

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Рецензент, инженер-программист

_____ / А.А. Лебедев /

« _____ » _____ 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой, д.т.н., с.н.с.

_____ / Б.М. Суховилов /

« _____ » _____ 2020 г.

Разработка программы «Бонусная система + товары»
для магазина видеоигр Sotcomm

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.03.2020.331.ВКР

Руководитель, к.т.н., доцент

_____ / В.А. Конов /

« _____ » _____ 2020 г.

Автор

студент группы ЭУ – 462

_____ / С.Г. Терсков /

« _____ » _____ 2020 г.

Нормоконтролер, доцент

/ Е.А. Конова /

« _____ » _____ 2020 г.

Челябинск 2020

АННОТАЦИЯ

Терсков С.Г. Разработка программы «Бонусная система + товары» для магазина видеоигр Sotcomm – Челябинск: ЮУрГУ, ЭУ-462, 37с., 25ил., 14табл., библиогр. список –бнаим., 3прил.

Выпускная квалификационная работа выполнена с целью разработки программы «Бонусная система + товары» для магазина видеоигр Sotcomm.

В ходе работы проанализирована работа сети магазинов, а именно как происходит продажа товаров, обмен видеоигр, приход товара и работа системы лояльности постоянных клиентов.

С помощью методологии DFD создана графическая модель деятельности сети магазинов, подробно описаны процессы, выполняемые работниками данной сети.

В результате работы автоматизирована работа сети магазинов видеоигр Sotcomm. Реализована работа со справочными материалами для работы магазина. Автоматизирован процесс продажи товаров по торговым точкам.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1 ПОСТАНОВКА ЗАДАЧИ	8
Вывод по разделу один	9
2 АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ	10
2.1 «1С:Розница 8»	10
2.2 «1С:Торговля и Склад»	11
2.3 «Мой склад»	13
2.4 Сравнительный анализ систем	13
Выводы по разделу два	14
3 СТРУКТУРА ПРОГРАММЫ	15
3.1 Блок «Бонусные карты»	16
3.2 Блок «Заказы»	17
3.3 Блок «Товары»	18
Выводы по разделу три	19
4 СТРУКТУРА БАЗЫ ДАННЫХ	20
Выводы по разделу четыре	23
5 РАЗРАБОТКА ПРОГРАММЫ	24
5.1 Подключение базы данных к программе	24
5.2 Форма авторизации пользователей	25
5.3 Форма «Меню»	26
5.4 Разработка форм для работ с данными	27
5.5 Товары	28
5.6 Форма редактирования товара	29
5.7 Форма «Бонусы»	29
5.8 Форма «Покупки»	30
5.9 Форма «Отчёт»	30
5.10 Форма «Ввод покупки»	32
5.11 Форма «Обмен»	33
5.12 Форма «Прокат»	34
5.13 Форма «Действующие прокаты»	35
Выводы по разделу пять	35
6 ЭКОНОМИЧЕСКАЯ ЧАСТЬ	36
Выводы по разделу шесть	37
ЗАКЛЮЧЕНИЕ	38
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	39
ПРИЛОЖЕНИЕ А Листинг кода формы «Ввод покупки»	40
ПРИЛОЖЕНИЕ Б Листинг кода формы «Обмен»	46
ПРИЛОЖЕНИЕ В Листинг кода формы «Бонусы»	49

ВВЕДЕНИЕ

Некоторые магазины до сих пор не автоматизировали процессы учета товаров, проведения продаж и хранения базы данных постоянных клиентов. В настоящее время остро стоит проблема автоматизация работы развивающихся магазинов.

Существует множество программ, автоматизирующие процесс продаж и работу сети магазинов в целом. Они имеют обширный функционал, удобный интерфейс и большое количество документации для отчётности. Однако, такие программы требуют высокую цену как за разработку, так и за поддержку, что является невыгодным для развивающихся сетей.

Как правило, такие сети выбирают узконаправленные решения, которые разрабатываются под их требования. Это выгодно как для руководителя сети, так и для разработчика программы. Заказчик даёт чёткий список требований к разработке, что значительно упрощает работу разработчика.

Целью дипломного проекта является автоматизация процесса продаж, учета товаров и реализация бонусной системы сети магазинов видеоигр Sotcomm.

Основные задачи дипломного проекта следующие.

1. Определить требования к разработке.
2. Сравнить существующие системы автоматизации продаж и определить целесообразность разработки программы.
3. Определить структуру программы и её основной функционал.
4. Создать и нормализовать базу данных для хранения товаров и бонусных карт, заполнить её данными.
5. Разработать программу для работы с базой данных.
6. Реализовать возможность редактирования данных в программе.
7. Реализовать многопользовательский режим работы – разграничение прав пользователей.

1 ПОСТАНОВКА ЗАДАЧИ

Для автоматизации работы сети магазинов видеоигр и развлекательных товаров необходимо разработать программу, в которой хранится информация о товарах, накопительных картах, а также заказах, поступивших от клиентов. Для удобства работы продавцов необходимо реализовать ежедневную отчетность.

Основная часть программы рассчитана на хранение базы данных клиентов и прикрепленных к ним бонусных карт. В них содержится вся информация об их покупках и подсчет накопленных бонусов. Бонусная программа работает следующим образом – клиенту начисляется 7% от суммы с каждой покупки. Покупатель может оплатить до 50% суммы следующей покупки доступными бонусами.

В базе данных программы хранится следующее.

1. Информация о бонусных картах (номер карты, количество бонусов).
2. Информация о размещенных заказах (дата размещения, наименование товара, номер клиента).
3. Информация о товарах (наименование товара, стоимость, количество, торговая точка).

Пользователи программы – менеджер и продавец. Менеджер обладает правами администратора. Эти права заключаются во внесении, редактировании и удалении всей информации о товарах.

Продавец вносит информацию о бонусных картах, размещает заказы клиентов, а также ведет учёт товаров, вносит информацию о продажах в соответствующий отчет.

Целесообразность создания программы заключается в отсутствии ведения в магазине учёта товаров, а также неудобной реализации хранения бонусных карт.

На данный момент хранение бонусных карт реализуется в программе MS OfficeExcel. На рисунке 1 представлен пример хранения бонусной карты.

	A	B	C	D	E	F	G	H
1	Дата продажи	Наименование	Описание	Сумма	Процент	Расход	Итого	
2	25.11.19	GTA5 хопе б/у		1 790,00 Р	125,30 Р		125,30 Р	89821021218
3					- Р			
4					- Р			
5					- Р			
6					Р			

89193221396 Лист1 89080605204 89127740673 89514875714 **89821021218** 89124768054 ...

Рисунок 1–Существующая реализация хранения бонусных карт

Каждая торговая точка имеет свой файл с данными о картах. В этих файлах номера карт хранятся в отдельных листах. Поэтому, из-за большого количества карт, поиск по номеру сильно усложняется. Также влияет низкая скорость работы ноутбуков, которыми пользуются продавцы в торговых точках, потому что они не рассчитаны на обработку большого объема памяти. Решением данной проблемы станет хранение и обработка данных на сервере.

Вывод по разделу один

Выполнена постановка задачи. Определены основные требования к разработке программы. Описаны пользователи программы и какие функции они выполняют.

2 АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ

2.1 «1С:Розница 8»

Универсальное решение для управления розничной торговлей [1].

Оно полностью автоматизирует все основные бизнес-процессы как отдельного магазина, так и крупной розничной сети, позволяет организовать эффективное управление продажами и закупками, запасами и складом, персоналом магазина, ассортиментом и ценообразованием, маркетинговыми акциями и системами лояльности.

Функционал «1С:Розница 8» может настраиваться в соответствии с принятой на конкретном предприятии методикой управления ассортиментом и ценообразованием, системой скидок, способом оформления складских операций и розничных продаж и другими организационными и технологическими особенностями работы магазинов. Профили пользователей, соответствующие типичным должностям сотрудников магазина, позволяют быстро начать работу без предварительной настройки.

«1С:Розница» обладает следующими возможностями.

1. Работа с нормативно-справочной информацией:

- a. удобные справочники для работы с номенклатурой и ценами, оформления операций, централизованного ввода и обновления данных;
- b. управление ассортиментом и ценообразованием;
- c. анализ покупательского спроса и ликвидности товаров, определение потребностей в товарах, формирование оптимального ассортимента;
- d. управление ценообразованием с учетом формата, местоположения и целевой группы покупателей каждого конкретного магазина, отложенное применение цен.

2. Управление программами лояльности: поддержка более 20 условий предоставления скидок, как в денежном выражении, так и в виде подарков. Учет и контроль продаж и использования подарочных сертификатов.

3. Управление складом:

а. оперативный учет товаров по складам хранения и отделам магазина, в том числе серийный учет по срокам годности;

б. комплектация и разуконплектация товаров, фасовка весовой продукции с функцией печати этикетки, настройка схем отбора товаров для выборочных инвентаризаций.

4. Рабочее место кассира:

а. специализированный интерфейс рабочего места кассира, который поддерживает более 40 команд для оформления розничных продаж с учетом специфики разных отраслей;

б. позволяет принимать различные виды оплаты, оформлять скидки, принимать возвраты, вести учет личных продаж продавца, подбирать товары разными способами и т. д.

2.2 «1С:Торговля и Склад»

Программа «Торговля и Склад» предназначена для учета любых видов торговых операций. Благодаря гибкости и настраиваемости, конфигурация способна выполнять все функции учета – от ведения справочников и ввода первичных документов до получения различных аналитических отчетов[2].

Функциональные возможности конфигурации:

1. Складской учет:

а. учет различных видов товарно-материальных ценностей (ТМЦ): товар, материал, продукция, тара;

б. учет ТМЦ на множестве складов;

с. учет ТМЦ в различных единицах измерения (упаковках);

д. партионный учет по себестоимости каждой единицы (возможность выбора партии списания);

е. возможность ведения партионного учета в разрезе материально ответственных лиц (кладовщиков);

f. оформление различных складских операций: перемещение, списание, оприходование, инвентаризация с выпиской всех необходимых документов;

g. возможность произвольного оприходования и списания ТМЦ по различным статьям затрат (в том числе реклама, подарки и т.д.);

h. учет дополнительных расходов, включаемых в себестоимость приобретаемых ТМЦ;

i. получение отчетной информации по наличию и движению ТМЦ, а также оборотных ведомостей и товарного отчета;

j. получение отчетной информации в разрезе свойств номенклатурных единиц (цвета, размера, срока хранения и т.д.).

2. Гибкий механизм ценообразования:

a. неограниченное количество различных типов цен (оптовая, розничная, мелкооптовая и т.д.);

b. возможность задания цен в любой валюте с учетом или без учета НДС и НП для выбранной упаковки ТМЦ;

c. автоматический расчет цен на основании цены другого типа и заданной торговой наценки;

d. задание индивидуальных цен и скидок по каждому договору;

e. обновление цен поставщиков из документа, сопровождающего получение ТМЦ;

f. автоматизация оптовой торговли.

3. Автоматизация розничной торговли:

a. оформление розничных продаж с применением или без применения ККМ;

b. учет остатков ТМЦ, переданных в розницу, по розничным ценам;

c. переоценка остатков ТМЦ в рознице;

d. печать ценников и этикеток;

е. взаимодействие с различными типами ККМ в режимах: ONLINE, OFFLINE, фискальный регистратор;

ф. работа с дополнительным торговым оборудованием: штрих-сканнерами, терминалами сбора данных, принтерами этикеток, электронными весами.

Программа «1С:Торговля и Склад» распространяется через сеть партнерских организаций в странах СНГ и Балтии.

