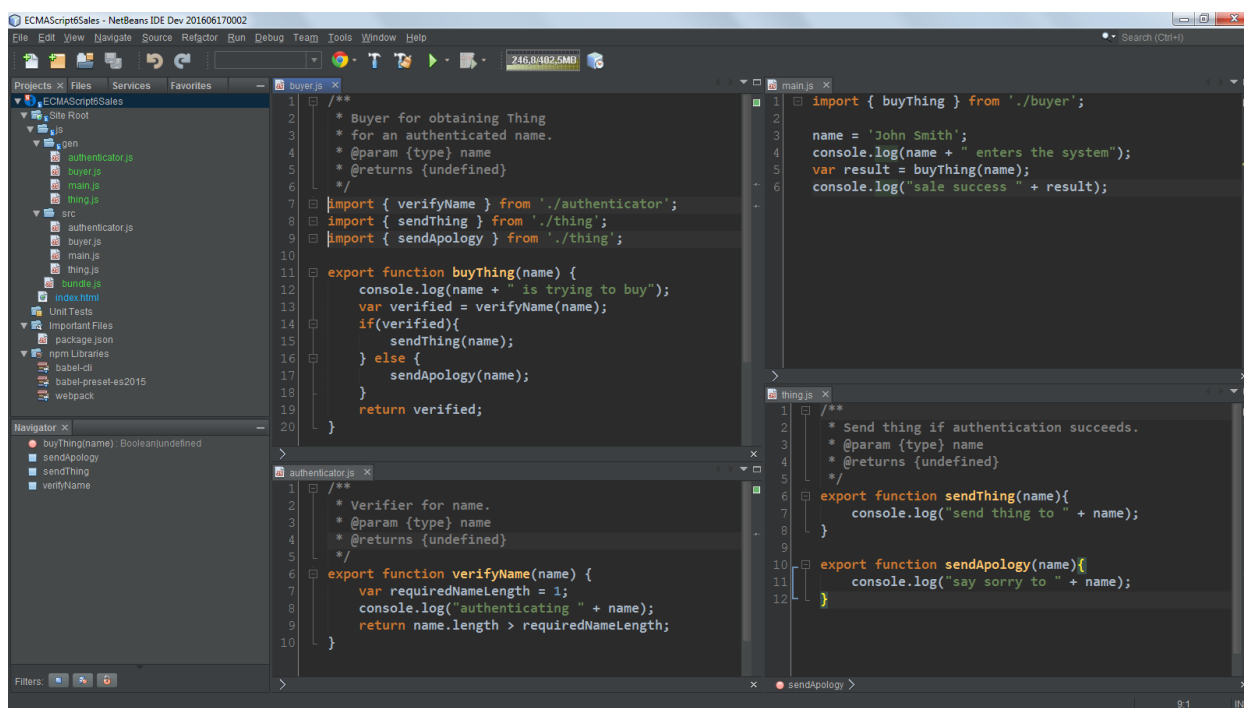


Рисунок 17 — Среда программирования NetBeans

В качестве среды разработки выбрана Qt Creator.

2.3 Вывод

В рамках этого раздела были достигнуты следующие результаты: произведён анализ ОС для функционирования приложения, а также анализ платформ для разработки приложения.

3 Техническое задание

3.1 Назначение разработки

Программа должна быть предназначена для обработки, хранения и передачи данных о заказах пользователя товара, представленного в книжном интернет-магазине.

3.2 Требования к программе

Разрабатываемая интеллектуальная система должна представлять собой программное обеспечение, которое способно осуществлять взаимодействие пользователя с интерфейсом приложения для оформления заказа на покупку книг в магазине, производить обработку и отправку письма с заказом пользователя с помощью почтового сервиса для дальнейшей выдачи. Также необходимо реализовать отдельный доступ к приложению для администратора с возможностью создания, удаления и редактирования списка доступных для покупки книг, выводением статистики по продажам за выбранный период в виде графика.

Программное обеспечение должно эксплуатироваться на персональном компьютере.

Программа должна обеспечивать возможность выполнения перечисленных ниже функции:

- сохранять название книги, название издательства, себестоимость, ФИО автора, год издания, цену для продажи,
- хранить данные о пользователях и их корзине с выбранными товарами,
- отображать статистику по продажам за год по месяцам в виде графика,
- возможность добавления, удаления, редактирования параметров книг,
- предоставлять возможность сортировки списка книг по дате издания, имени, цене,

- авторизация пользователя в приложении с возможностью создания новой учетной записи и выхода из нее,
- проверка надежности и скрытие пароля при вводе данных в соответствующем поле,
- предоставлять пользователю возможность добавления товаров в корзину,
- оформление заказа с указанием данных о городе, номере почтового отделения, номере телефона,
- отправка письма с информацией о заказе пользователя магазину с использованием почтового сервиса.

Для реализации данной интеллектуальной системы будет использован язык программирования C# и сопутствующие ему библиотеки. Приложение должно иметь интуитивно понятный и привлекательный интерфейс.

Проектирование интерфейса будет реализовано с помощью платформы Windows Forms. Должны быть разработаны окна авторизации в программе интернет-магазина, главная страница для пользователя. Окно администратора предусматривает панель продаж и редактирования списка книг. Для внесения изменений должны быть созданы окна добавления и изменения книг.

Клиентская программа предназначена для пользователя и администратора которая будет общаться с базой данных посредством отправки запросов ей.

База данных должна принимать запросы от клиента, а также хранить и передавать данные о пользователях, их корзинах с выбранными товарами в интернет-магазине, списке товаров доступных для заказа, информацию о продажах.

Проектирование базы данных будет реализовано с помощью СУБД Microsoft SQL Server. Для конфигурирования, управления и администрирования компонентов базы данных будет использована

специальная утилита SQL Server Management Studio (SSMS). Она включает в себя скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера. Главным инструментом SSMS является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера и управлять ими.

3.3 Стадии и этапы разработки

Стадии и этапы разработки приложения представлены в таблице 1.

Таблица 1 – Стадии и этапы разработки приложения

Стадии разработки	Этапы разработки
Техническое задание	Обоснование необходимости разработки программы
	Научно-исследовательские работы
	Разработка и утверждение технического задания
Рабочий проект	Разработка программы
	Разработка базы данных
Внедрение проекта	Подготовка и передача программы

3.4 Выводы

В рамках этого раздела определены требования к программе, а также стадии и этапы разработки.

4 Описание программы

4.1 Общие сведения

Интеллектуальная система для обработки, хранения и передачи данных написана на языке C# в среде разработки Microsoft Visual Studio 2022.

Разработанная программа имеет название «Книжный магазин» и располагается в файле ShopStore.exe. Листинг программы представлен в приложении А.

4.2 Функциональное назначение

Функциональное назначение интеллектуальной системы для обработки, хранения и передачи данных состоит в следующем:

- хранение данных о книжном ассортименте, включая информацию о названии, авторах, издательствах, ценах, и других характеристиках,
- обеспечение безопасности и целостности данных, а также защиты от несанкционированного доступа,
- оформление пользователем заказов из интернет-магазина,
- отправка собранной информации о заказе с помощью почтового сервиса,
- отображение данных о продажах в виде таблиц,
- предоставлять возможность сортировки списка книг по дате издания, имени, цене,
- авторизация пользователя в приложении с возможностью создания новой учетной записи и выхода из нее,

4.3 Проектирование взаимодействия между клиентом и приложением

Взаимодействие пользователя и администратора с приложением книжного интернет-магазина можно продемонстрировать в виде схем, приведённых на рисунке 18 и на рисунке 19.

