МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра анализа данных и искусственного интеллекта**

**ЛАБОРАТОРНАЯ РАБОТА №10**

**ДОКУМЕНТИРОВАНИЕ И РАЗВЕРТЫВАНИЕ ПС**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.Е. Дакукин

(подпись)

Направление подготовки   01.03.02 Прикладная математика и информатика курс 4

Направленность (профиль) Программирование и информационные технологии

Проверила  
канд. науч. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.Ю. Добровольская

Краснодар

2023

Объект информатизации: столовая «Симфония». Работает на рынке с 2015 года. Количество сотрудников: 20 человек. Объём годовой выручки: 35 млн. рублей. Директор: Анна Иванова. Основные направления: европейская кухня, бизнес-ланчи и обеды для корпоративных клиентов, ужины и мероприятия для семей и друзей, организация тематических мероприятий.

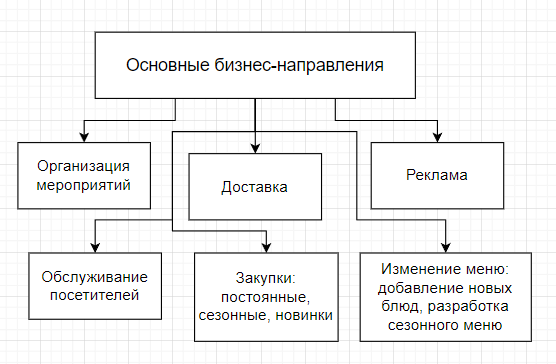


Рисунок 1 – Бизнес-направления

Для информатизации выбирается процесс обработки заказа. Он включает в себя: формирование позиций меню; формирование заказа; подсчитывание конечной стоимости заказа; выставление счёта клиенту для оплаты; запись в журнале отчёта; передача заказа клиенту.

К действующим лицам (актерам) данного бизнес-процесса относятся:

* Клиент: формирует заказ.
* Кассир: принимает заказ, проверяет наличие товаров, координирует процесс подготовки заказа с другими отделами (например, кухней, складом).



Рисунок 2 – Диаграмма «Чёрный ящик»

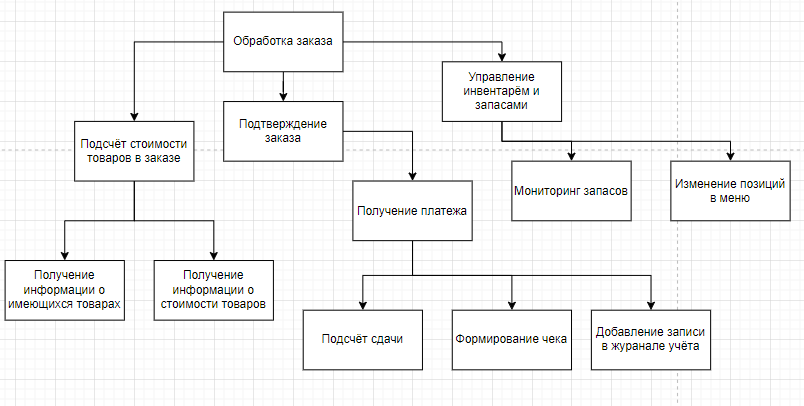


Рисунок 3 – Декомпозиция бизнес-процесса

Задачи:

– Задача получения данных о заказе;

– Задача формирования заказа;

– Задача проведения инвентаризации;

– Задача сохранения данных о заказе;

– Задача формирования чека.

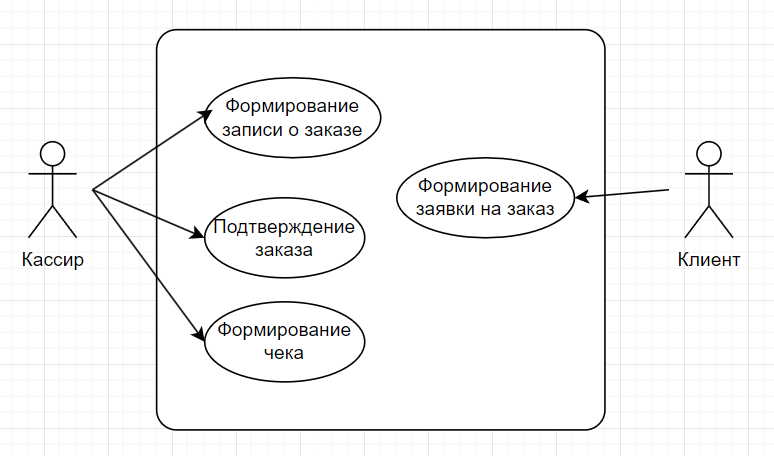


Рисунок 4 – модель вариантов использования UML для бизнес-процесса

Таблица 1. Реестр входных информационных потоков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование  и назначение  потока | Форма  представлен  ия | Обработчик  (Кто  обрабатывает  ) | Корреспон  дент  (Откуда) | Характеристики обработки | |
| Трудозатраты  , чел⋅ч | Периодичность  , регламент |
| 1 | Заявка на заказ | Устное сообщение | Кассир | Клиент | 0,1 | По запросу |
| 2 | Меню | Таблица | Кассир | Сервис | 0.1 | Каждую неделю |
| 3 | Информация о имеющихся продуктах | Таблица | Кассир | Сервис | 0.1 | По запросу |