2.3 «Мой склад»

Программа «Мой склад» – коммерческий SaaS продукт для управления торговлей и складского учёта, предназначенный для автоматизации малого и среднего бизнеса [3]. Сервис реализует функции обработки заказов, управления продажами и закупками, складского учёта и контроля финансовых расчетов. МойСклад стал одним из первых сервисов SaaS, разработанных в России и предназначенных для российского рынка.

Функциональные возможности программы:

- оптовые продажи и электронная коммерция;
- розничная торговля;
- закупки и складской учет;
- crm, или работа с клиентской базой;

2.4 Сравнительный анализ систем

Сравнительный анализ систем представлен в таблице 2.1

Таблица 2.1 – Сравнительный анализ систем

Критерии	«1С:Розница 8»	«1С:Торговля и Склад»	«Мой склад»	Дипломный проект
Отсутствие излишней функциональности	–	–	+	+
Ведение складского учета	+	+	+	+

Критерии	«1С:Розница 8»	«1С:Торговля и Склад»	«Мой склад»	Дипломный проект
Реализация системы с бонусными картами	–	–	–	+
Дешевизна программы	–	–	–	+
Простота интерфейса пользователя	–	+	–	+
Узконаправленное решение	–	–	+	+
Возможность доработки программы	–	–	–	+
Функция обмена товаров	–	–	–	+

Таким образом, исходя из данных сравнительного анализа, можно сделать вывод о том, что целесообразность разработки программы для дипломного проекта оправдана по следующим причинам:

- программа является узконаправленным решением;
- в программе реализована система с бонусными картами (лояльность клиентов);
- реализован простой и понятный пользователю интерфейс;
- предусмотрена возможность доработки программы в случае необходимости;
- оптимально низкая цена программы, по сравнению с обозреваемыми системами.

Выводы по разделу два

Проведен сравнительный анализ существующих систем. Выделены конкурентные преимущества дипломного проекта и определена целесообразность разработки программы.

3 СТРУКТУРА ПРОГРАММЫ

Структура программы построена по методологии DFDc помощью построителя диаграмм ErwinProcessModeler.

На рисунке 2 показаны основные пользователи программы.

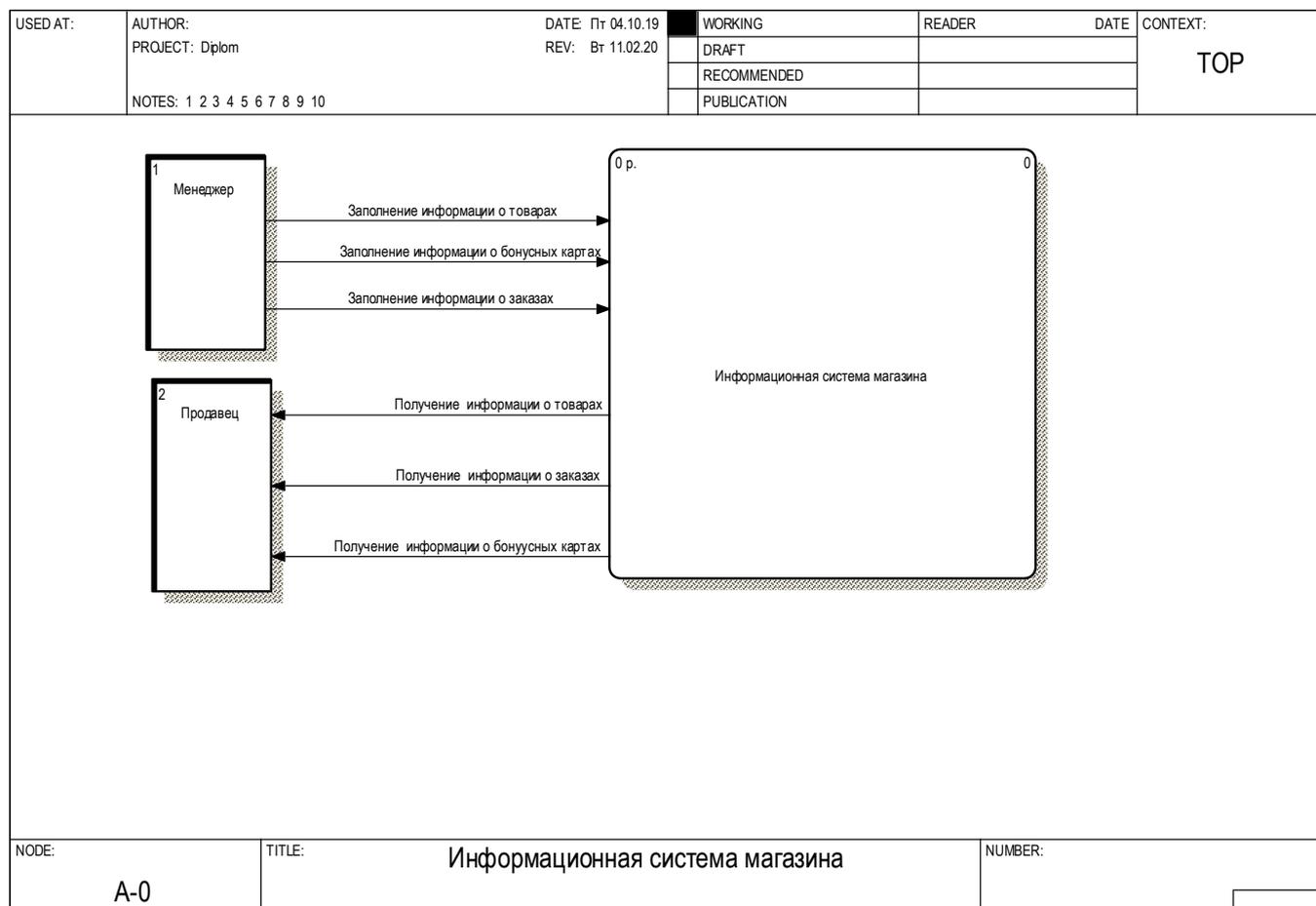


Рисунок 2 – Блок «Информационная система магазина»

Менеджер заполняет справочную информацию о товарах и бонусных картах, а также заполняет информацию о заказах на товары. В программе он выполняет роль администратора, т.е. он может добавлять, изменять и удалять информацию в базе данных.

Продавец занимается внесением продаж товаров. Он может получать справочную информацию и информацию о заказах, чтобы сформировать его в магазине и отдать покупателю.

На рисунке 3 изображена подробная структура программы.

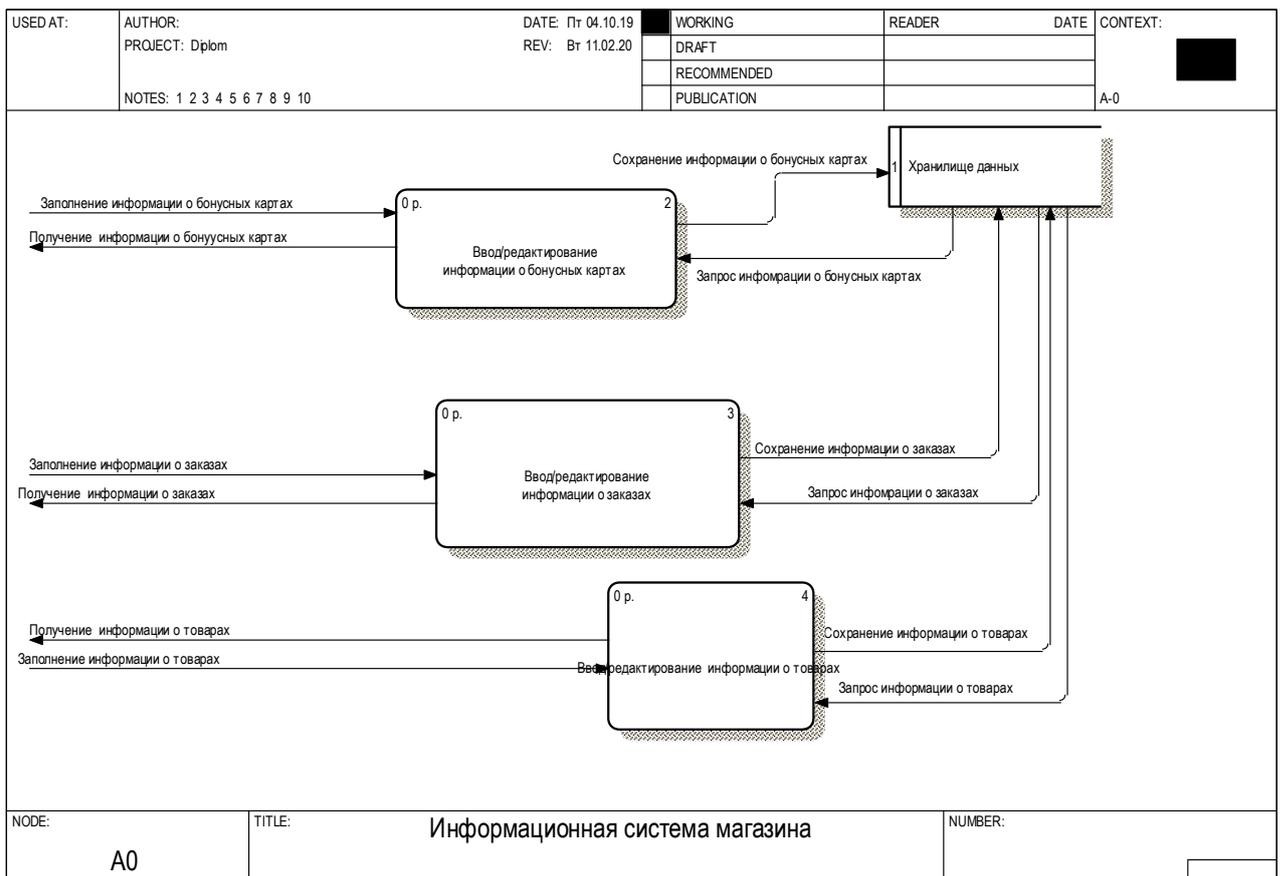


Рисунок 3 – Диаграмма структуры программы

Структура поделена на три блока: бонусные карты, заказы и товары. На диаграмме показано, как каждый блок взаимодействует с базой данных и каким образом происходит заполнение данных.

3.1 Блок «Бонусные карты»

На рисунке 4 изображена диаграмма «Ввод/редактирование информации о бонусных картах».