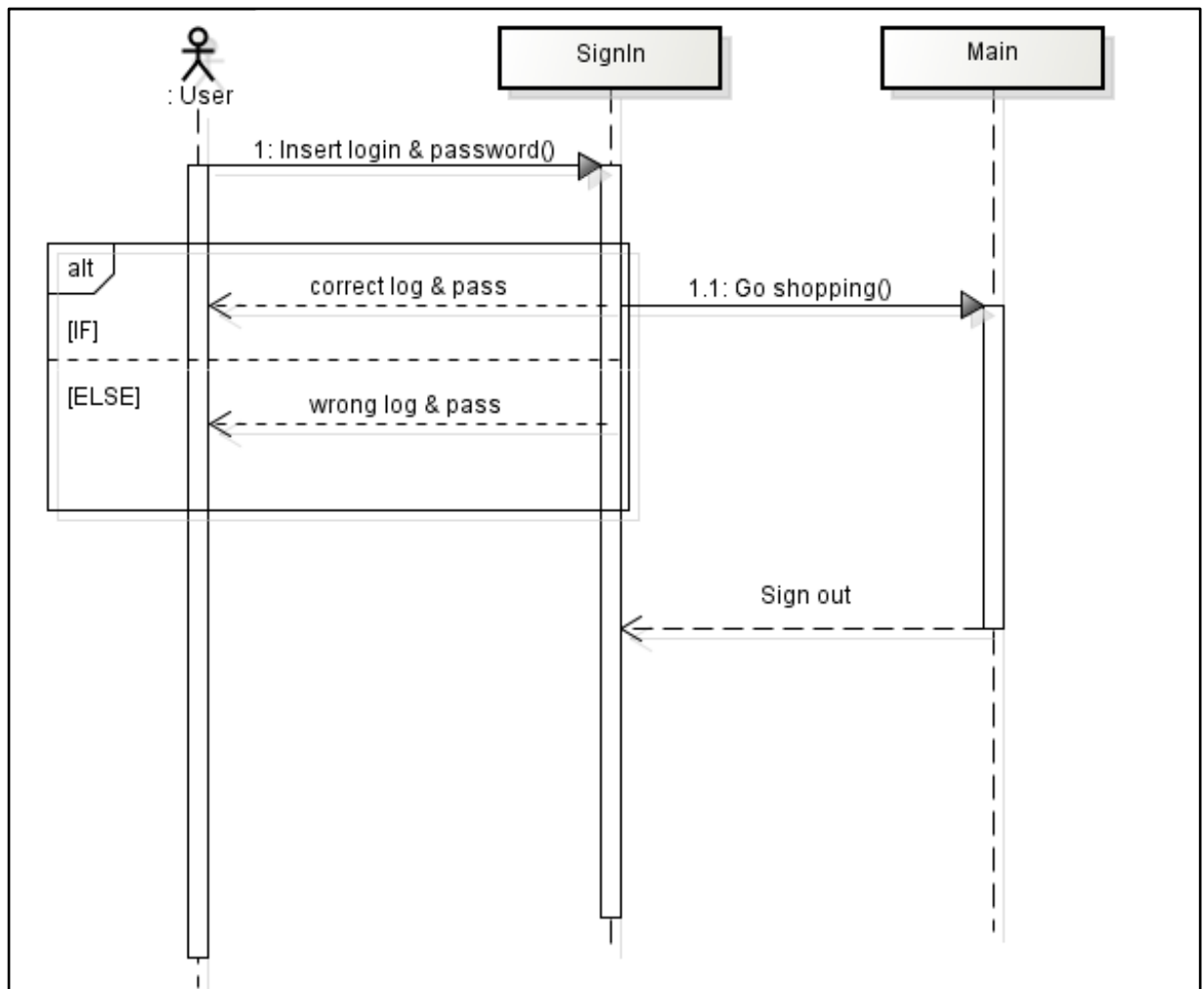


Рисунок 18 – Диаграмма взаимодействий пользователя и приложения

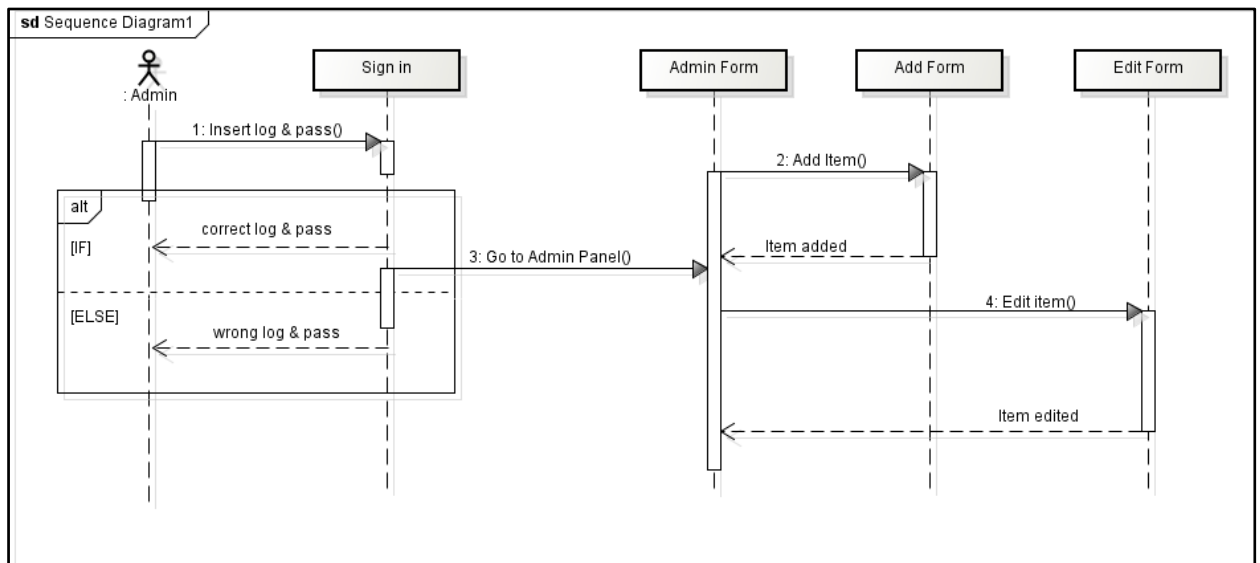


Рисунок 19 – Диаграмма взаимодействий администратора и приложения

4.4 Архитектура программы

Архитектура программы состоит из четырех таблиц базы данных, окон пользователя с передачей данных о заказе и администратора с добавлением, удалением данных о новых книгах, редактировании уже имеющихся товаров и построения графиков на основе продаж. Их взаимодействие показано на рисунке 20.

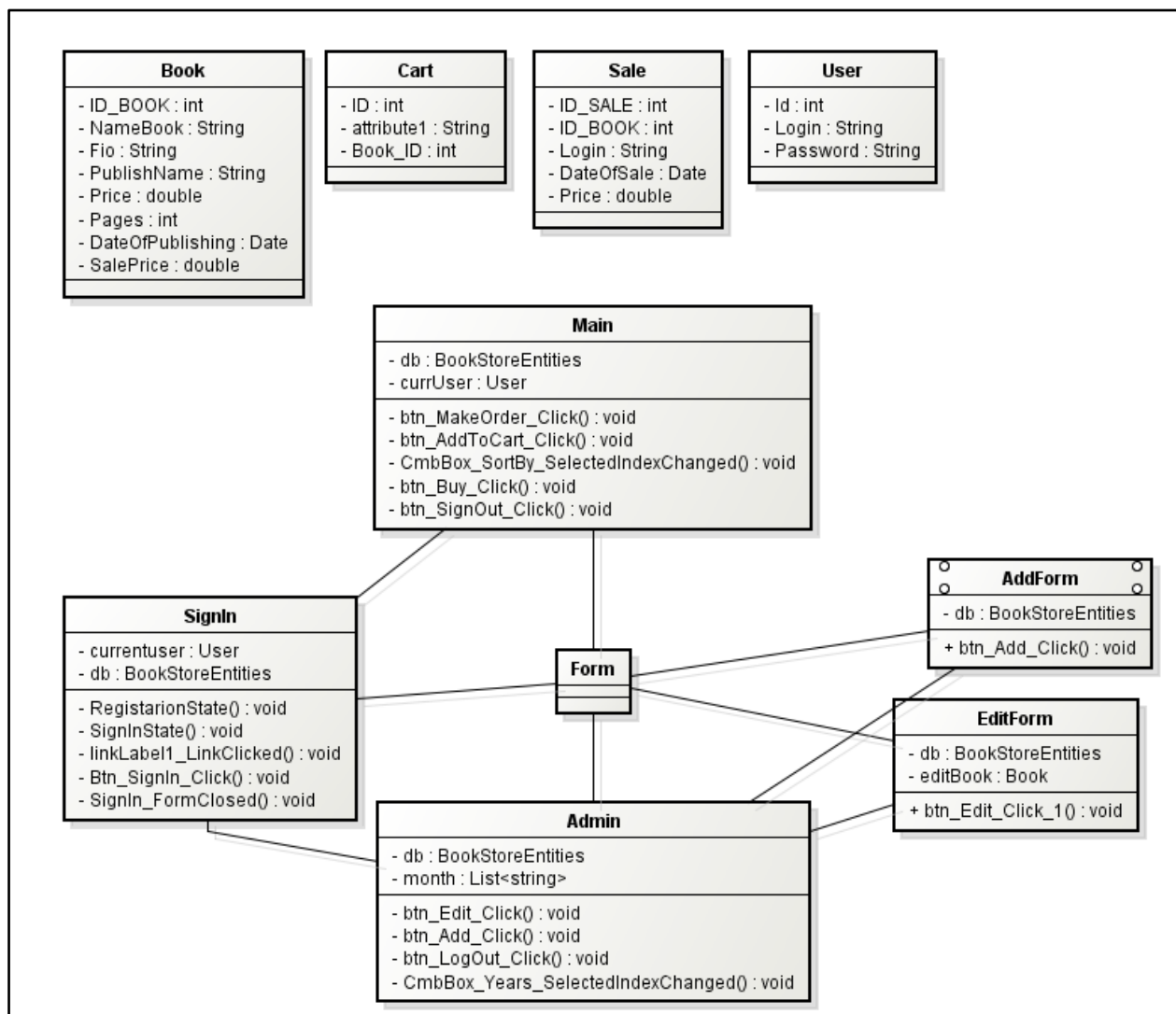


Рисунок 20 – Архитектура программы

4.5 Архитектура базы данных

В данном проекте база данных была реализована с помощью Microsoft SQL Server. Также для и администрирования БД использована специальная утилита SQL Server Management Studio. Для оперирования этими данными произведём подключение к серверу как показано на рисунке 21.

Когда соединение с сервером было успешно установлено, отроем окно создания элементов, добавим новую базу данных и назовем ее «BookStore». Окно создания базы данных представлено на рисунке 22.

Connect to Server

SQL Server

Login Connection Properties Always Encrypted Additional Connection Parameters

Server

Server type: Database Engine

Server name: DESKTOP-A10QJIN

Authentication: Windows Authentication

User name: DESKTOP-A10QJIN\Daren

Password:

☐ Remember password

Connection Security

Encryption: Optional

☐ Trust server certificate

Host name in certificate:

Connect Cancel Help Options <<

Рисунок 21 – Подключение к серверу SQL

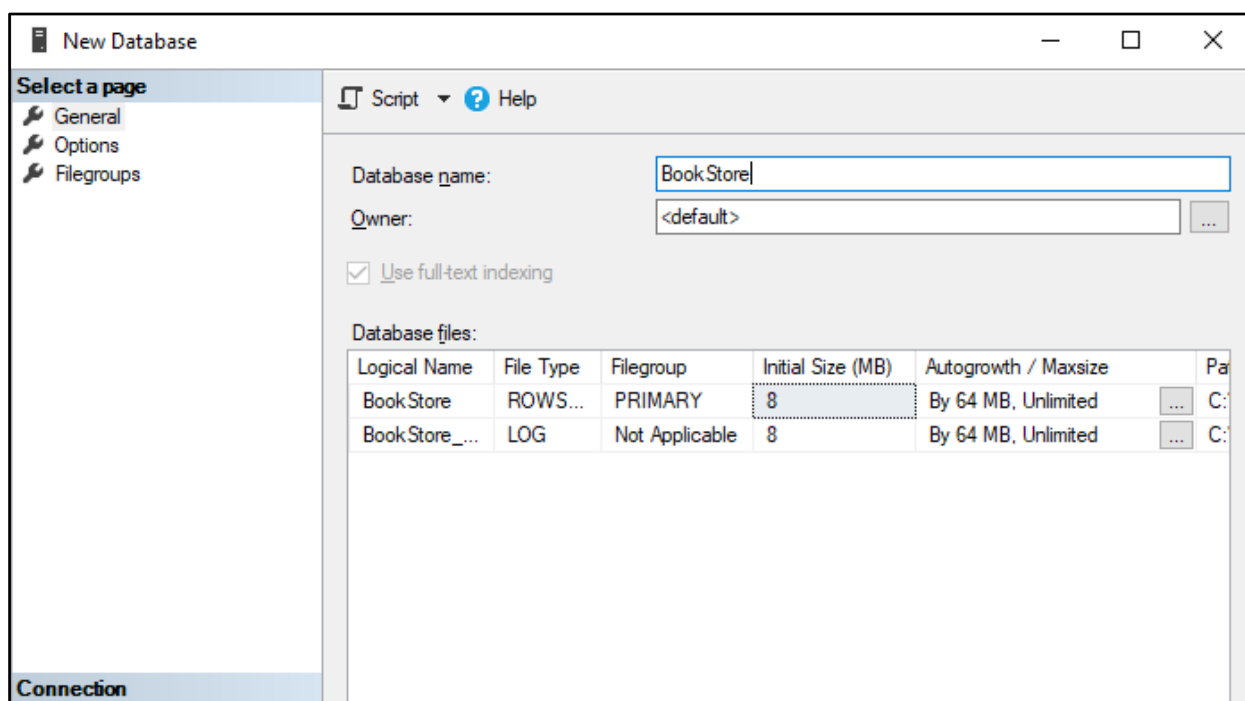


Рисунок 22 – Создание базы данных

Для хранения данных, полученных из приложения, были созданы таблицы «User», «Cart», «Book» и «Sales» представленные на рисунке 23.

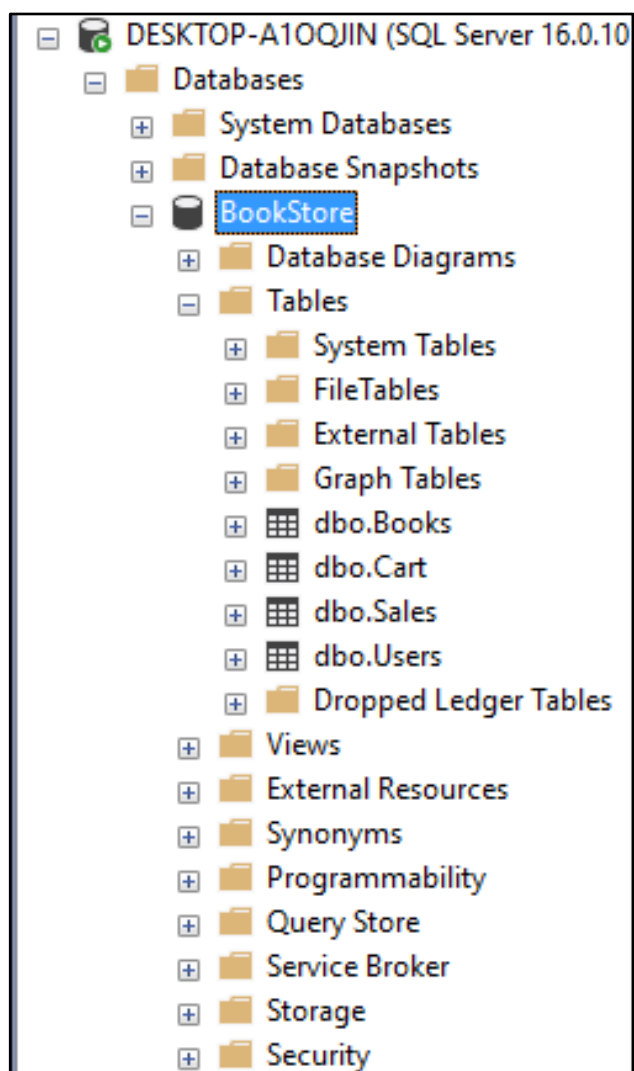


Рисунок 23 – Архитектура БД и таблиц

4.6 Структура хранения данных

Данные о продажах интернет-магазина сохраняются в таблице и представлены в виде полей номер операции, идентификатор книги, имя пользователя, даты продажи и цены (рисунок 24).

```
ID_SALE int primary key identity(1,1),  
ID_BOOK int not null,  
Login nvarchar(32) not null,  
DateOfSale date not null,  
Price money not null,
```

Рисунок 24 – Структура таблицы базы данных о продажах

Информация о пользователях хранится в хешированном виде и представлена полями идентификатор пользователя, его логин и пароль (рисунок 25).

```
Id int primary key identity(1,1),  
Login nvarchar(32) not null unique,  
Password nvarchar(32) not null,
```

Рисунок 25 – Структура таблицы базы данных о пользователях

В базе данных также реализована таблица, описывающая корзину для каждого пользователя. Она реализована полями идентификатор пользователя, его логин и номер книги в списке представленных товаров (рисунок 26).

```
Id int primary key identity(1,1),  
User_Login nvarchar(32) not null,  
Book_ID int not null,
```

Рисунок 26 – Структура таблицы базы данных о корзине

Данные о книгах доступных к покупке представлены с помощью полей идентификатор книги, название, ФИО автора,

название издательства, цена покупки, количество страниц, дата публикации, цена для продажи. Они показаны ниже на рисунке 27.

```
ID_BOOK int primary key identity(1,1),  
NameBook nvarchar(32) not null,  
Fio nvarchar(64) not null,  
PublishName nvarchar(64) not null,  
Price money not null,  
Pages int not null,  
DateOfPublishing date not null,  
SalePrice money not null
```

Рисунок 27 – Структура таблицы базы данных о книгах

4.7 Отладка и тестирование

В ходе разработки отладка проводилась с использованием встроенных в Visual Studio средств. Применялись такие инструменты как точка остановки, пошаговое выполнение программного кода, анализ содержимого глобальных и локальных переменных, данные о работе некоторых функций, вывод информационных сообщений. Тестирование проводилось в процессе всего цикла разработки. Все выявленные ошибки были исправлены.

В качестве примера тестирования и отладки на рисунке 28 представлен процесс исправления ошибки в работе функции, осуществляющей проверку данных пользователя на корректность условий безопасности.

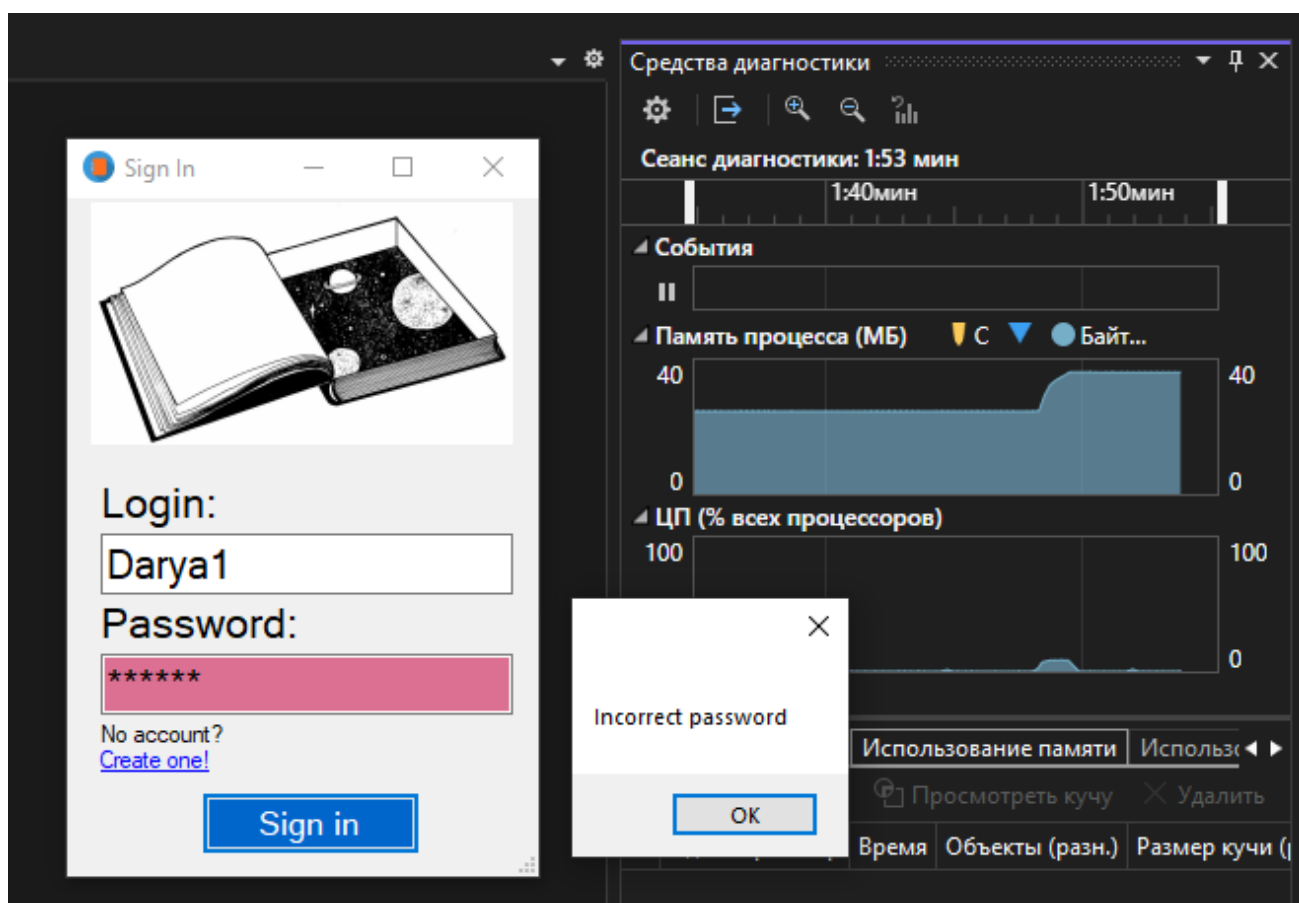


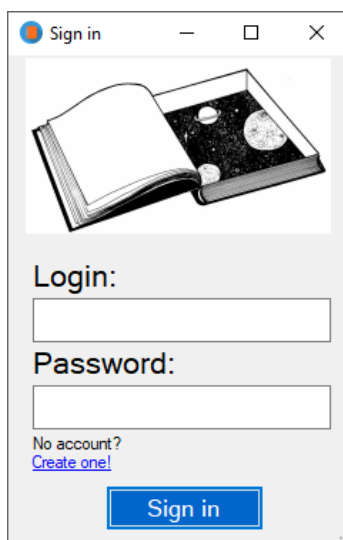
Рисунок 28 – Пример ошибки во время тестирования

В рамках работ, описанных в данном разделе, все компоненты финального варианта программы были проверены на работоспособность в ходе итераций тестирования, которые выполнялись на протяжении всего цикла разработки каждого из компонентов программы.