Таблица 1. Реестр выходных информационных потоков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование  и назначение  потока | Форма  представлен  ия | Обработчик  (Кто  обрабатывает  ) | Корреспон  дент  (Куда) | Характеристики обработки | |
| Трудозатраты  , чел⋅ч | Периодичность  , регламент |
| 1 | Чек | Квитанция | Кассир | Клиент | 0,1 | По запросу |
| 2 | Запись в журнале отчёта | Электронная | Кассир | Клиент | 0,1 | По запросу |

Правила обработки информации и возможные ограничения:

К правилам обработки информации и возможным ограничениям относятся:

* Данные о заказе идентифицируются как уникальная запись в базе;
* Система должна требовать аутентификации для доступа к данным;
* Регулирование доступа сотрудников: предоставление различных уровней доступа в зависимости от роли.

Нормативно-справочная документация, регламентирующая бизнес- процесс:

* Гражданский кодекс РФ;
* Нормы СанПиН;
* Стандарты и ГОСТы по предоставлению услуг общественного питания.

# 1 сервис:

# 1С:Касса - работающее в Интернете (в облаке ) кассовое и товароучётное приложение с встроенным рабочим местом кассира, используемое для автоматизации одного или нескольких магазинов или точек оказания услуг.

# Функционал сервиса:

# Работа в браузере на любом устройстве

* Возможность подключения доп. оборудования, например, кассовых аппаратов, весов
* Ведение учета остатков товаров и денежных средств
* Отчёты о продажах, статистика.
* Приемка и торговля алкоголем (ЕГАИС)
* Работа с маркированными товарами
* Оплата банковскими картами, комбинированная оплата
* Учёт перемещения товаров между торговыми точками
* Печать ценников и этикеток
* Ручные скидки
* Наличие личного кабинета
* Поддержка нескольких точек продаж
* Интеграция с другими продуктами 1С: 1С Розница, 1С Бухгалтерия, 1С БизнесСтарт и пр.