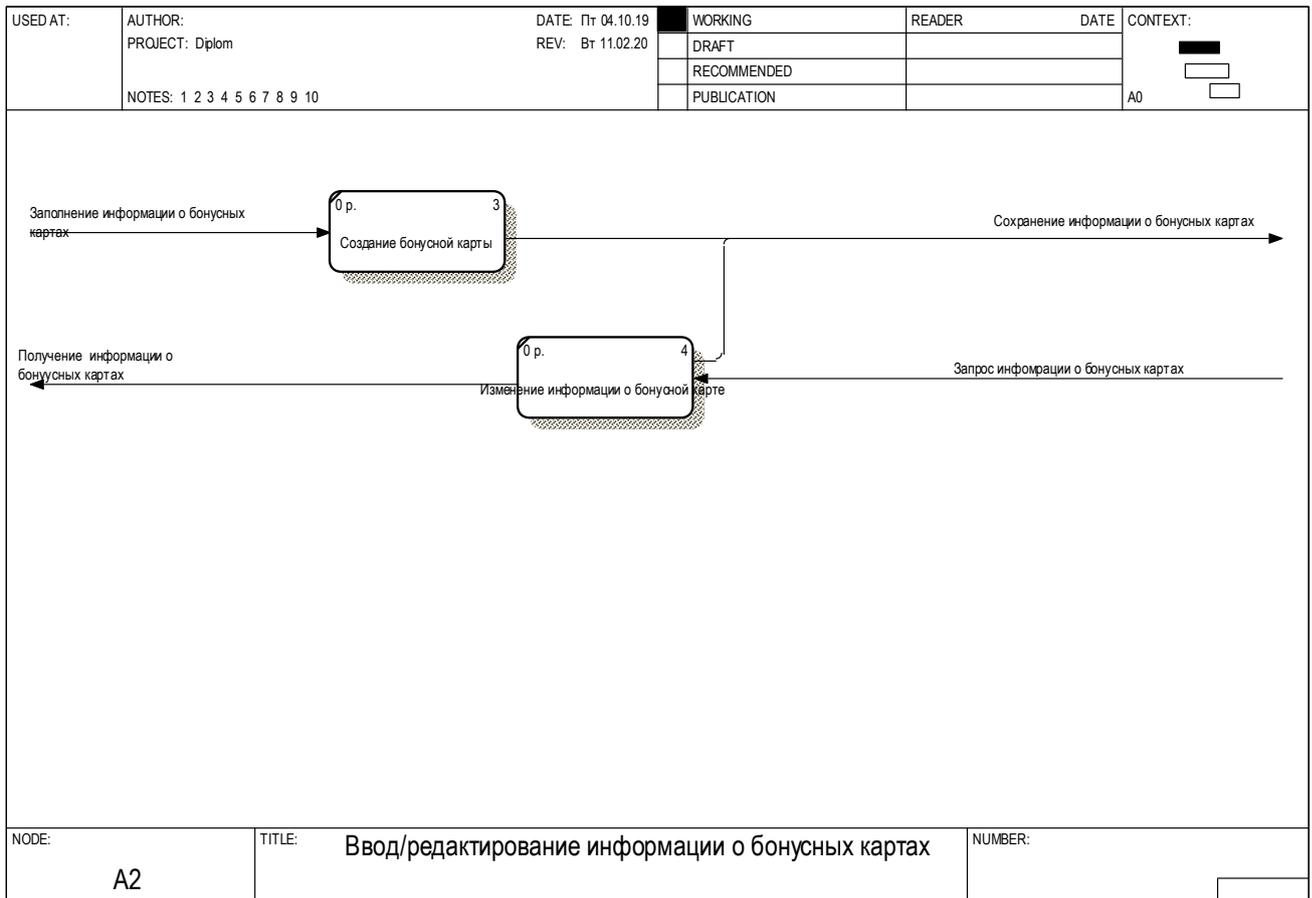


Рисунок 4 – Блок «Ввод/редактирование информации о бонусных картах»

В диаграмме показаны основные процессы, происходящие с бонусными картами – создание карты и изменение информации о ней.

3.2 Блок «Заказы»

На рисунке 5 изображена диаграмма «Ввод/редактирование информации о заказах».

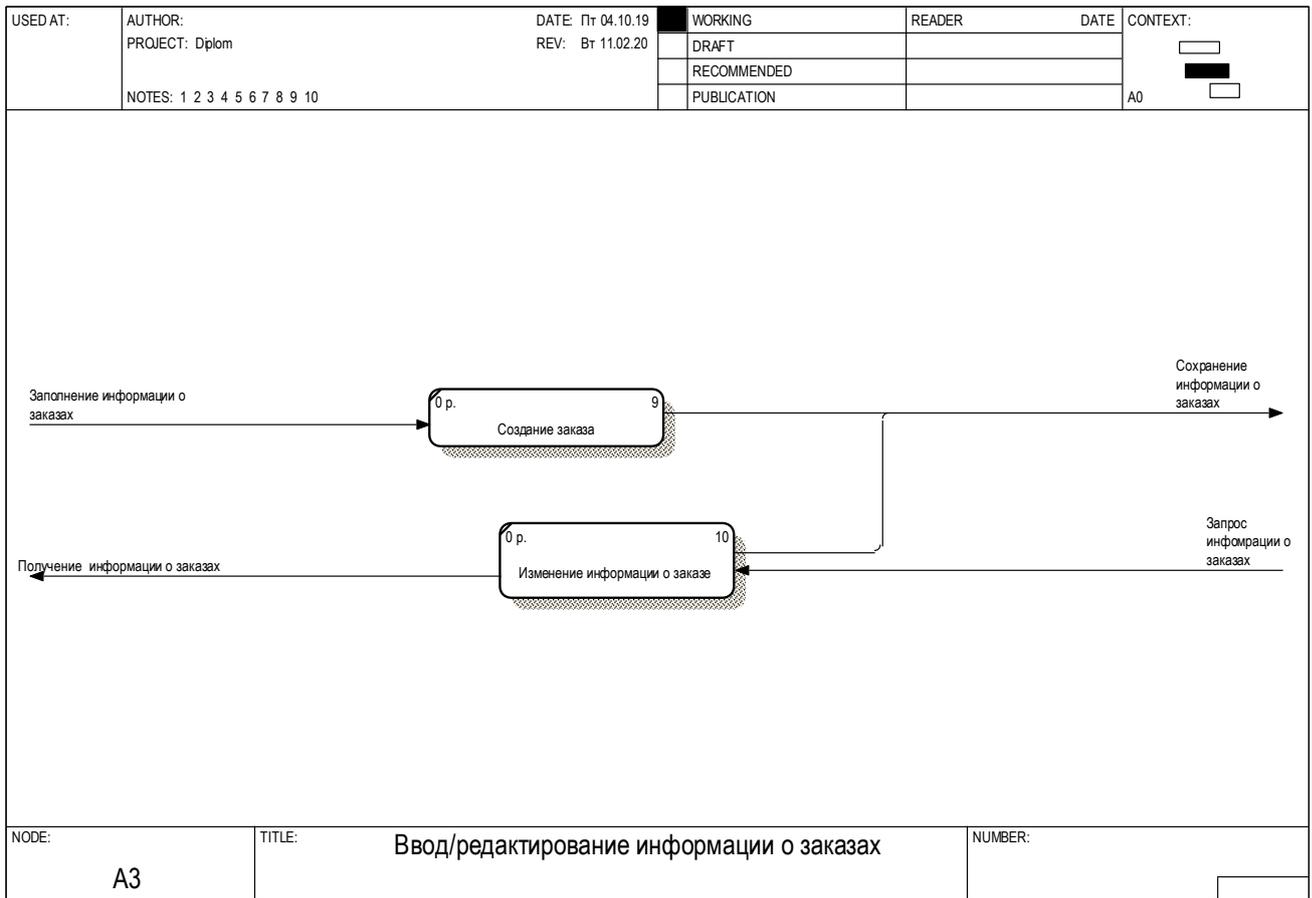


Рисунок 5 – Диаграмма «Ввод/редактирование информации о заказах»

В диаграмме показан процесс создания заказа и редактирования информации о заказе.

3.3 Блок «Товары»

На рисунке 6 изображена диаграмма «Ввод/редактирование информации о товарах».

4 СТРУКТУРА БАЗЫ ДАННЫХ

База данных разработана в phpMyAdmin[4].

Схема данных представлена на рисунке 7.

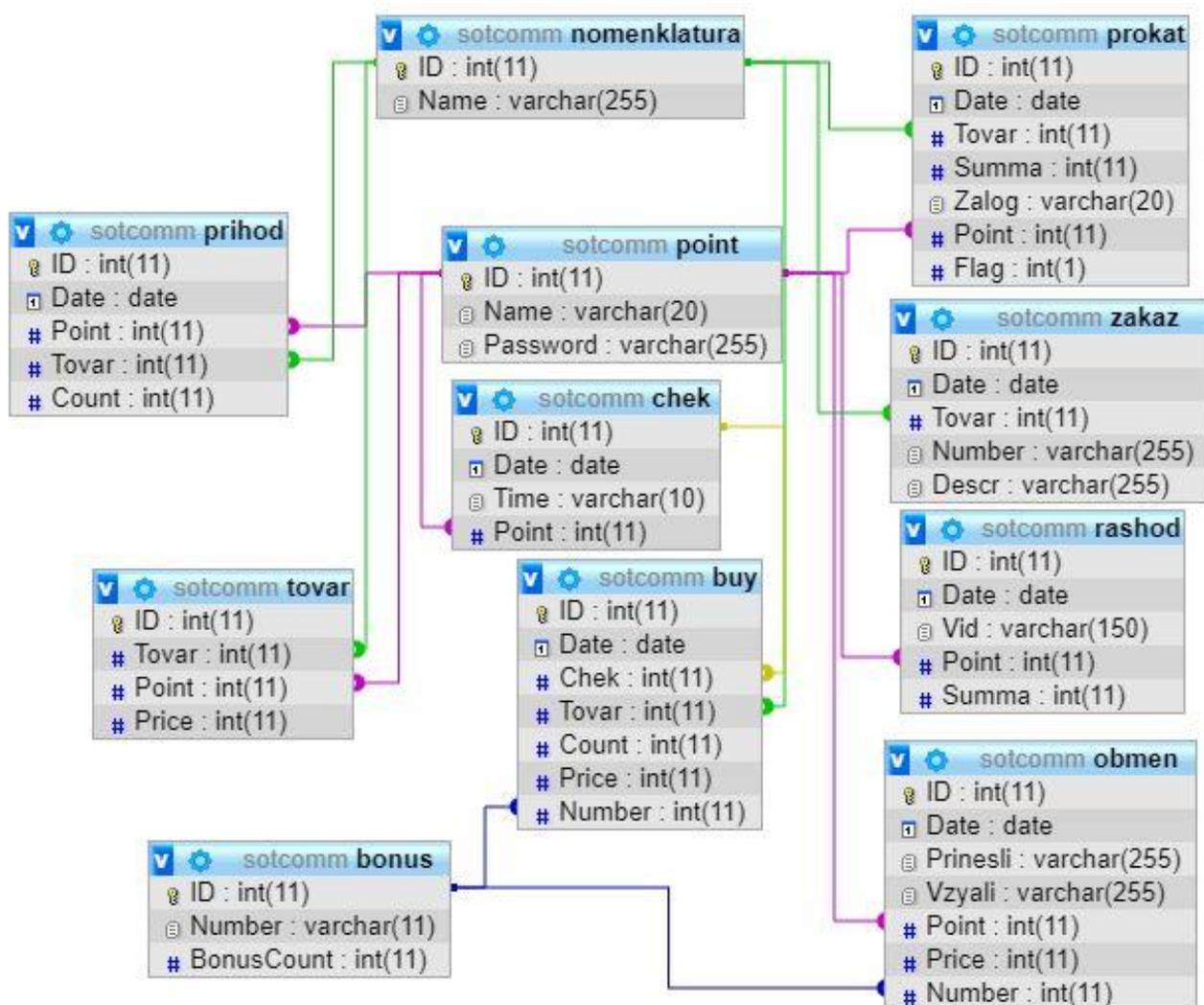


Рисунок 7 – Схема базы данных

Таблицы являются объектами, которые содержат все данные в базах данных. В таблицах данные логически организованы в виде строк и столбцов по аналогии с электронной таблицей[5].