4.8 Руководство пользователя по использованию программы

После запуска пользователю выводится окно, представленное

на рисунке 29. В этом окне необходимо ввести логин и пароль пользователя.



Sign in

— □ ×

Illustration of an open book with a space-themed cover.

Login:

—

Password:

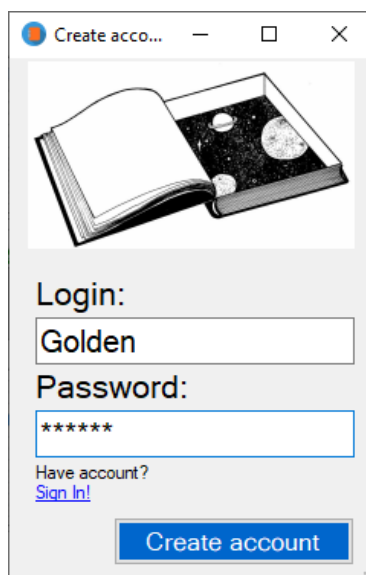
—

No account?
[Create one!](#)

Sign in

Рисунок 29 – Окно авторизации

Если пользователь еще не имеет своей учетной записи, он может создать ее с помощью кнопки «Create one!». Создание аккаунта нового пользователя продемонстрировано на рисунке 30.



Create acco...

— □ ×

Illustration of an open book with a space-themed cover.

Login:

Golden

Password:

Have account?
[Sign In!](#)

Create account

Рисунок 30 – Окно создания аккаунта

На рисунке 31 демонстрируется корректный ввод данных учетной записи пользователя.

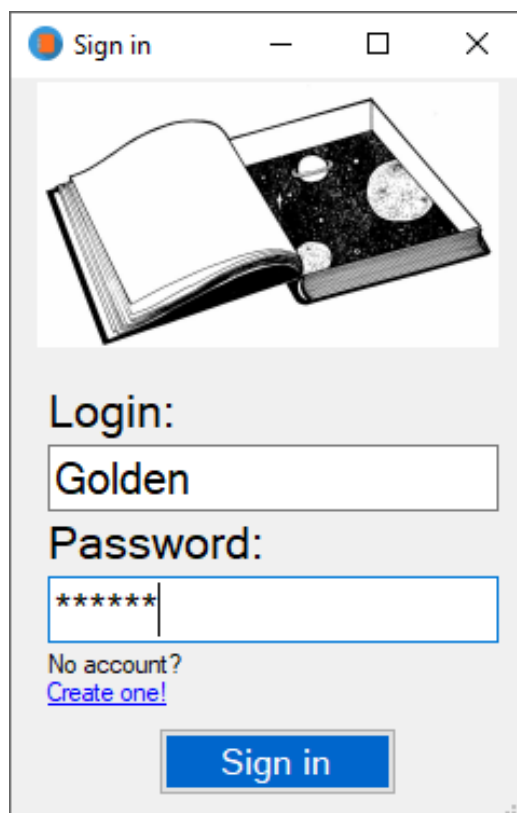


Рисунок 31 – Корректный ввод данных пользователя

После входа в систему пользователя встречает основное окно магазина, изображенное на рисунке 32, в котором он может ознакомиться со списком доступных к покупке книг, отсортировать их по времени добавления, имени и цене. Так же реализована кнопка выхода из приложения. Выбрав понравившуюся книгу и нажав кнопку «Add to cart», отобразится корзина, в которую будет помещен выбранный товар.

Интерфейс корзины продемонстрирован на рисунке 33. Когда

пользователь закончит выбор товаров и нажмет на кнопку «Order» ему станет доступно окно ввода данных, необходимых для отправки. Так же на рисунке 34 демонстрируется отсортированный по имени список товаров, название выбранной для заказа книги и ее цена. После нажатия на кнопку «Buy», заказ будет автоматически отправлен в магазин с помощью почтового сервиса. Письмо содержит данные о городе, номере почтового отделения и номере телефона и заказ пользователя. Пример заполнения полей для доставки приведен на рисунке 35. Интерфейс после завершения покупки представлен на рисунке 36.

ID_BOOK	NameBook	Fio	PublishName	Price	Pages	DateOfPublishing	SalePrice
1	Затерянный мир	Артур Конан До...	GOD1	128,0000	254	16.10.1912	3255,0000
5	Звездные войны	Джордж Лукас	GOD1	128,0000	254	25.05.1977	3255,0000
6	Игра престолов	Джордж Р.Р. М...	GOD1	64,0000	254	01.08.1996	12,0000
7	Марсианин	Энди Уир	GOD1	128,0000	254	06.10.2011	3255,0000
9	Бойцовский клуб	Чак Паланик	GOD1	128,0000	254	17.08.1996	3255,0000
10	Матрица	Лана и Лилли В...	GOD1	64,0000	254	31.03.1999	12,0000
11	Затерянный след	Артур Конан До...	GOD1	128,0000	254	16.10.1912	3255,0000
12	Кровавый след	Стивен Кинг	GOD1	64,0000	254	28.06.1977	12,0000
14	След на земле	Фрэнк Герберт	GOD1	64,0000	254	01.06.1965	12,0000
15	След в космосе	Жюль Верн	GOD1	128,0000	254	25.11.1864	3255,0000
16	След в тени	Джордж Р.Р. М...	GOD1	64,0000	254	01.08.1996	12,0000
17	След в огне	Энди Уир	GOD1	128,0000	254	06.10.2011	3255,0000
18	След в снегу	Чак Паланик	GOD1	64,0000	254	17.08.1996	12,0000