bonus – таблица для бонусных карт. Структура таблицы описана в таблице 4.1

Таблица 4.1 – Структура таблицы bonus

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Number	varchar(11)	Номер бонусной карты
BonusCount	Int32	Количество бонусов

buy – таблица для покупок. Структура таблицы описана в таблице 4.2

Таблица 4.2 – Структура таблицы buy

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Датапокупки
Chek	Int32	Код чека
Tovar	Int32	Код товара
Count	Int32	Количество товара
Price	Int32	Сумма покупки
Number	Int32	Код бонусной карты

chek – таблица для чеков. Структура таблицы описана в таблице 4.3

Таблица 4.3 – Структура таблицы chek

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Дата создания чека
Time	varchar(11)	Время создания чека
Tovar	Int32	Код товара
Point	varchar(11)	Точка,на которой совершена покупка

point – таблицаторговых точек и паролей входа в программу. Структура таблицы описана в таблице 4.4

Таблица 4.4 – Структура таблицы point

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Login	varchar(64)	Логин
Password	varchar(64)	Пароль

obmen – таблица для чеков. Структура таблицы описана в таблице 4.5

Таблица 4.5 – Структура таблицы obmen

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Дата обмена
Prinesli	varchar(255)	Принесенный товар
Vzyali	varchar(255)	Взятый товар

Название столбца	Тип данных столбца	Описание столбца
Point	varchar(11)	Точка, на которой обмен
Price	Int32	Сумма обмена
Number	Int32	Код бонусной карты

prokat – таблица для прокатов. Структура таблицы описана в таблице 4.6

Таблица 4.6 – Структура таблицы prokat

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Дата проката
Tovar	Int32	Код товара, взятого в прокат
Summa	Int32	Сумма проката
Zalog	varchar(20)	Залог за товар
Point	varchar(100)	Код точки, на которой совершён прокат
Flag	Int32	Состояние проката

rashod – таблица для расходов. Структура таблицы описана в таблице 4.7

Таблица 4.7 – Структура таблицы rashod

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Дата совершения расхода
Vid	varchar(150)	Вид расхода
Point	varchar(20)	Код точки, на которой совершён расход
Summa	Int32	Сумма расхода

tovari – таблица для распределения товаров по точкам. Структура таблицы описана в таблице 4.8

Таблица 4.8 – Структура таблицы tovari

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Name	varchar(255)	Наименование товара
Point	Int32	Код точки
Price	Int32	Стоимость товара

zakaz – таблица для заказов, оставленных клиентами. Структура таблицы описана в таблице 4.9

Таблица 4.9 – Структура таблицы zakaz

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Дата размещения заказа
Tovar	Int32	Код товара
Number	varchar(255)	Номер для связи с клиентом
Descr	varchar(255)	Описание заказа

nomenklatura – таблица для хранения наименований товаров. Структура таблицы описана в таблице 4.10

Таблица 4.10 – Структура таблицы nomenklatura

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Name	varchar(255)	Наименование товара

prihod – таблица ввода поступившего товара на точку. Структура таблицы описана в таблице 4.11

Таблица 4.11 – Структура таблицы prihod

Название столбца	Тип данных столбца	Описание столбца
ID	PRIMARYKEY, int	Ключевой столбец таблицы
Date	Date	Дата прихода
Point	Int32	Код точки
Tovar	Int32	Код товара
Count	Int32	Количество поступившего товара

Выводы по разделу четыре

Разработана база данных для работы с программой. Описаны структуры таблиц базоданных.

5 РАЗРАБОТКА ПРОГРАММЫ

Программа разработана с помощью среды разработки Microsoft Visual Studio 2017 Community. Язык программирования – C#.

Разработка программы разделена на несколько этапов.

1. Подключение базы данных к программе.
2. Создание формы для авторизации пользователей, написание кода для событий формы, обработка авторизации пользователей.
3. Создание форм для работы с программой.
4. Разработка форм – заполнение элементами управления, обработка событий форм и элементов, написание кода.
5. Отладка программы.

5.1 Подключение базы данных к программе

Для подключения базы данных MySQL, необходимо установить программное обеспечение в виде MySQL for Visual Studio и MySQL Connector [6].

В обозревателе серверов подключена база данных, созданная в веб-приложении phpMyAdmin (рисунок 8).

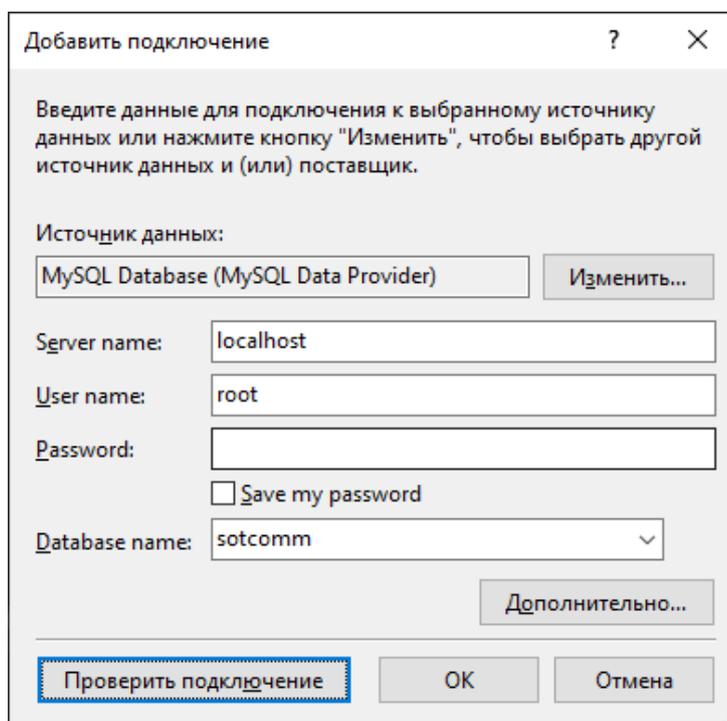


Рисунок 8 – Подключение базы данных в Visual Studio

После этой операции стала доступна строка подключения, по которой происходит соединение с базой данных (рисунок 9).

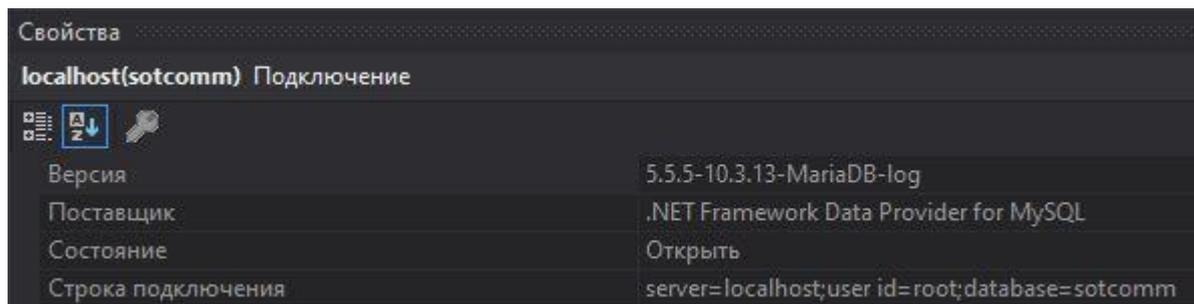


Рисунок 9 – строка подключения

5.2 Форма авторизации пользователей

Форма авторизации пользователей (рисунок 10) запускается при открытии программы.

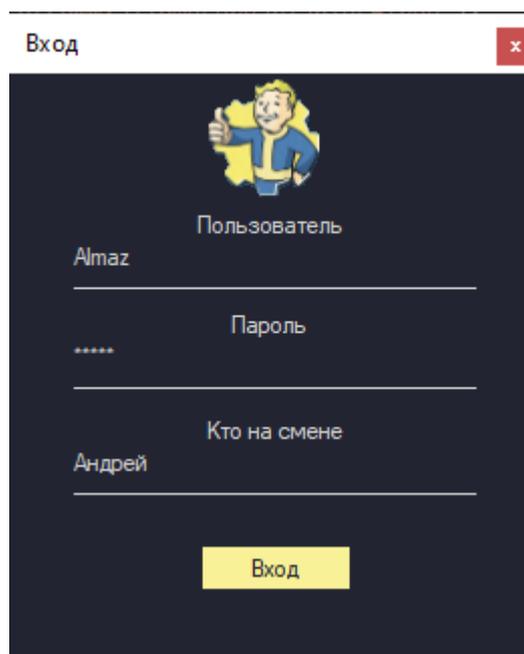


Рисунок 10 – форма авторизации пользователей

Текстовое поле «Пароль» использует свойство PasswordChar, которое скрывает содержимое поля символом «*».

Чтобы идентифицировать пользователя, создаётся запрос, проверяющий введённые данные. Кодобработки входа приведен в листинге 1.

Листинг 1 – Код входа

```
Inti = 0;  
G.con.Open();
```

```

MySQLCommandcmd = G.con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select * from login where Login = '" + textBox1.Text + "' and
password = '" + textBox2.Text + "'";
cmd.ExecuteNonQuery();
DataTable dt = newDataTable();
MySQLDataAdapter da = newMySQLDataAdapter(cmd);
da.Fill(dt);
i = Convert.ToInt32(dt.Rows.Count.ToString());
G.smena = textBox3.Text.ToString();
if (i == 0)
{
label3.Text = "Неверное имя пользователя или пароль";
textBox2.Clear();
textBox1.Clear();
}
else
{
G.login = textBox1.Text;
this.Hide();
MainForm mf = newMainForm();
mf.Show();
}
G.con.Close();

```

Если вход прошел успешно, т.е. пользователь ввел верный пароль, откроется форма «Меню». Если пользователь закроет окно, программа закроется полностью.

Текущий пользователь сохраняется в поле login класса G, в котором хранятся данные для перехода из формы в форму.

При нажатии кнопки «Enter», происходят аналогичные действия, как и при нажатии на кнопку «Вход».

5.3 Форма «Меню»

Меню (рисунок 11) предназначено для открытия форм, в которых производится работа с базой данных.

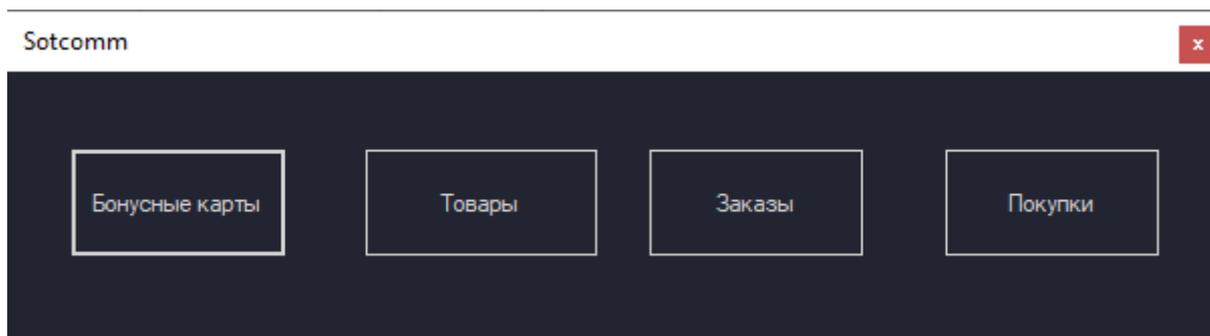


Рисунок 11–форма «Меню»

Если пользователь закроет окно, программа закроется полностью.

5.4 Разработка форм для работ с данными

Форма «Заказы» представляет собой список заказов, размещённых клиентами.