Рисунок 32 – Главный экран пользователя

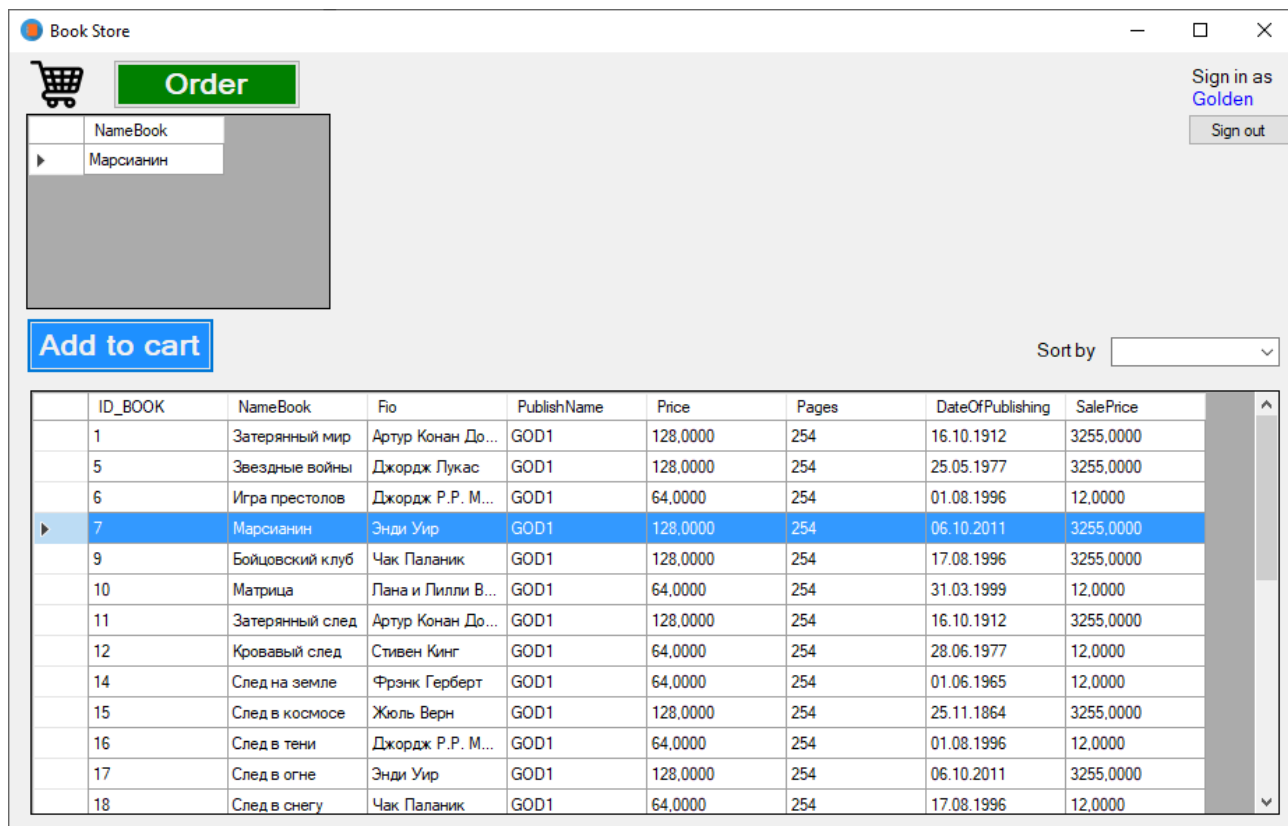


Рисунок 33 – Интерфейс корзины пользователя

Order

NameBook
Марсианин
Затерянный след

Checkout

City

Post office number

Phone number

Марсианин
Price: 128\$

Sign in as Golden

Sign out

Add to cart

Buy

Sort by Name

ID_BOOK	NameBook	Fio	PublishName	Price	Pages	DateOfPublishing	SalePrice
9	Бойцовский клуб	Чак Паланик	GOD1	128,0000	254	17.08.1996	3255,0000
1	Затерянный мир	Артур Конан До...	GOD1	128,0000	254	16.10.1912	3255,0000
11	Затерянный след	Артур Конан До...	GOD1	128,0000	254	16.10.1912	3255,0000
5	Звездные войны	Джордж Лукас	GOD1	128,0000	254	25.05.1977	3255,0000
6	Игра престолов	Джордж Р.Р. М...	GOD1	64,0000	254	01.08.1996	12,0000
12	Кровавый след	Стивен Кинг	GOD1	64,0000	254	28.06.1977	12,0000
7	Марсианин	Энди Уир	GOD1	128,0000	254	06.10.2011	3255,0000
10	Матрица	Лана и Лилли В...	GOD1	64,0000	254	31.03.1999	12,0000
19	След в воде	Лана и Лилли В...	GOD1	128,0000	254	31.03.1999	3255,0000
27	След в воде	Лана и Лилли В...	GOD1	128,0000	254	31.03.1999	3255,0000
34	След в воде	Лана и Лилли В...	GOD1	128,0000	254	31.03.1999	3255,0000
31	След в городе	Джордж Р.Р. М...	GOD1	64,0000	254	01.08.1996	12,0000
24	След в городе	Джордж Р.Р. М...	GOD1	64,0000	254	01.08.1996	12,0000

Рисунок 34 – Интерфейс оформления заказа

Order

NameBook
Путешествие к ...
Дюна
Морской след
Темный след
Космическая од...

Checkout

City
Penza

Post office number
440000

Phone number
88005553535

Космическая одиссея
Price: 128\$

Sign in as Coconat

Sign out

Add to cart

Buy

Sort by

ID_BOOK	NameBook	Fio	PublishName	Price	Pages	DateOfPublishing	SalePrice
2	Путешествие к ...	Жюль Верн	GOD1	64,0000	254	25.11.1864	12,0000
3	Космическая од...	Артур Кларк	GOD1	128,0000	254	03.04.1968	3255,0000
4	Дюна	Фрэнк Герберт	GOD1	64,0000	254	01.06.1965	12,0000
5	Звездные войны	Джордж Лукас	GOD1	128,0000	254	25.05.1977	3255,0000
6	Игра престолов	Джордж Р.Р. М...	GOD1	64,0000	254	01.08.1996	12,0000
7	Марсианин	Энди Уир	GOD1	128,0000	254	06.10.2011	3255,0000
9	Бойцовский клуб	Чак Паланик	GOD1	128,0000	254	17.08.1996	3255,0000
10	Матрица	Лана и Лилли В...	GOD1	64,0000	254	31.03.1999	12,0000
11	Затерянный след	Артур Конан До...	GOD1	128,0000	254	16.10.1912	3255,0000
12	Кровавый след	Стивен Кинг	GOD1	64,0000	254	28.06.1977	12,0000
13	Морской след	Артур Кларк	GOD1	128,0000	254	03.04.1968	3255,0000
14	След на земле	Фрэнк Герберт	GOD1	64,0000	254	01.06.1965	12,0000
15	След в космосе	Жюль Верн	GOD1	128,0000	254	25.11.1864	3255,0000

Рисунок 35 – Пример введенных данных покупателя

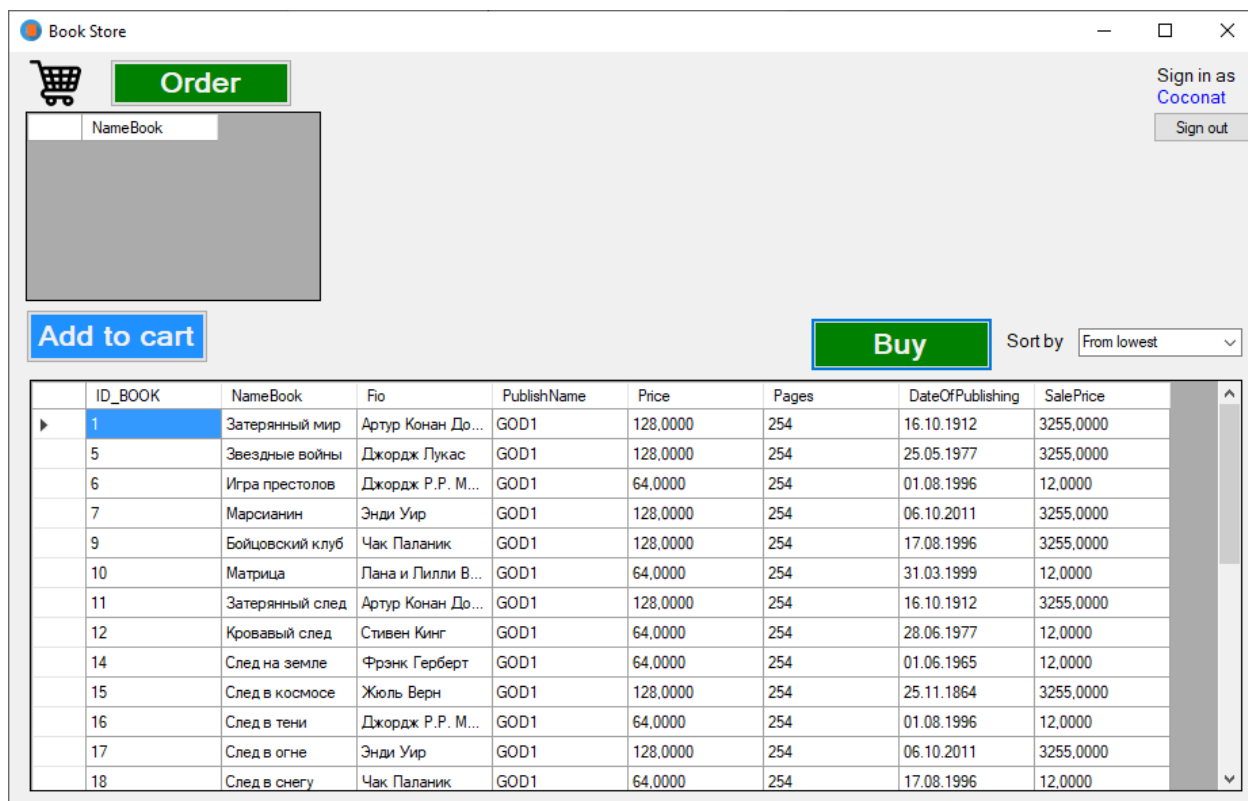


Рисунок 36 – Окно завершения покупки

Если при вводе логина и пароля будут указаны данные учетной записи администратора, программа отобразит окно с вкладками изменения информации о товаре и статистике покупок.

Во вкладке продаж, изображенном на рисунке 37, отображается информация о покупках пользователей в виде списка. Так же можно выбрать период, за который будет показана выручка интернет-магазина по месяцам в виде графика.

Во вкладке редактора товаров (рисунок 38) администратор может удалять, добавлять и редактировать записи о книгах.

При нажатии на кнопки «Add» будет открыто окно создания новой

записи (рисунок 39). Интерфейс представлен полями, в которые вносятся название книги, ФИО автора, название издательства, закупочная цена, количество страниц, цена для продажи и дата публикации книги. После добавления в систему, книге так же будет присвоен свободный идентификатор из базы данных товаров.