Источник данных для `datagridview` – таблица БД `zakaz`. Столбец ID скрыт от пользователей. На форме(рисунок 12) расположены поля для добавления строки в таблицу БД `zakaz` по нажатию кнопки «Добавить». Кнопка «Удалить выбранный» удаляет выделенную в `datagridview` запись. Код представлен в приложении

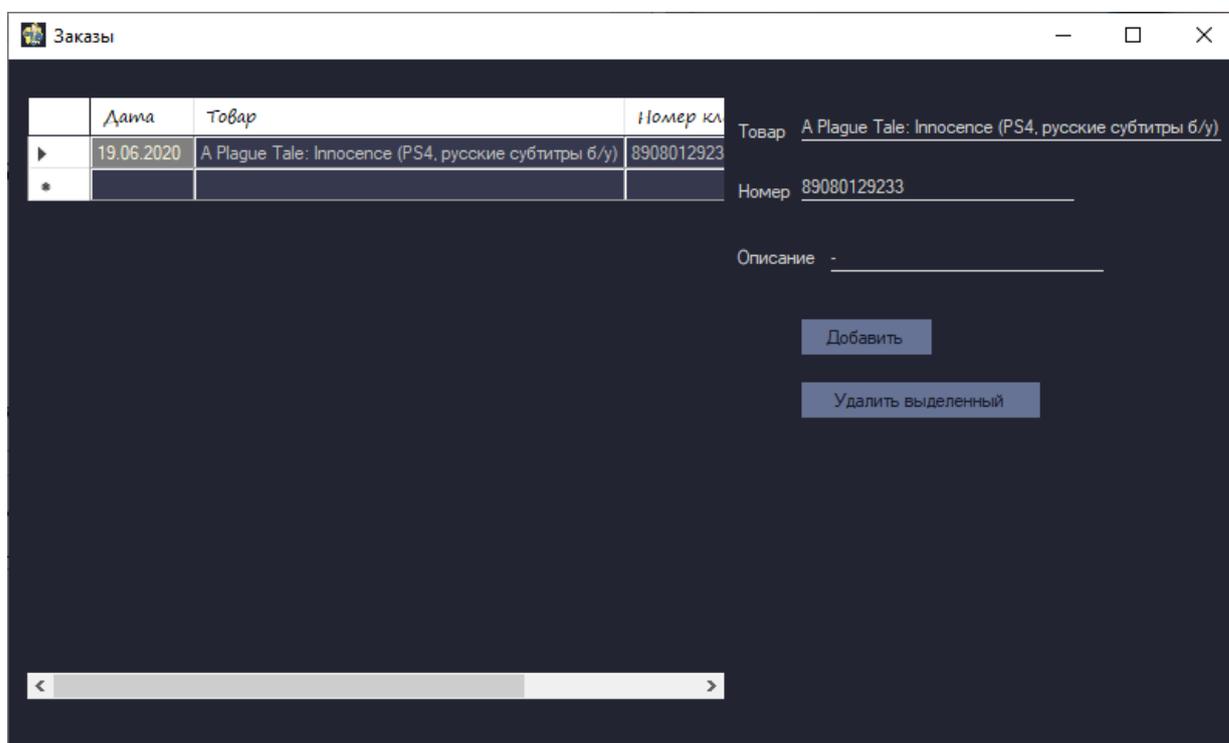
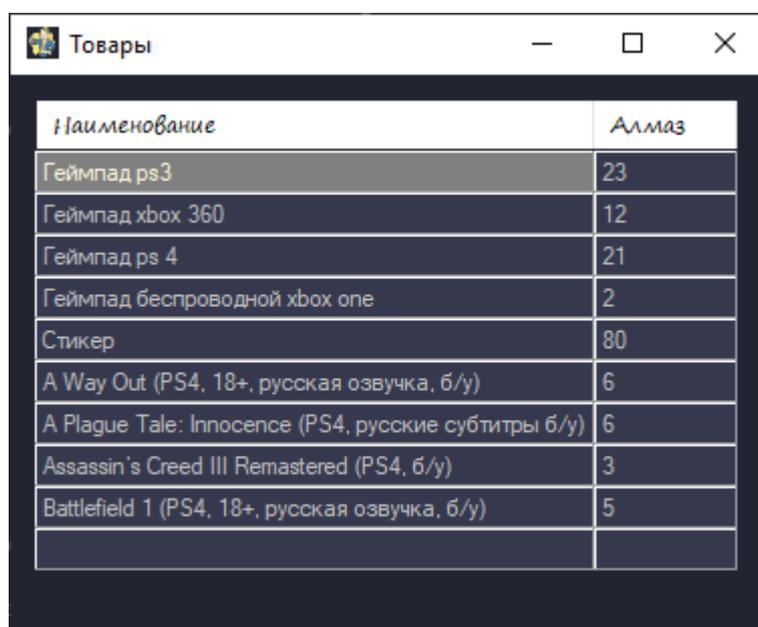


Рисунок 12 – форма «Заказы»

5.5 Товары

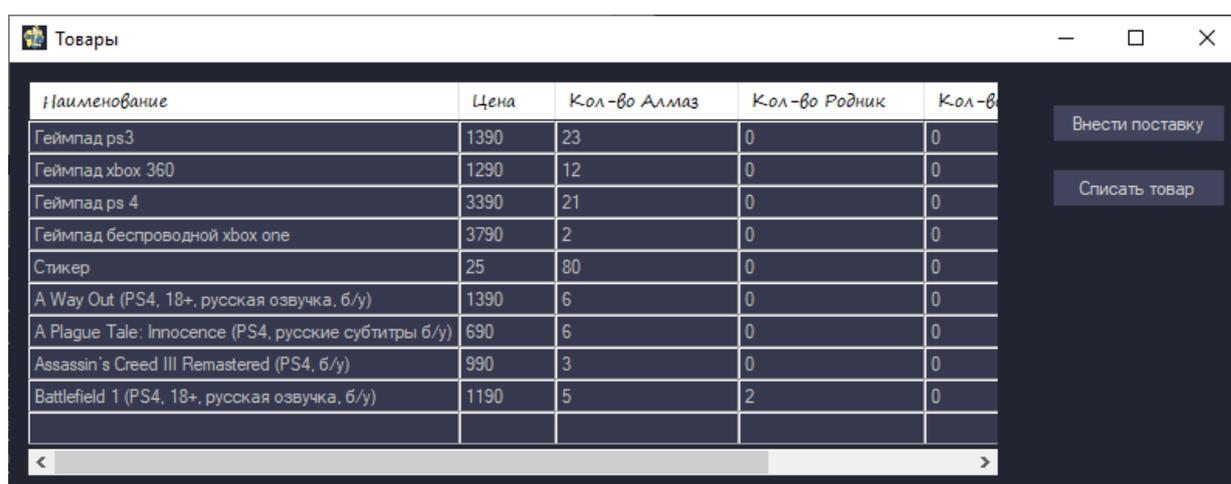
Форма «Товары» предназначена для отображения данных в datagridview из таблицы БД `товари`. Продавец отслеживает количество товара на его точке (рисунок 13).



Наименование	Алмаз
Геймпад ps3	23
Геймпад xbox 360	12
Геймпад ps 4	21
Геймпад беспроводной xbox one	2
Стикер	80
A Way Out (PS4, 18+, русская озвучка, б/у)	6
A Plague Tale: Innocence (PS4, русские субтитры б/у)	6
Assassin's Creed III Remastered (PS4, б/у)	3
Battlefield 1 (PS4, 18+, русская озвучка, б/у)	5

Рисунок 13 – форма «Товары» отображение для продавцов

Администратор отслеживает количество товаров на всех точках (рисунок 14) и имеет права вносить поставки, списывать товар и менять стоимость товаров.



Наименование	Цена	Кол-во Алмаз	Кол-во Родник	Кол-во
Геймпад ps3	1390	23	0	0
Геймпад xbox 360	1290	12	0	0
Геймпад ps 4	3390	21	0	0
Геймпад беспроводной xbox one	3790	2	0	0
Стикер	25	80	0	0
A Way Out (PS4, 18+, русская озвучка, б/у)	1390	6	0	0
A Plague Tale: Innocence (PS4, русские субтитры б/у)	690	6	0	0
Assassin's Creed III Remastered (PS4, б/у)	990	3	0	0
Battlefield 1 (PS4, 18+, русская озвучка, б/у)	1190	5	2	0

Внести поставку
Списать товар

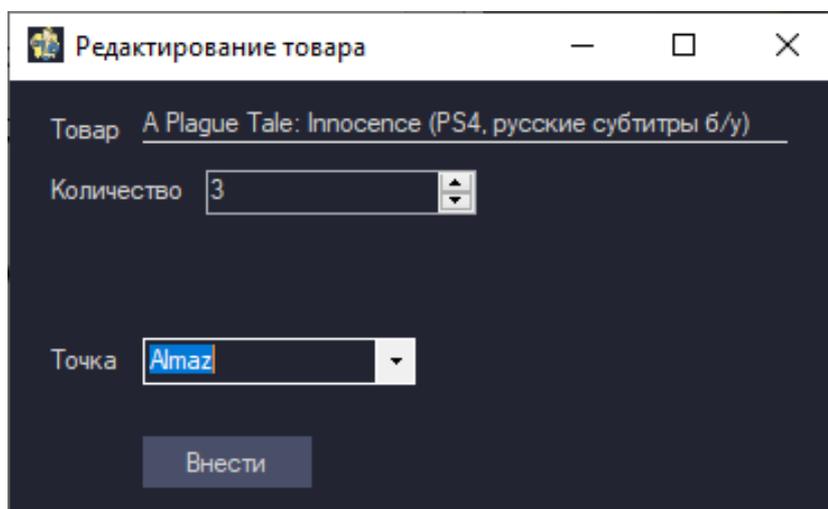
Рисунок 14 – форма «Товары» отображение для администратора

При нажатии одной из кнопок, открывается форма для работы с количеством товара.

5.6 Форма редактирования товара

Поле «Товар» содержит коллекцию наименований товаров из таблицы БД `tovari`. При вводе в поле, появляется отфильтрованный выпадающий список. Количество товара регулируется в `numericupdown`. Точка, на которой производится товара, выбирается из `combobox`, заполненного названиями точек (рисунок 15).

Количество товара увеличивается или уменьшается в зависимости от нажатой кнопки на форме «Товары».



The screenshot shows a window titled "Редактирование товара" with a dark background. It contains the following elements:

- A text field labeled "Товар" containing the text "A Plague Tale: Innocence (PS4, русские субтитры б/у)".
- A numeric spinner control labeled "Количество" with the value "3".
- A dropdown menu labeled "Точка" with the selected item "Almaz".
- A button labeled "Внести" at the bottom center.

Рисунок 15 – форма редактирования товара

5.7 Форма «Бонусы»

Форма «Бонусы» (рисунок 16) отображает количество накопленных бонусов. Поиск по номеру осуществляется в соответствующем поле. В поле «Списать бонусы» вводится количество бонусов для списания. При нажатии кнопки «Показать покупки клиентов» открывается форма с совершёнными покупками найденной бонусной карты.

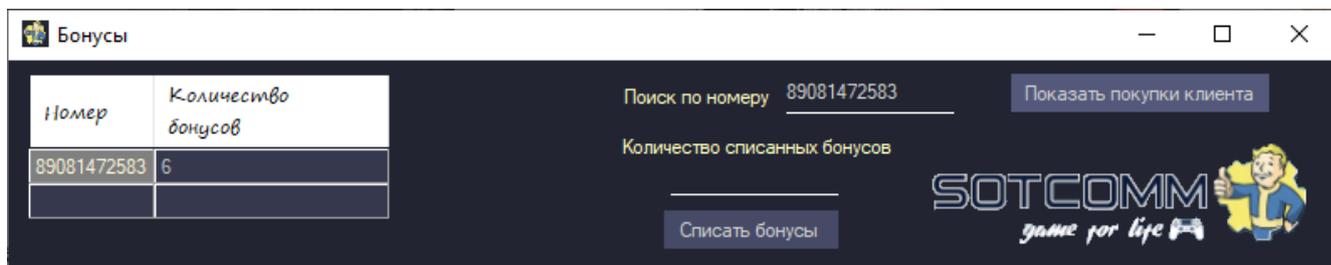


Рисунок 16 – форма «Бонусы»

5.8 Форма «Покупки»

Форма «Покупки» отображает в datagridview покупки номера, введённого на форме «Бонусы» (рисунок 17).

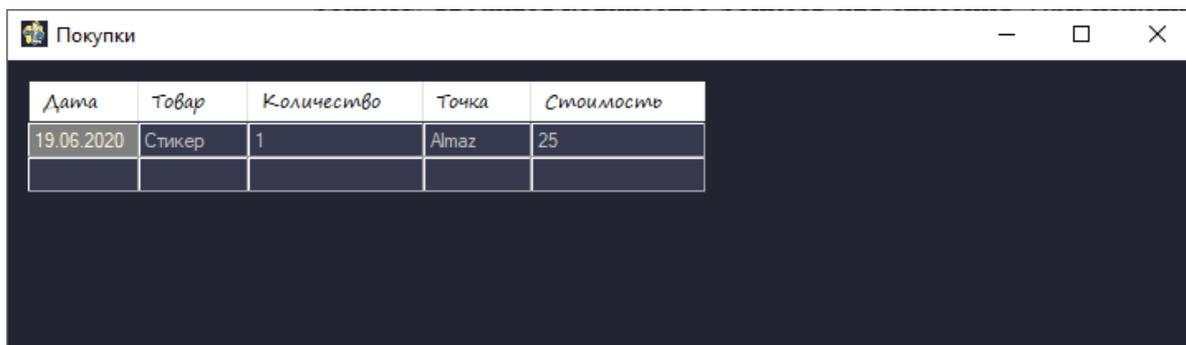


Рисунок 17 – форма «Покупки»

5.9 Форма «Отчёт»

На форме «Отчёт» продавец вносит все операции, совершённую за смену (рисунок 18).

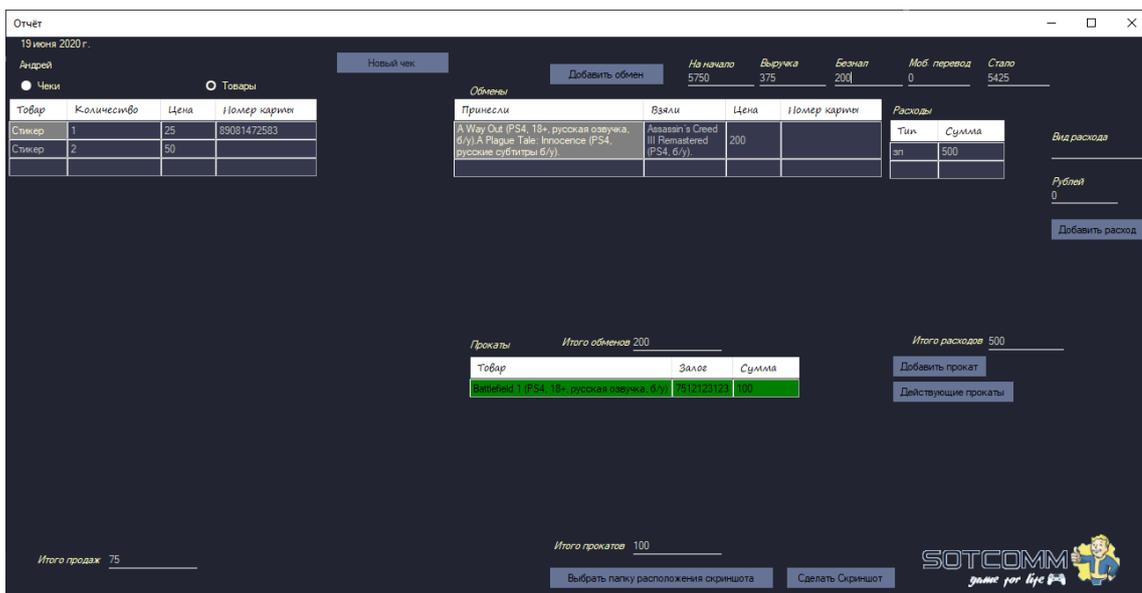


Рисунок 18 – форма «Отчёт»

В левом верхнем углу формы отображается текущая дата. Ниже – имя продавца, работающего в текущую смену.

Таблица «Товары» отображает информацию о покупках, внесённых продавцом в течение смены. При нажатии кнопки «Новый чек», открывается форма «Ввод покупки». При выборе radiobutton «Чеки» демонстрируются покупки, совершённые клиентом (рисунок 19).



Рисунок 19 – Чеки

При двойном нажатии на строку, отображается форма «Покупки» (рисунок 20), показывающая товары, содержащиеся в чеке.

Дата	Товар	Количество	Точка	Стоимость
19.06.2020	Стикер	1	Алмаз	25

Рисунок 17 – форма «Покупки»

Таблица «Обмены» содержит данные об обменах, произведённых в течение смены. При нажатии на кнопку «Добавить обмен», открывается форма «Обмен». Сумма обменов автоматически вносится в поле «Итого продаж».

Таблица «Прокаты» хранит прокаты, совершённые в течение смены. Зелёным цветом выделяются действующие прокаты, красным цветом – истекшие. При нажатии на кнопку «Добавить прокат», открывается форма «Прокат». Сумма прокатов автоматически вносится в поле «Итого прокатов». Нажатие кнопки «Действующие прокаты» открывает форму «Действующие прокаты» (рисунок 21).

Таблица «Расходы» отображает введённые расходы, совершённые в течение смены. При нажатии кнопки «Добавить расход», вводится запись, основанная на введённых данных в полях «Вид расхода» и «Рублей». Сумма расходов автоматически вносится в поле «Итого расходов».

Поле «Выручка» суммирует значения полей «Итого продаж», «Итого обменов» и «Итого прокатов»

Поле «Стало» хранит сумму полей «Выручка» и «На начало», а также разность полей «Итого расходов», «Безнал» и «Моб перевод».

5.10 Форма «Вводпокупки»

Форма «Ввод покупки» предназначена для внесения продаж, совершённых клиентами (рисунок 21).

Рисунок 18 – форма «Ввод покупки»

При открытии формы создаётся новая запись в таблице БД *check*. Все покупки будут вноситься в соответствующий чек (кнопка «Добавить») вплоть до закрытия формы (кнопка «Закреть чек»).

Поле «Товар» содержит коллекцию наименований товаров из таблицы БД *tovari*. При вводе в поле, появляется отфильтрованный выпадающий список. Количество товара регулируется в *numericupdown*. Поле «Стоимость» заполняется автоматически, в зависимости от выбранного товара и его количества. Если поле «Номер бонусной карты» содержит номер карты клиента, бонусы автоматически зачисляются на карту в таблице БД *bonus*.

5.11 Форма «Обмен»

Форма «Обмен» предназначена для ввода обменов товарами (дисков, консолей и т.д.) в процессе смены (рисунок 22). Количество товара меняется в зависимости от введённых данных.

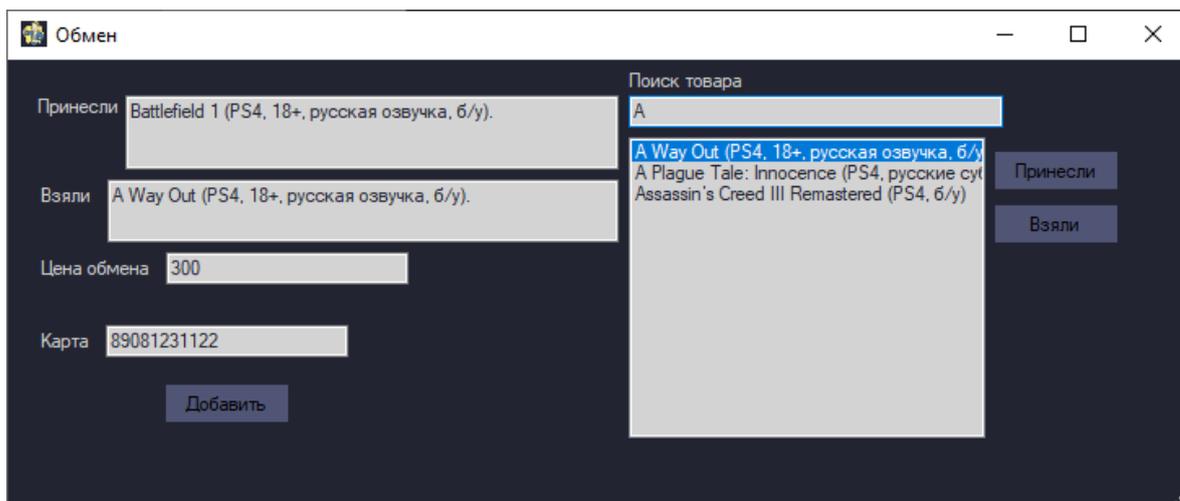


Рисунок 19 – форма «Обмен»

Поиск товаров происходит в соответствующем поле. Найденные элементы отображаются в списке. Выделенный товар заносится в поле «Принесли» или «Взяли» при нажатии аналогичных кнопок. В поле «Цена обмена» указывается стоимость обмена. При наличии у клиента бонусной карты, номер записывается в поле «Карта». Запись введённых данных в таблицу БД обменпроизводится при нажатии на кнопку «Добавить».

5.12 Форма «Прокат»

В данной форме происходит ввод данных в таблицу БД prokat(рисунок 23). Количество товара меняется в зависимости от введённых данных.



Рисунок 20 – форма «Прокат»

Поиск и выбор товара происходит аналогично форме «Обмен». Клиент оставляет залог в виде паспорта или полной стоимости товара (поле «Залог»). Сумма проката указана в соответствующем поле.

5.13 Форма «Действующие прокаты»

Форма отображает прокаты со статусом «Активный» (рисунок 24).

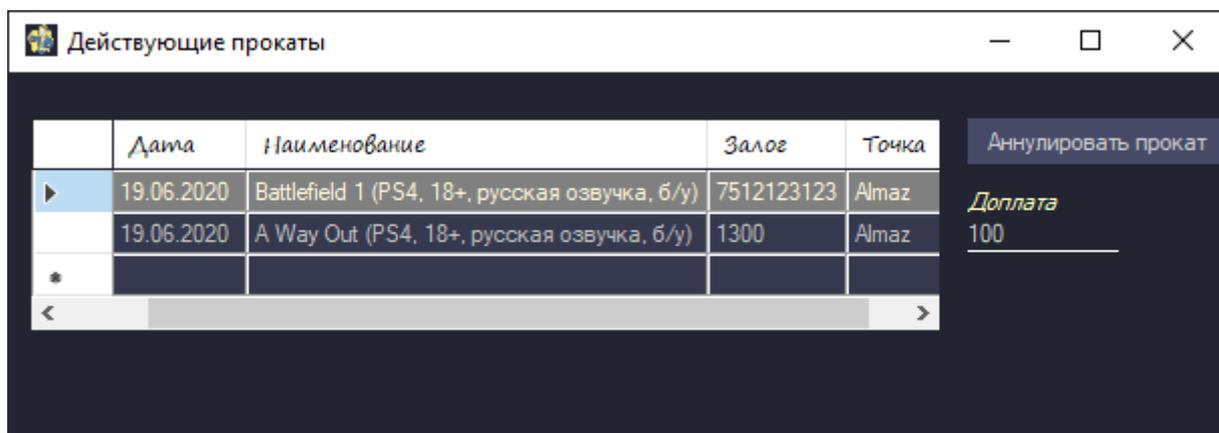


Рисунок 21 – форма «Действующие прокаты»

Аннулирование проката происходит при нажатии соответствующей кнопки. В случае, если клиент возвращает товар позже срока, необходимо указать стоимость дополнительного проката (поле «Доплата»). Аннулированный прокат отобразится на форме «Отчёт» в таблице «Прокаты» (рисунок 25).