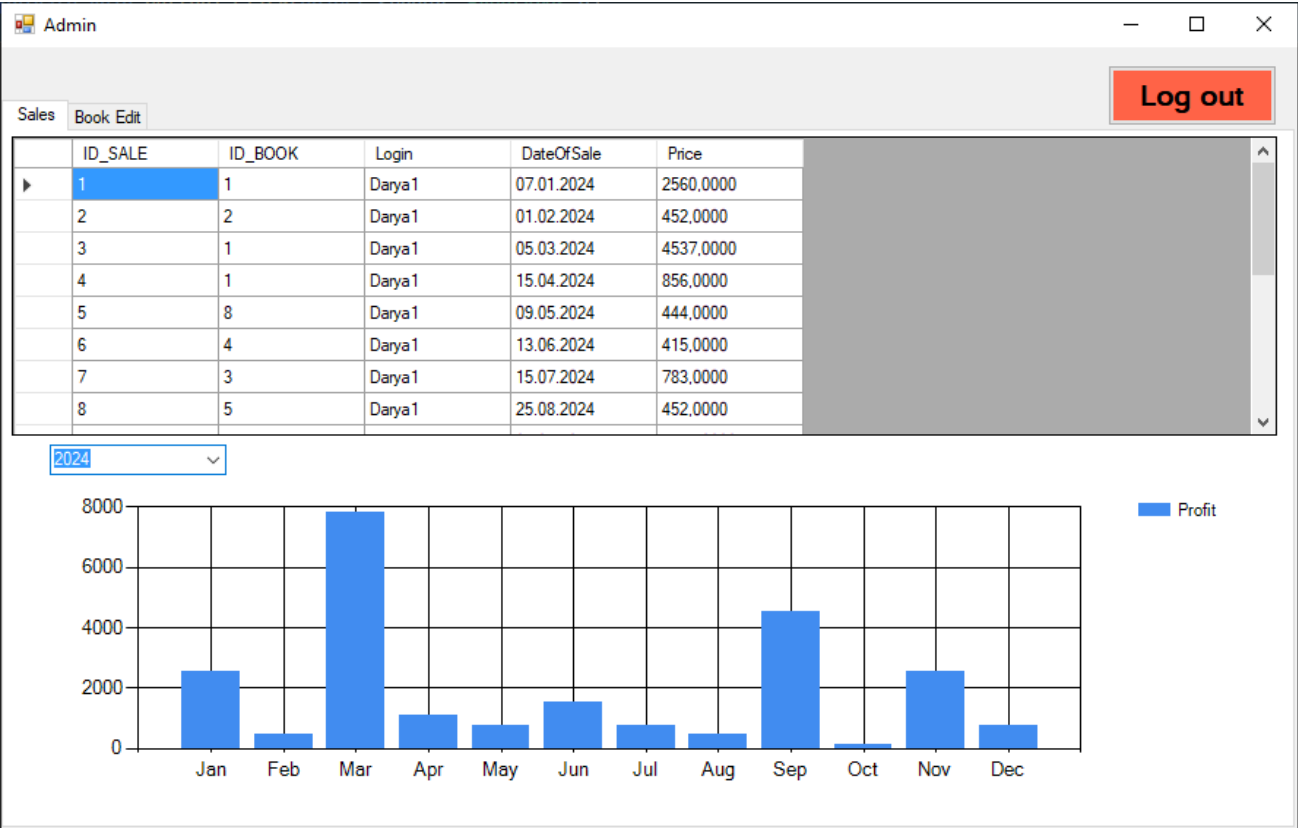


Рисунок 37 – Окно статистики продаж

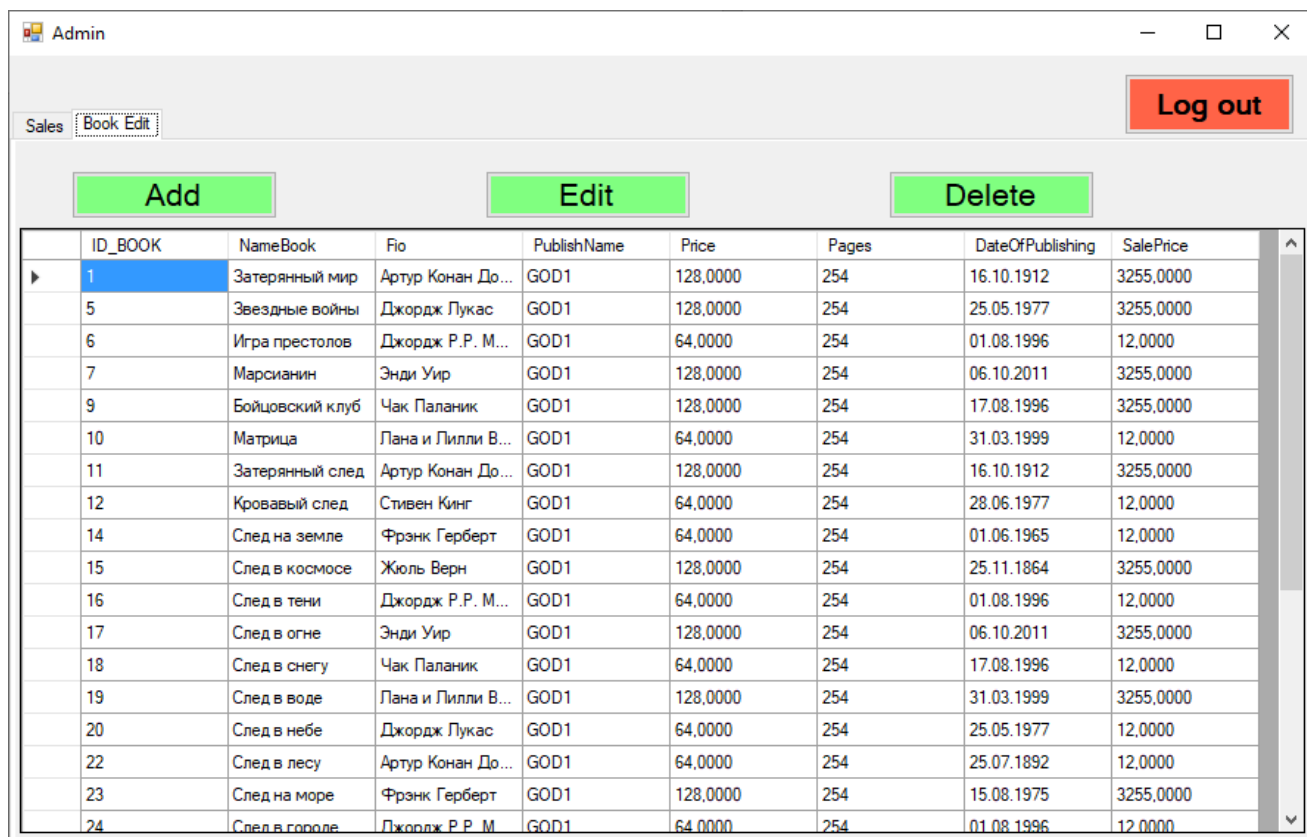


Рисунок 38 – Окно редактора товаров

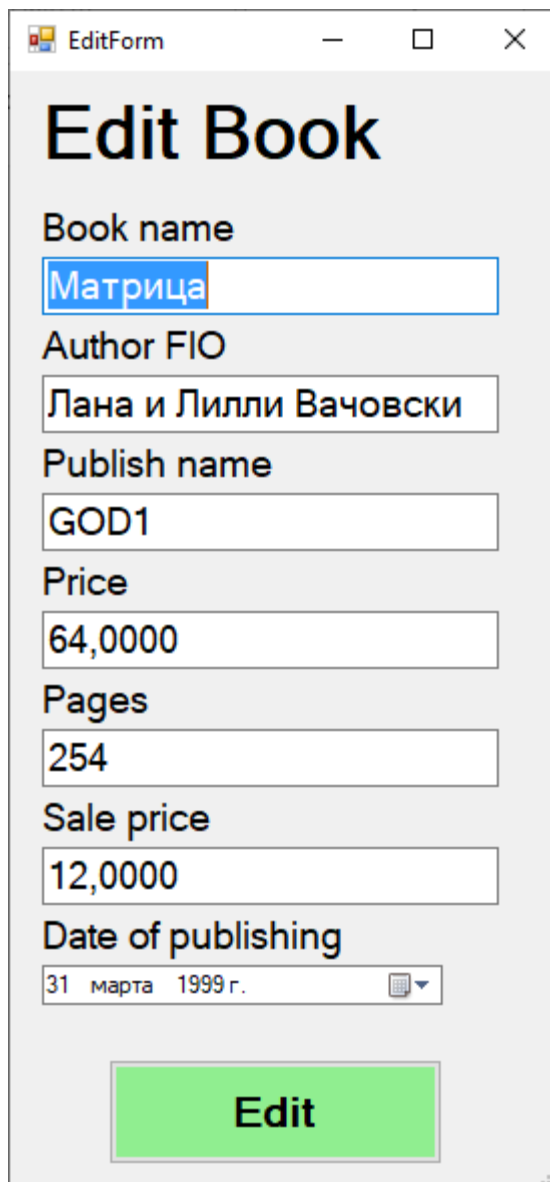
The image shows a software window titled "AddForm" with a standard Windows title bar (minimize, maximize, close buttons). The main content area has a light gray background and is titled "New Book" in a large, bold, black font. Below the title, there are seven labeled input fields arranged vertically: "Book name" with the text "Алиса в стране чудес", "Author FIO" with "Льюис Кэрролл", "Publish name" with "GOD1", "Price" with "300", "Pages" with "254", "Sale price" with "350", and "Date of publishing" with a date picker showing "22" and "июня 2024 г.". At the bottom center of the form is a large green button with the word "Add" in bold black text.

Рисунок 39 – Окно добавление новой книги

При нажатии на кнопки «Edit» будет открыто окно редактирования записи о товаре (рисунок 40). Интерфейс данного окна представлен полями, в которые вносятся исправления в поля название книги, ФИО автора, название издательства, закупочная цена, количество страниц, цена для продажи и дата публикации книги. После принятия изменений в систему.

Пользователь и администратор могут закрыть программу нажав на крестик в правом верхнем углу окна при этом, не потеряв данные сессии. Рекомендуется выходить из аккаунта с помощью кнопок расположенных также в правом верхнем углу экрана и только после

этого закрывать приложение.



Edit Book

Book name
Матрица

Author FIO
Лана и Лилли Вачовски

Publish name
GOD1

Price
64,0000

Pages
254

Sale price
12,0000

Date of publishing
31 марта 1999 г.

Edit

Рисунок 40 – Окно изменения информации о книге

4.9 Вывод

В этом разделе было приведено описание итогового приложения, описаны тестовые данные и методы, с помощью которых проводилось итоговое функциональное тестирование разработанного приложения, а также составлено руководство пользователя, где подробно с иллюстрациями был описан алгоритм взаимодействия пользователя с приложением.

ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы была разработана интеллектуальная система для обработки, хранения и передачи данных для использования администраторами и пользователями книжного магазина. В качестве пользовательского интерфейса было реализовано приложение для ПК на ОС «Windows». Таким образом, можно заключить, что разработанный проект полностью соответствует поставленным целям.

Данным приложением, благодаря простому пользовательскому интерфейсу, может пользоваться любой уверенный пользователь ПК.