The screenshot shows a table with the following data:

Товар	Залог	Сумма
Battlefield 1 (PS4, 18+, русская озвучка, б/у)	7512123123	0

Рисунок 22 – Аннулированный прокат

Выводы по разделу

Описаны этапы разработки программы, показан интерфейс форм. Разработаны функции для упрощения работы некоторых форм.

6 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Для разработки программы потребовались услуги разработчика. Оклад разработчика указана в таблице 6.1.

Таблица 6.1 – Оклад разработчика программы

Должность	Ставка	Оклад (руб/мес)
Программист-разработчик	1.0	29 000

Время, потраченное на разработку программы – 3 месяца. Исходя из отработанного времени, рассчитана зарплата разработчика за сделанную программу. Данные представлены в таблице 6.2.

Таблица 6.2 – Зарплата разработчика программы

Должность	Оклад (руб/мес)	Отработано часов	Зарплата (руб)
Программист-разработчик	29 000	90	87 000

Для того чтобы у каждой торговой точки был доступ к базе данных, куплен тариф на хостинге Veget с доступом на 1 год. Стоимость тарифа представлена в таблице 6.3.

Таблица 6.3 – Стоимость тарифа хостинга

Название тарифа	Количество месяцев	Стоимость (руб)
Vlog	12	1 980

В экономические затраты проекта входит покупка оборудования для продавцов, работающих в сети.

На данный момент сеть насчитывает четыре торговые точки. На каждую точку закуплены ноутбуки и компьютерные мыши для удобства работы. Затраты на оборудование представлены в таблице 6.4.

Таблица 6.4 – Затраты на приобретение оборудования

Наименование	Количество	Цена за 1 единицы (руб)	Общая сумма
Ноутбук HP 15-rb073ur	4	18 990	75 960
Мышь беспроводная Oklick 465MW черный	4	230	920
ИТОГО:			76 880

Таким образом, общие затраты на разработку проекта и покупку оборудования составили 165860 рублей.

Выводы по разделу шесть

Рассчитана зарплата разработчика программы. Описаны затраты на приобретение оборудования для работы с программой на торговых точках.

ЗАКЛЮЧЕНИЕ

Результатом дипломного проекта является разработанная программа «Бонусная система+товары».

Цель дипломного проекта – автоматизация процесса продаж, учета товаров и реализация бонусной системы сети магазинов видеоигр Sotcomm.

Проведен сравнительный анализ существующих систем, в результате которого обоснована целесообразность разработки.

Построены и описаны диаграммы, показывающие структуру программы.

Выполнены следующие задачи, поставленные в требованиях к разработке:

- создана и нормализована база данных для хранения товаров и бонусных карт;
- разработана программа для работы с базой данных;
- реализована возможность редактирования данных в программе;
- реализован многопользовательский режим работы – разграничены права пользователей.

Рассчитаны затраты на покупку оборудования для работы с программой на торговых точках и зарплата программисту-разработчику за сделанную программу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 1С:Розница 8 – <https://v8.1c.ru/retail/>.
- 2 1С:Торговля и склад 8 – <https://1c.ru/rus/products/1c/predpr/torg77.htm>.
- 3 Складской учет товаров онлайн: Мой склад – <https://www.moysklad.ru>.
- 4 Делисл М. Освоение Phpmyadmin для эффективного управления MySQL: учебное пособие / М. Делисл – Москва: Изд. PacktPublishing, 2009. – 316 с.
- 5 Таблицы – SQL Server – <https://docs.microsoft.com/ru-ru/sql/relational-databases/tables/tables?view=sql-server-ver15>.
- 6 Гольцман В. MySQL 5.0. Библиотека программиста: учебное пособие / В. Гольцман – Санкт-Петербург: Изд. ПИТЕР, 2010. – 253 с.

ПРИЛОЖЕНИЕ А

Листинг кода формы «Ввод покупки»

```
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 MySql.Data.MySqlClient;
namespace Sotcomm
{
    public partial class Vvodp : Form
    {
        int i, ID,s = 0;
        public Vvodp()
        {
            InitializeComponent();
        }
        private void Vvodp_Load(object sender, EventArgs e)
        {
            textBox4.Focus();
            G.con.Open();
            string query = "select * from tovari";
            MySqlDataAdapter da = new MySqlDataAdapter(query, G.con);
            DataTable dt = new DataTable();
            da.Fill(dt);
            foreach (DataRow dr in dt.Rows)
            {
                textBox4.AutoCompleteCustomSource.Add(dr["Name"].ToString());
            }
            MySqlCommand cmd1 = G.con.CreateCommand();
            cmd1.CommandType = CommandType.Text;
            cmd1.CommandText = "Insert INTO chek (Date, Time, Point,Count,Summa)
VALUES ('" + DateTime.Today.ToString("yyyy-MM-dd") + "','" +
DateTime.Now.ToString("HH:mm") + "','" + G.login + "','" + i + "','" + s + "')";
            cmd1.ExecuteNonQuery();
            string queryID = "Select MAX(ID) as ID from chek ";
            MySqlDataAdapterchek = new MySqlDataAdapter(queryID, G.con);
            DataTablech = new DataTable();
            chek.Fill(ch);
            ID = Convert.ToInt32(ch.Rows[0]["ID"].ToString());
            G.con.Close();
        }
        private void Vvodp_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (i == 0)
            {
                G.con.Open();
            }
        }
    }
}
```

```

MySQLCommand cmd1 = G.con.CreateCommand();
    cmd1.CommandType = CommandType.Text;
    cmd1.CommandText = "DELETE FROM `chek` WHERE ID = '" + ID + "'";
    cmd1.ExecuteNonQuery();
G.con.Close();
    }
    Report r = new Report();
r.Show();
    }
    private void button1_Click_1(object sender, EventArgs e)
    {
        if (textBox2.Text != "")
        {
            i = i + 1;
                s = s + Convert.ToInt32(textBox2.Text.ToString());
            }
            int number = 0;
G.con.Open();
MySQLDataAdapter t = new MySQLDataAdapter("select ID from tovari Where Name ='" +
textBox4.Text.ToString() + "'", G.con);
DataTable T = new DataTable();
t.Fill(T);
        int b;
        if (checkBox1.Checked == false &&textBox3.Text.Count() == 11)
        {
            b = Convert.ToInt32(textBox2.Text) * 7 / 100;
        }
        else
        {
            b = 50;
        }
        try
        {
            string query = "Select * from bonus where Number = '" + text-
Box3.Text.ToString() + "'";
MySQLCommand cmd3 = new MySQLCommand(query, G.con);
MySQLDataAdapter da = new MySQLDataAdapter(query, G.con);
DataTable dt = new DataTable();
da.Fill(dt);
            int c = dt.Rows.Count;
            int IDt = Convert.ToInt32(T.Rows[0]["ID"].ToString()); //ID Товара
            if (c > 0)
            {
                number = Convert.ToInt32(dt.Rows[0]["ID"].ToString());
MySQLCommand cmd2 = G.con.CreateCommand();
                cmd2.CommandType = CommandType.Text;
                cmd2.CommandText = "Update bonus Set `BonusCount` = `Bo-
nusCount`+' + b + "' where ID = '" + number + "'";
                cmd2.ExecuteNonQuery();
            }
            else if (c == 0 &&textBox3.Text.Count() == 11)
            {
MySQLCommand cmd4 = G.con.CreateCommand();

```

```

cmd4.CommandType = CommandType.Text;
cmd4.CommandText = "Insert INTO bonus (Number, BonusCount) VALUES (" + text-
Box3.Text.ToString() + ", " + b + ")";
cmd4.ExecuteNonQuery();
string query5 = "Select * from bonus where Number = " + text-
Box3.Text.ToString() + """;
MySqlDataAdapter da5 = new MySqlDataAdapter(query, G.con);
DataTable dt5 = new DataTable();
da5.Fill(dt5);
number = Convert.ToInt32(dt5.Rows[0][0].ToString());
}
MySqlCommand cmd1 = G.con.CreateCommand();
cmd1.CommandType = CommandType.Text;
cmd1.CommandText = "UPDATE `tovari` SET `Count` " + G.login +
" `=`Count` " + G.login + "`-`" + Convert.ToInt32(numericUpDown1.Value) + "`
WHERE Name = " + textBox4.Text.ToString() + """;
cmd1.ExecuteNonQuery();
if (number == 0)
{
MySqlCommand cmd = G.con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "INSERT INTO buy (Date,Tovar,Chek,Count,Point,Price,Number) VAL-
UES (" + DateTime.Today.ToString("yyyy-MM-dd") + ", " + Idt + ", " + ID + ", " +
+ Convert.ToInt32(numericUpDown1.Value) + ", " + G.login + ", " + Con-
vert.ToInt32(textBox2.Text.ToString()) + ", NULL)";
cmd.ExecuteNonQuery();
}
else
{
MySqlCommand cmd = G.con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "INSERT INTO buy (Date,Tovar,Chek,Count,Point,Price,Number) VAL-
UES (" + DateTime.Today.ToString("yyyy-MM-dd") + ", " + Idt + ", " + ID + ", " +
+ Convert.ToInt32(numericUpDown1.Value) + ", " + G.login + ", " + Con-
vert.ToInt32(textBox2.Text.ToString()) + ", " + number + ")";
cmd.ExecuteNonQuery();
}
}
catch
{
MessageBox.Show("Проверьте введенные данные");
}
MySqlCommand cmd5 = G.con.CreateCommand();
cmd5.CommandType = CommandType.Text;
cmd5.CommandText = "Update chek Set Count = " + I + ", Summa = " + s
+ " where ID = " + ID + """;
cmd5.ExecuteNonQuery();
G.con.Close();
textBox4.Clear();
textBox2.Clear();
numericUpDown1.Value = 1;
}

```

```

        private void numericUpDown1_ValueChanged(object sender, EventArgs e)
        {
            G.con.Open();
            MySqlCommand cmd = G.con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "select * from tovari where Name = '" + textBox4.Text.ToString()
            + "' ";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            MySqlDataAdapter da = new MySqlDataAdapter(cmd);
            da.Fill(dt);
            foreach (DataRow dr in dt.Rows)
            {
                int w = Convert.ToInt32(dr["Price"]) * Convert.ToInt32(numericUpDown1.Value);
                textBox2.Text = w.ToString();
            }
            G.con.Close();
        }
        private void textBox4_TextChanged(object sender, EventArgs e)
        {
            G.con.Open();
            MySqlCommand cmd = G.con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "select * from tovari where Name = '" + textBox4.Text.ToString()
            + "' ";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            MySqlDataAdapter da = new MySqlDataAdapter(cmd);
            da.Fill(dt);
            foreach (DataRow dr in dt.Rows)
            {
                int w = Convert.ToInt32(dr["Price"]) * Convert.ToInt32(numericUpDown1.Value);
                textBox2.Text = w.ToString();
            }
            G.con.Close();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            if (textBox2.Text != "")
            {
                i = i + 1;
                s = s + Convert.ToInt32(textBox2.Text.ToString());

                int number = 0;

                G.con.Open();
                MySqlDataAdapter t = new MySqlDataAdapter("select ID from tovari Where Name ='" +
                textBox4.Text.ToString() + "'", G.con);
                DataTable T = new DataTable();
                t.Fill(T);
                int b;
            }
        }
    }
}

```

```

        if (checkBox1.Checked == false && textBox3.Text.Count() == 11)
        {
            b = Convert.ToInt32(textBox2.Text) * 7 / 100;
        }
        else
        {
            b = 50;
        }
        try
        {
            string query = "Select * from bonus where Number = '" + text-
Box3.Text.ToString() + "'";
            MySqlCommand cmd3 = new MySqlCommand(query, G.con);
            MySqlDataAdapter da = new MySqlDataAdapter(query, G.con);
            DataTable dt = new DataTable();
            da.Fill(dt);

            int c = dt.Rows.Count;
            int IDt = Convert.ToInt32(T.Rows[0]["ID"].ToString()); //ID
Товара

            if (c > 0)
            {
                number = Convert.ToInt32(dt.Rows[0]["ID"].ToString());
                MySqlCommand cmd2 = G.con.CreateCommand();
                cmd2.CommandType = CommandType.Text;
                cmd2.CommandText = "Update bonus Set `BonusCount` = `Bo-
nusCount`+' " + b + "' where ID = '" + number + "'";
                cmd2.ExecuteNonQuery();
            }
            else if (c == 0 && textBox3.Text.Count() == 11)
            {
                MySqlCommand cmd4 = G.con.CreateCommand();
                cmd4.CommandType = CommandType.Text;
                cmd4.CommandText = "Insert INTO bonus (Number, BonusCount)
VALUES ('" + textBox3.Text.ToString() + "', '" + b + "')";
                cmd4.ExecuteNonQuery();
                string query5 = "Select * from bonus where Number = '" +
textBox3.Text.ToString() + "'";
                MySqlDataAdapter da5 = new MySqlDataAdapter(query, G.con);
                DataTable dt5 = new DataTable();
                da5.Fill(dt5);
                number = Convert.ToInt32(dt5.Rows[0][0].ToString());
            }
            MySqlCommand cmd1 = G.con.CreateCommand();
            cmd1.CommandType = CommandType.Text;
            cmd1.CommandText = "UPDATE `товари` SET `Count` '" + G.login +
"'`=`Count` '" + G.login + "'`-' " + Convert.ToInt32(numericUpDown1.Value) + "'
WHERE Name = '" + textBox4.Text.ToString() + "'";
            cmd1.ExecuteNonQuery();
            if (number == 0)
            {
                MySqlCommand cmd = G.con.CreateCommand();
                cmd.CommandType = CommandType.Text;

```

```

cmd.CommandText = "INSERT INTO buy (Date,Tovar,Chek,Count,Point,Price,Number) VAL-
UES ('" + DateTime.Today.ToString("yyyy-MM-dd") + "','" + IDt + "','" + ID + "','"
+ Convert.ToInt32(numericUpDown1.Value) + "','" + G.login + "','" + Con-
vert.ToInt32(textBox2.Text.ToString()) + "',' NULL)";
cmd.ExecuteNonQuery();
    }
    else
    {
MySQLCommandcmd = G.con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "INSERT INTO buy (Date,Tovar,Chek,Count,Point,Price,Number) VAL-
UES ('" + DateTime.Today.ToString("yyyy-MM-dd") + "','" + IDt + "','" + ID + "','"
+ Convert.ToInt32(numericUpDown1.Value) + "','" + G.login + "','" + Con-
vert.ToInt32(textBox2.Text.ToString()) + "','" + number + "')";
cmd.ExecuteNonQuery();
    }
}
catch
{
MessageBox.Show("Проверьте введенные данные");
}
MySQLCommand cmd5 = G.con.CreateCommand();
cmd5.CommandType = CommandType.Text;
cmd5.CommandText = "Update chek Set Count = '" + i + "',Summa = '"
+ s + "' where ID = '" + ID + "'";
cmd5.ExecuteNonQuery();

G.con.Close();
}
this.Close();

    }
}
}

```

ПРИЛОЖЕНИЕ Б

Листинг кода формы «Обмен»

```
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 MySql.Data.MySqlClient;
namespace Sotcomm
{
    public partial class Obmen1 : Form
    {
        public Obmen1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            textBox1.Text.Remove(0, textBox1.Text.Count());
            textBox2.Text.Remove(0, textBox2.Text.Count());
            int b = Convert.ToInt32(textBox3.Text) * 7 / 100;
            char[] t = { '.' };
            string[] p = textBox1.Text.Split(t);
            string[] v = textBox2.Text.Split(t);
            for (int i = 0; i < v.Count(); i++)
            {
                G.con.Open();
                MySqlCommand cmd1 = G.con.CreateCommand();
                cmd1.CommandType = CommandType.Text;
                cmd1.CommandText = "UPDATE `tovari` SET `Count` " + G.login +
                "'`=`Count` " + G.login + "'`-1 WHERE `Name` = '" + v[i].ToString()+"'";
                cmd1.ExecuteNonQuery();
                G.con.Close();
            }
            for (int i = 0; i < p.Count(); i++)
            {
                G.con.Open();
                MySqlCommand cmd1 = G.con.CreateCommand();
                cmd1.CommandType = CommandType.Text;
                cmd1.CommandText = "UPDATE `tovari` SET `Count` " + G.login +
                "'`=`Count` " + G.login + "'`+1 WHERE `Name` = '" + p[i].ToString() + "'";
                cmd1.ExecuteNonQuery();
                G.con.Close();
            }
            G.con.Open();
            MySqlCommand cmd = G.con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "INSERT INTO obmen (Date, Prinesli, Vzyali, Point, Price, Number)
            VALUES ('" + DateTime.Today.ToString("yyyy-MM-dd") + "', '" + text
```

```

Box1.Text.ToString() + "','" + textBox2.Text.ToString() + "','" + G.login + "','"
+ Convert.ToInt32(textBox3.Text.ToString()) + "','" + textBox4.Text.ToString() +
"')";
cmd.ExecuteNonQuery();
string query = "Select * from bonus where Number = '" + textBox4.Text.ToString() +
"'"";
MySQLCommand cmd3 = newMySQLCommand(query, G.con);
MySQLDataAdapter da = newMySQLDataAdapter(query, G.con);
DataTable dt = newDataTable();
da.Fill(dt);
int c = dt.Rows.Count;
if (c > 0)
{
MySQLCommand cmd2 = G.con.CreateCommand();
cmd2.CommandType = CommandType.Text;
cmd2.CommandText = "Update bonus Set `BonusCount` = `Bo-
nusCount`+''" + b + "'" where `Number` = '" + textBox4.Text.ToString() +
"'"";
cmd2.ExecuteNonQuery();
}
elseif (c == 0 &&textBox3.Text.Count() == 11)
{
MySQLCommand cmd4 = G.con.CreateCommand();
cmd4.CommandType = CommandType.Text;
cmd4.CommandText = "Insert INTO bonus (Number, `BonusCount`) VAL-
UES ('" + textBox4.Text.ToString() + "','" + b + "')";
cmd4.ExecuteNonQuery();
}
G.con.Close();
this.Close();
}
privatevoid Obmen1_FormClosing(object sender, FormClosingEventArgs e)
{
Report r = newReport();
r.Show();
}
privatevoid textBox5_TextChanged(object sender, EventArgs e)
{
G.con.Open();
MySQLCommandcmd = newMySQLCommand("SELECT `Name` FROM `tovari` Where `Name` LIKE
'" + textBox5.Text.ToString() + "' '%" , G.con);
MySQLDataReader reader = cmd.ExecuteReader();
listBox1.Items.Clear();
if (textBox1.Text != null)
{
while (reader.Read())
{
listBox1.Items.Add(reader["Name"].ToString());
}
}
G.con.Close();
}
}

```

```
privatevoid button2_Click_1(object sender, EventArgs e)
{
    if (textBox5.Text != "")
    {
        textBox1.Text = textBox1.Text + listBox1.SelectedItem.ToString() +
        '.';
    }
}
privatevoid button3_Click(object sender, EventArgs e)
{
    if (textBox5.Text != "")
    {
        textBox2.Text = textBox2.Text + listBox1.SelectedItem.ToString() +
        '.';
    }
}
privatevoid button4_Click(object sender, EventArgs e)
{
    MessageBox.Show("Введи название игры в поле над таблицей, выбери нужную игру в
таблице, нажми кнопку Принесли/Взяли, выбранная игра вставится в поле
Принесли/Взяли, укажи цену обмена и карту обмена при наличии");
}
```

ПРИЛОЖЕНИЕ В

Листинг кода формы «Бонусы»

```
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 MySql.Data.MySqlClient;
namespace Sotcomm
{
    public partial class Bonus : Form
    {
        public Bonus()
        {
            InitializeComponent();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void Bonus_FormClosed(object sender, FormClosedEventArgs e)
        {
            MainForm mf = new MainForm();
            mf.Show();
        }

        private void Bonus_Load(object sender, EventArgs e)
        {
            MySqlDataAdapter da = new MySqlDataAdapter("select Number, BonusCount from bonus",
            G.con);
            DataSet DS = new DataSet();
            da.Fill(DS);
            dataGridView1.DataSource = DS.Tables[0];
            dataGridView1.Columns[0].HeaderText = "Номер";
            dataGridView1.Columns[1].HeaderText = "Количество бонусов";
            //close connection
            G.con.Close();
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            MySqlDataAdapter da = new MySqlDataAdapter("select Number, BonusCount from bonus
            Where `Number` = '" + textBox1.Text.ToString() + "'", G.con);
            DataSet DS = new DataSet();
            da.Fill(DS);
            dataGridView1.DataSource = DS.Tables[0];
        }
    }
}
```

```

        dataGridView1.Columns[0].HeaderText = "Номер";
        dataGridView1.Columns[1].HeaderText = "Количество бонусов";
//close connection
G.con.Close();
    }

private void button2_Click(object sender, EventArgs e)
    {
G.con.Open();
MySQLCommand cmd2 = G.con.CreateCommand();
        cmd2.CommandType = CommandType.Text;
        cmd2.CommandText = "Update bonus Set `BonusCount` = `BonusCount` - '" +
Convert.ToInt32(textBox2.Text) + "' where `Number` = '" + textBox1.Text.ToString()
+ "'";
        cmd2.ExecuteNonQuery();
MySQLDataAdapter da = new MySQLDataAdapter("select Number, BonusCount from bonus
Where `Number` = '" + textBox1.Text.ToString() + "'", G.con);
DataSet DS = new DataSet();
da.Fill(DS);
        dataGridView1.DataSource = DS.Tables[0];
        dataGridView1.Columns[0].HeaderText = "Номер";
        dataGridView1.Columns[1].HeaderText = "Количество бонусов";
G.con.Close();
    }

private void button1_Click(object sender, EventArgs e)
    {
MySQLDataAdapter da = new MySQLDataAdapter("select ID, Number, BonusCount from bonus
Where `Number` = '" + textBox1.Text.ToString() + "'", G.con);
DataTable DT = new DataTable();
da.Fill(DT);
G.Number = Convert.ToInt32(DT.Rows[0]["ID"].ToString());
G.flag = false;
        buys b = new buys();
b.Show();
    }
}

```