Дальнейшее развитие разработанной системы может включать в себя разработку мобильного приложения, добавление инструментов редактирования корзины и пользователей магазина, добавление ленты рекомендаций товаров для покупателей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Маккэндлесс Д. Инфографика. Самые интересные данные в графическом представлении. – М.: Манн, Иванов и Фербер, 2013. – 264 с.
2. StudFiles: сайт. – URL: <https://studfile.net/preview/5411061/page:63/> (дата обращения: 21.10.2022). – Средства структурирования и визуализации данных. Электронные помощники аналитика.
3. Алексей Колоколов. Институт бизнес аналитики: сайт. – URL: <https://alexkolokolov.com/ru/blog/instrumenty-vizualizacii> (дата обращения: 17.11.2022). – Инструменты визуализации: сервисы, программы и средства обработки данных.
4. Нейт Сильвер. Сигнал и Шум. Почему одни прогнозы сбываются, а другие – нет // Азбука-Аттикус, КоЛибри, 2015
5. Агапов, А.В. Обработка и обеспечение безопасности электронных данных: Учебное пособие / А.В. Агапов, Т.В. Алексеева и др. - М.: МФПУ Синергия, 2012. - 592 с.
6. Сара Бослаф. Статистика для всех. – М.: ДМК Пресс, 2017 – 586.
7. Лучшие платформы для разработки приложений в 2023 году: сайт. – URL: [https:// g2.com/categories/application-development-platforms](https://g2.com/categories/application-development-platforms) (дата обращения: 19.11.2022). – Лучшие платформы для разработки приложений.
8. Желязны Д. Говори на языке диаграмм: Пособие по визуальным коммуникациям для руководителей / Пер. с англ. - М.: Институт комплексных стратегических исследований, 2004 - 220 с.
9. Винстон Уэйн, Бизнес-моделирование и анализ данных. Решение актуальных задач с помощью Microsoft Excel. 6-е издание. – СПб.: Питер, 2021. – 944 с.: ил. – (Серия «IT для бизнеса»).
10. Фридл Дж. Регулярные выражения, 3-е издание. – Пер. с

англ. – СПб.: Символ Плюс, 2008. – 608 с.

11. Официальный сайт «QueryList». [Электронный ресурс]. URL: <https://querylist.cc/> – Свободный доступ. (дата посещения 21.03.2023).

12. Батаев А.В. Операционные системы и среды / Батаев А.В., Налютин Н.Ю., Синицын С.В. – М.: Академия, 2015. – 272 с.

13. Архитектура интеллектуальных информационных систем: сайт. – URL: <https://textarchive.ru/c-1757577-p3.html>.

14. Поклад, П.М. Анализ методов интеллектуализации управления сложными динамическими объектами / Вестник ИГЭУ. 2010. № 2. С.1–4 URL:<https://cyberleninka.ru/article/n/analiz-metodovintellektualizatsiipravleniya-slozhnymi-dinamicheskimi-obektami>.

15. Современные интеллектуальные системы сайт. – URL: <https://studme.org/201004/informatika/zaklyuchenie>.

16. Тенденции развития информационных систем сайт. – URL: <https://infopedia.su/17xe45c.html>.

17. Современные информационные технологии [Электронный ресурс]. – URL: https://bstudy.net/sovremennye_informatsionnye_tehnologii.

18. Интеллектуальные системы хранения данных сайт. – URL: <https://controlengrussia.com/programmnye-sredstva/intellektualnye-sistemy-khraneniya-dannykh-v-asu-tp>.

19. Программы для торговли в магазине [Электронный ресурс]. URL: <https://www.cleverence.ru/articles/elektronnaya-kommertsiya/programmy-dlya-torgovli-v-magazine-reyting-luchshikh-po-dlya-rozничной-prodazhi/>

20. Программы для учёта товаров: сайт. – URL: <https://zhiza.evotor.ru/luchshie-programmy-dlya-tovarnogo-uchyota-funkcionalnye-vozmozhnosti-i-sravnenie-top-8-programm/>

21. Что такое базы данных: сайт. – URL: <https://practicum.yandex.ru/blog/chto-takoe-bazy-dannyh/>

ПРИЛОЖЕНИЕ А

(обязательное)

Интеллектуальная система для обработки, хранения и передачи
данных

Листинг программы

Файл «main.cs»

```
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;
using System.Configuration;
using System.Data.SqlClient;
using System.Data.Linq;
using System.Net.Mail;
using System.Net;

namespace ShopStore
{
    public partial class Main : Form
    {
        BookStoreEntities db = null;
        public User currUser = null;

        int BOOK_ID = 0;
        public Main()
        {
            InitializeComponent();

            this.MinimumSize = new System.Drawing.Size(938, 597); ;
            this.MaximumSize = new System.Drawing.Size(938, 597); ;

            db = new BookStoreEntities();

            SignIn signInForm = new SignIn();
            if (signInForm.ShowDialog() == DialogResult.OK)
            {
                currUser = signInForm.currentUser;
                linkLable_login.Text = currUser.Login;
            }
            else
            {
                return;
            }
            dataGridView_Books.DataSource = db.Books.ToList();

            dataGridView_Cart.DataSource = (from item in db.Books
                                           join item2 in db.Carts on item.ID_BOOK equals item2.Book_ID
                                           where item2.User_Login == currUser.Login
                                           select item).ToList();
        }
    }
}
```

```

dataGridView_Cart.Columns["ID_BOOK"].Visible = false;
dataGridView_Cart.Columns["Fio"].Visible = false;
dataGridView_Cart.Columns["PublishName"].Visible = false;
dataGridView_Cart.Columns["Price"].Visible = false;
dataGridView_Cart.Columns["Pages"].Visible = false;
dataGridView_Cart.Columns["DateOfPublishing"].Visible = false;
dataGridView_Cart.Columns["SalePrice"].Visible = false;

cmbBox_SortBy.SelectedIndexChanged +=
CmbBox_SortBy_SelectedIndexChanged;
cmbBox_SortBy.Items.Add("Old");
cmbBox_SortBy.Items.Add("New");
cmbBox_SortBy.Items.Add("Name");
cmbBox_SortBy.Items.Add("From lowest");
cmbBox_SortBy.Items.Add("From highest");

}

private void CmbBox_SortBy_SelectedIndexChanged(object sender, EventArgs e)
{
    int filter = cmbBox_SortBy.SelectedIndex;

    switch (filter)
    {
        case 0:
            dataGridView_Books.DataSource = db.Books.OrderBy(sort =>
sort.DateOfPublishing).ToList();
            break;
        case 1:
            dataGridView_Books.DataSource = db.Books.OrderByDescending(sort =>
sort.DateOfPublishing).ToList();
            break;
        case 2:
            dataGridView_Books.DataSource = db.Books.OrderBy(sort =>
sort.NameBook).ToList();
            break;
        case 3:
            dataGridView_Books.DataSource = db.Books.OrderBy(sort =>
sort.Price).ToList();
            break;
        case 4:
            dataGridView_Books.DataSource = db.Books.OrderByDescending(sort =>
sort.Price).ToList();
            break;
    }
}

private void btn_SignOut_Click(object sender, EventArgs e)
{

```

```

SignIn signInForm = new SignIn();
this.Hide();
if (signInForm.ShowDialog() == DialogResult.OK)
{
    currUser = signInForm.currentUser;
}
linkLabel_login.Text = currUser.Login;
this.Show();
dataGridView_Cart.DataSource = (from item in db.Books
                                join item2 in db.Carts on item.ID_BOOK equals item2.Book_ID
                                where item2.User_Login == currUser.Login
                                select item).ToList();
}

private void btn_Buy_Click(object sender, EventArgs e)
{
    string nameOfBook = "";
    var deleteObj = from item in db.Books
                    where item.ID_BOOK == BOOK_ID
                    select item;

    foreach (var item in deleteObj) {
        db.Books.Remove(item);
        nameOfBook = item.NameBook;
        Sale new_sale = new Sale() { ID_BOOK = BOOK_ID, Login = currUser.Login,
        DateOfSale = DateTime.Now, Price = item.SalePrice };
        db.Sales.Add(new_sale);
    }
    db.SaveChanges();

    //Send to message
    string msg = "City: " + textBox_City.Text + ", Post office: " + textBox_Post.Text + ",
    Phone number: " + textBox_Phone.Text + ", Book: " + nameOfBook;
    MailMessage message = new MailMessage("konda*****@gmail.com",
    "mina*****@gmail.com", "order", msg);

    SmtplibClient client = new SmtplibClient("smtp.gmail.com", 587);

    client.Credentials = new NetworkCredential("konda*****@gmail.com", "***** *****
***** *****");
    client.EnableSsl = true;
    client.Send(message);

    dataGridView_Cart.DataSource = (from item in db.Books
                                    join item2 in db.Carts on item.ID_BOOK equals item2.Book_ID
                                    where item2.User_Login == currUser.Login
                                    select item).ToList();
    dataGridView_Books.DataSource = db.Books.ToList();
    dataGridView_Cart.Refresh();
}

```

```

        dataGridView_Cart.Update();

        groupBox1.Visible = false;
    }
    private void pictureBox_Cart_Click(object sender, EventArgs e)
    {
        if (dataGridView_Cart.Visible)
        {
            dataGridView_Cart.Visible = false;
            btn_MakeOrder.Visible = false;

        }
        else
        {
            dataGridView_Cart.Visible = true;
            btn_MakeOrder.Visible = true;

        }
        dataGridView_Cart.ClearSelection();
    }
    void btn_AddToCart_Click(object sender, EventArgs e)
    {
        if (dataGridView_Books.SelectedRows.Count == 1)
        {
            int index = dataGridView_Books.SelectedRows[0].Index;
            int ID = 0;
            if (Int32.TryParse(dataGridView_Books[0, index].Value.ToString(), out ID))
            {
                Cart cartItem = new Cart() { Book_ID = ID, User_Login = currUser.Login };
                db.Carts.Add(cartItem);
                db.SaveChanges();
                dataGridView_Cart.DataSource = (from item in db.Books
                                                join item2 in db.Carts on item.ID_BOOK equals
item2.Book_ID
                                                where item2.User_Login == currUser.Login
                                                select item).ToList();
                dataGridView_Cart.Refresh();
                dataGridView_Cart.Update();
            }
        }
        else
        {
            MessageBox.Show("Chose one book");
        }
    }

    private void btn_MakeOrder_Click(object sender, EventArgs e) {
        groupBox1.Visible = true;
        btn_Buy.Visible = true;
    }

```

```

//label3.Text = " ";
if (dataGridView_Cart.SelectedRows.Count == 1) {
    int index = dataGridView_Cart.SelectedRows[0].Index;

    if (Int32.TryParse(dataGridView_Cart[0, index].Value.ToString(), out BOOK_ID))
    {
        var deleteObj = from item in db.Books
                        where item.ID_BOOK == BOOK_ID
                        select item;
        foreach (var item in deleteObj) {
            lbl_Price.Text = Convert.ToInt32(item.Price).ToString() + "$";
            lbl_BookForBuy.Text = item.NameBook.ToString();
        }
    }
}
}
}
}
}
}
}
}

```

Файл «Admin.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Linq;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ShopStore
{
    public partial class Admin : Form
    {
        BookStoreEntities db = null;
        List<string> month = null;
        public Admin()
        {
            InitializeComponent();
            db = new BookStoreEntities();

            cmbBox_Years.Items.Add("2023");
            cmbBox_Years.Items.Add("2024");
            cmbBox_Years.Items.Add("2025");

            cmbBox_Years.SelectedIndexChanged +=
CmbBox_Years_SelectedIndexChanged;

            dataGridView_Sales.DataSource = db.Sales.ToList();

```

```

        month = new List<string>()
        {
            "Jan",
            "Feb",
            "Mar",
            "Apr",
            "May",
            "Jun",
            "Jul",
            "Aug",
            "Sep",
            "Oct",
            "Nov",
            "Dec"
        };
        chart1.ChartAreas[0].AxisX.Interval = 1;
        dataGridView_Books.DataSource = db.Books.ToList();
    }

    private void CmbBox_Years_SelectedIndexChanged(object sender, EventArgs e)
    {
        chart1.Series["Profit"].Points.Clear();
        int year = 0;

        int index = cmbBox_Years.SelectedIndex;

        switch (index)
        {
            case 0: year = 2023; break;
            case 1: year = 2024; break;
            case 2: year = 2025; break;
        }
        var select = db.Sales.Where(g => g.DateOfSale.Year == year).GroupBy(s =>
s.DateOfSale.Month).Select(s => new { Month = s.Key, Profit = s.Sum(p => p.Price) });

        foreach (var item in select)
        {
            chart1.Series["Profit"].Points.AddXY(month[item.Month - 1],
Convert.ToInt32(item.Profit).ToString());
        }
    }

    private void btn_LogOut_Click(object sender, EventArgs e)
    {
        this.DialogResult = DialogResult.OK;
    }

    private void btn_Add_Click(object sender, EventArgs e)
    {
        AddForm addForm = new AddForm();
    }

```

```

        if (addForm.ShowDialog() == DialogResult.OK)
        {
            dataGridView_Books.DataSource = db.Books.ToList();
            dataGridView_Books.Refresh();
            dataGridView_Books.Update();
        }
    }

    private void btn_Edit_Click(object sender, EventArgs e)
    {
        if (dataGridView_Books.SelectedRows.Count == 1)
        {
            int index = dataGridView_Books.SelectedRows[0].Index;
            int ID = 0;
            if (Int32.TryParse(dataGridView_Books[0, index].Value.ToString(), out ID))
            {
                var editObj = db.Books.Where(g => g.ID_BOOK == ID).Select(g =>
g).FirstOrDefault();

                EditForm editForm = new EditForm(editObj);

                if (editForm.ShowDialog() == DialogResult.OK)
                {
                    db.SaveChanges();
                    dataGridView_Books.DataSource = db.Books.ToList();
                    dataGridView_Books.Refresh();
                    dataGridView_Books.Update();
                }
            }
            else
            {
                MessageBox.Show("Chose one book");
            }
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (dataGridView_Books.SelectedRows.Count == 1)
        {
            int index = dataGridView_Books.SelectedRows[0].Index;
            int ID = 0;
            if (Int32.TryParse(dataGridView_Books[0, index].Value.ToString(), out ID))
            {
                var deleteObj = from item in db.Books
                                where item.ID_BOOK == ID
                                select item;
                foreach (var item in deleteObj)

```

```

        {
            db.Books.Remove(item);
        }
    }
}
db.SaveChanges();
dataGridView_Books.DataSource = db.Books.ToList();
dataGridView_Books.Refresh();
dataGridView_Books.Update();
}
}
}

```

Файл «SignIn.cs»

```

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;
using System.Data.Entity;
using System.Data.Linq;
using System.Text.RegularExpressions;

namespace ShopStore
{
    public partial class SignIn : Form
    {
        public string password = null;
        public bool signInForm = true;
        public User currentUser = null;

        BookStoreEntities db = null;

        string admin_login = "admin1";
        string admin_pass = "admin1";

        public SignIn()
        {
            InitializeComponent();
            txtBox_password.PasswordChar = '*';
            this.FormClosed += SignIn_FormClosed;
            db = new BookStoreEntities();
            this.MaximumSize = new System.Drawing.Size(255, 383);
            this.MinimumSize = new System.Drawing.Size(255, 383);
        }
    }
}

```

```

private void SignIn_FormClosed(object sender, FormClosedEventArgs e)
{
    if (this.DialogResult == DialogResult.OK)
    {

    }
    else
    {
        Application.Exit();
    }
}

private void Btn_SignIn_Click(object sender, EventArgs e)
{
    Regex regex = new Regex("^[a-zA-Z0-9]+([. _]?[a-zA-Z0-9]){5}$");

    string input_login = textBox_login.Text;
    string input_password = textBox_password.Text;

    if (regex.Match(input_login).Success && regex.Match(input_password).Success)
    {
        textBox_password.BackColor = Color.White;
        textBox_login.BackColor = Color.White;

        if (signInForm) //SIGN IN
        {
            //ADMIN
            if (input_login == admin_login && input_password == admin_pass)
            {
                Admin adminForm = new Admin();

                this.Hide();
                if (adminForm.ShowDialog() == DialogResult.OK)
                {
                    this.Show();
                }
                else
                {
                    Application.Exit();
                }
            }
            else
            {
                int correct_login = db.Users.Where(user => user.Login ==
input_login).Count();

                if (correct_login == 1)
                {
                    var correct_password = db.Users.Where(user => user.Login ==
input_login).Where(user => user.Password == input_password.ToString()).Count();

```

```

        if (corect_password == 1)
        {
            this.DialogResult = DialogResult.OK;
            currentUser = new User() { Login = input_login, Password = password
};
        }
        else
        {
            txtBox_password.BackColor = Color.PaleVioletRed;
            MessageBox.Show("Incorrect password");
        }
    }
    else
    {
        txtBox_login.BackColor = Color.PaleVioletRed;
        MessageBox.Show("Incorrect username");
    }
} //END
} //END

else //Create Account
{
    using (BookStoreEntities db = new BookStoreEntities())
    {
        User newUser = new User() { Login = txtBox_login.Text, Password =
txtBox_password.Text };
        db.Users.Add(newUser);
        db.SaveChanges();
        MessageBox.Show("Created new user");
    }
} //End
}
else
{
    if (!Regex.Match(input_password).Success)
    {
        txtBox_password.BackColor = Color.PaleVioletRed;
    }
    if (!Regex.Match(input_login).Success)
    {
        txtBox_login.BackColor = Color.PaleVioletRed;
    }
}
}

private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    if (signInForm)
    {
        RegistrarionState();
    }
}

```

```

        signInForm = false;
    }
    else
    {
        SignInState();
        signInForm = true;
    }
}
public void RegistrarionState()
{
    txtBox_login.Text = "";
    txtBox_password.Text = "";

    this.Text = "Create account";
    label3.Text = "Have account?";
    linkLabel1.Text = "Sign In!";

    btn_SignIn.Text = "Create account";
    btn_SignIn.Width = 158;
    btn_SignIn.Height = 32;
}
public void SignInState()
{
    txtBox_login.Text = "";
    txtBox_password.Text = "";

    this.Text = "Sign In";
    label3.Text = "No account?";
    linkLabel1.Text = "Create one!";

    btn_SignIn.Text = "Sign in";
    btn_SignIn.Width = 111;
    btn_SignIn.Height = 32;
}
}
}

```

Файл «EditForm.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Linq;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace ShopStore
{
    public partial class EditForm : Form
    {
        BookStoreEntities db = null;
        public Book editBook = null;
        public EditForm(Book editObj)
        {
            InitializeComponent();
            db = new BookStoreEntities();

            editBook = editObj;

            txtBox_Fio.Text = editBook.Fio;
            txtBox_PublishName.Text = editBook.PublishName;
            txtBox_BookName.Text = editBook.NameBook;
            txtBox_Price.Text = editBook.Price.ToString();
            txtBox_Pages.Text = editBook.Pages.ToString();
            txtBox_SalePrice.Text = editBook.SalePrice.ToString();
            dateTimePicker.Value = editBook.DateOfPublishing;
        }
        private void btn_Edit_Click_1(object sender, EventArgs e)
        {
            editBook.DateOfPublishing = dateTimePicker.Value.Date;
            editBook.NameBook = txtBox_BookName.Text;
            editBook.Pages = Convert.ToInt32(txtBox_Pages.Text);
            editBook.Price = Convert.ToDecimal(txtBox_Price.Text);
            editBook.SalePrice = Convert.ToDecimal(txtBox_SalePrice.Text);
            editBook.PublishName = txtBox_PublishName.Text;
            editBook.Fio = txtBox_Fio.Text;

            this.DialogResult = DialogResult.OK;
        }
    }
}

```

Файл «AddForm.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Linq;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace ShopStore
{
    public partial class AddForm : Form
    {
        BookStoreEntities db = null;

        public AddForm()
        {
            InitializeComponent();
            db = new BookStoreEntities();
        }

        private void btn_Add_Click(object sender, EventArgs e)
        {
            Book new_bok = new Book()
            {
                Fio = txtBox_Fio.Text,
                PublishName = txtBox_PublishName.Text,
                NameBook = txtBox_BookName.Text,
                Price = Convert.ToInt32(txtBox_Price.Text),
                Pages = Convert.ToInt32(txtBox_Pages.Text),
                SalePrice = Convert.ToInt32(txtBox_SalePrice.Text),
                DateOfPublishing = dateTimePicker.Value.Date
            };
            db.Books.Add(new_bok);
            db.SaveChanges();
            this.DialogResult = DialogResult.OK;
        }
    }
}

```

ПРИЛОЖЕНИЕ Б

(обязательное)

Интеллектуальная система для обработки, хранения и передачи
данных

Презентация