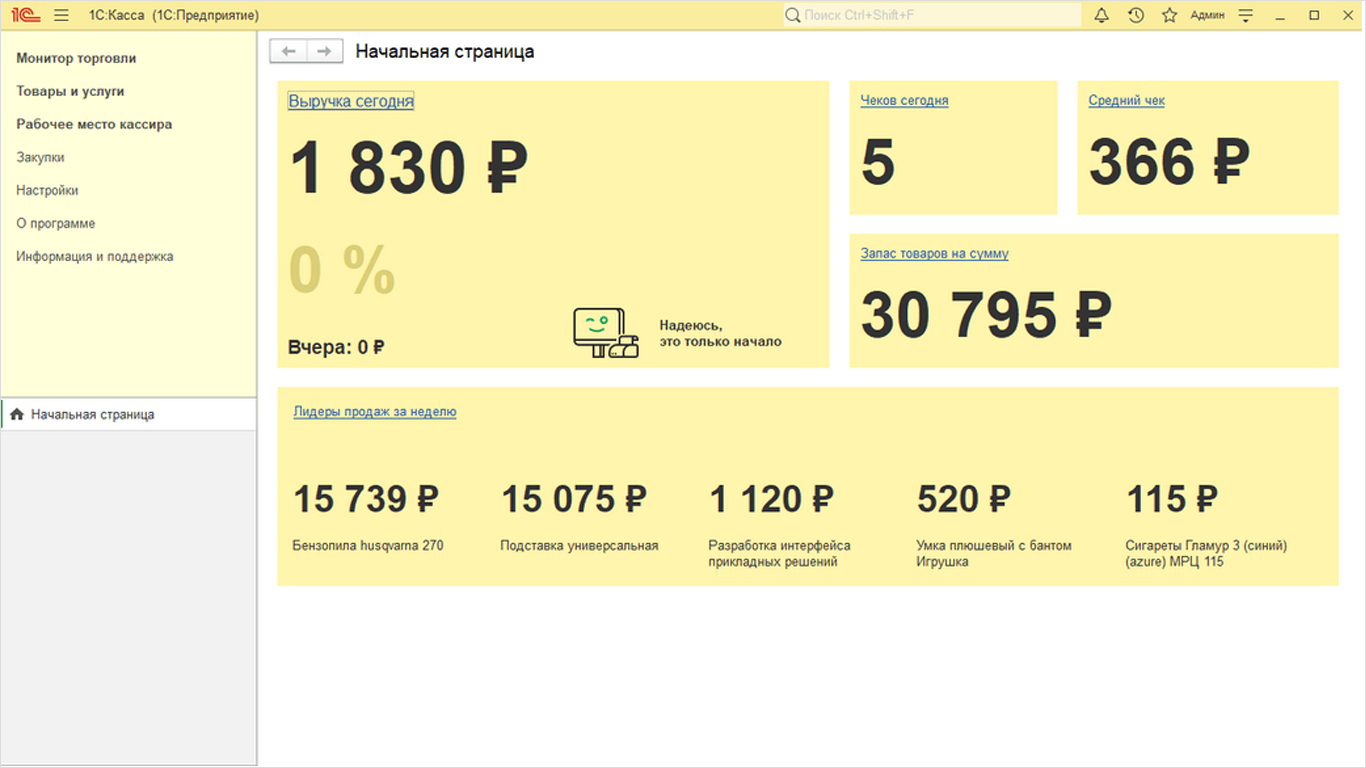


Рисунок 1 – Главная страница сервиса 1С:Касса

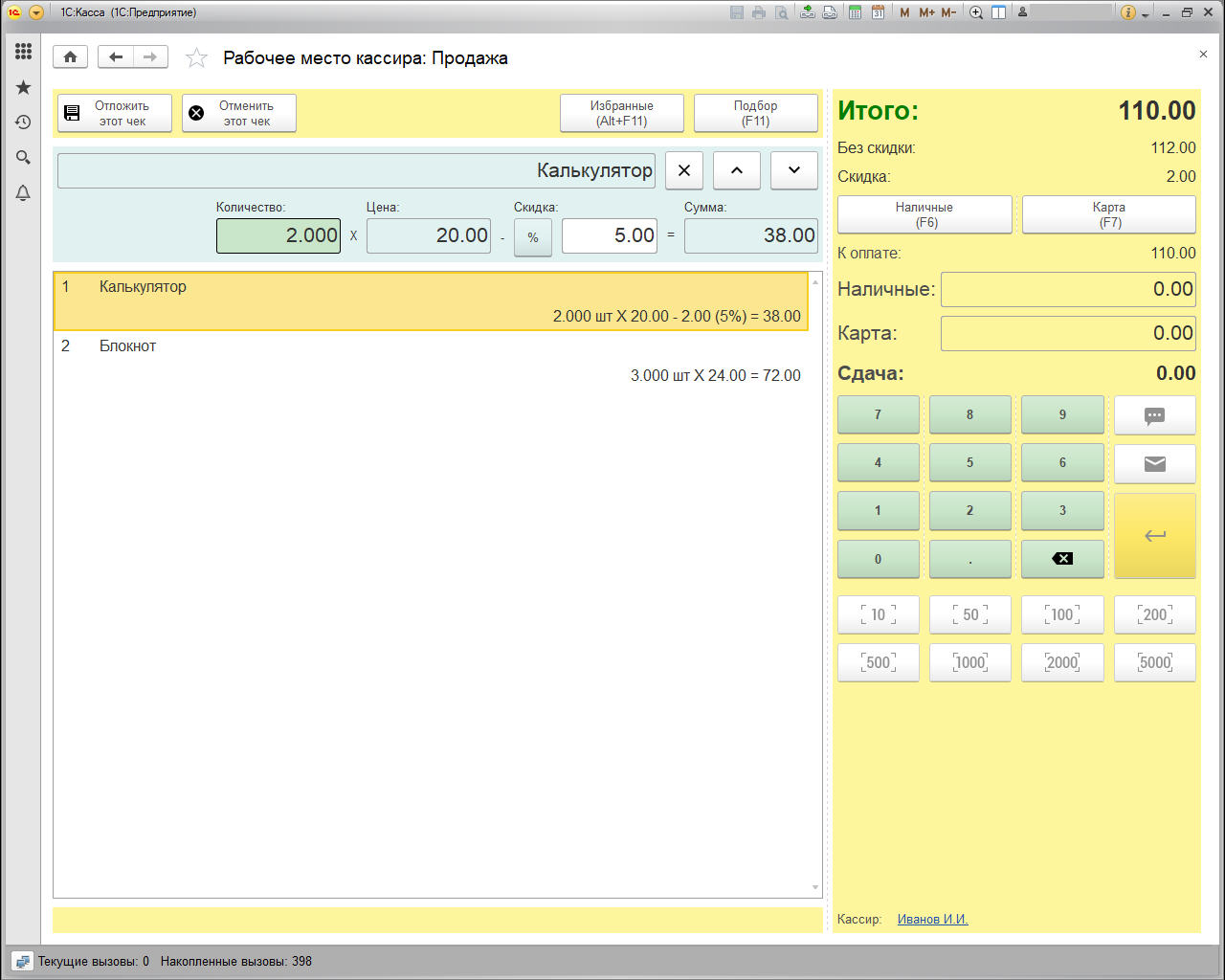


Рисунок 2 – Оформление заказа

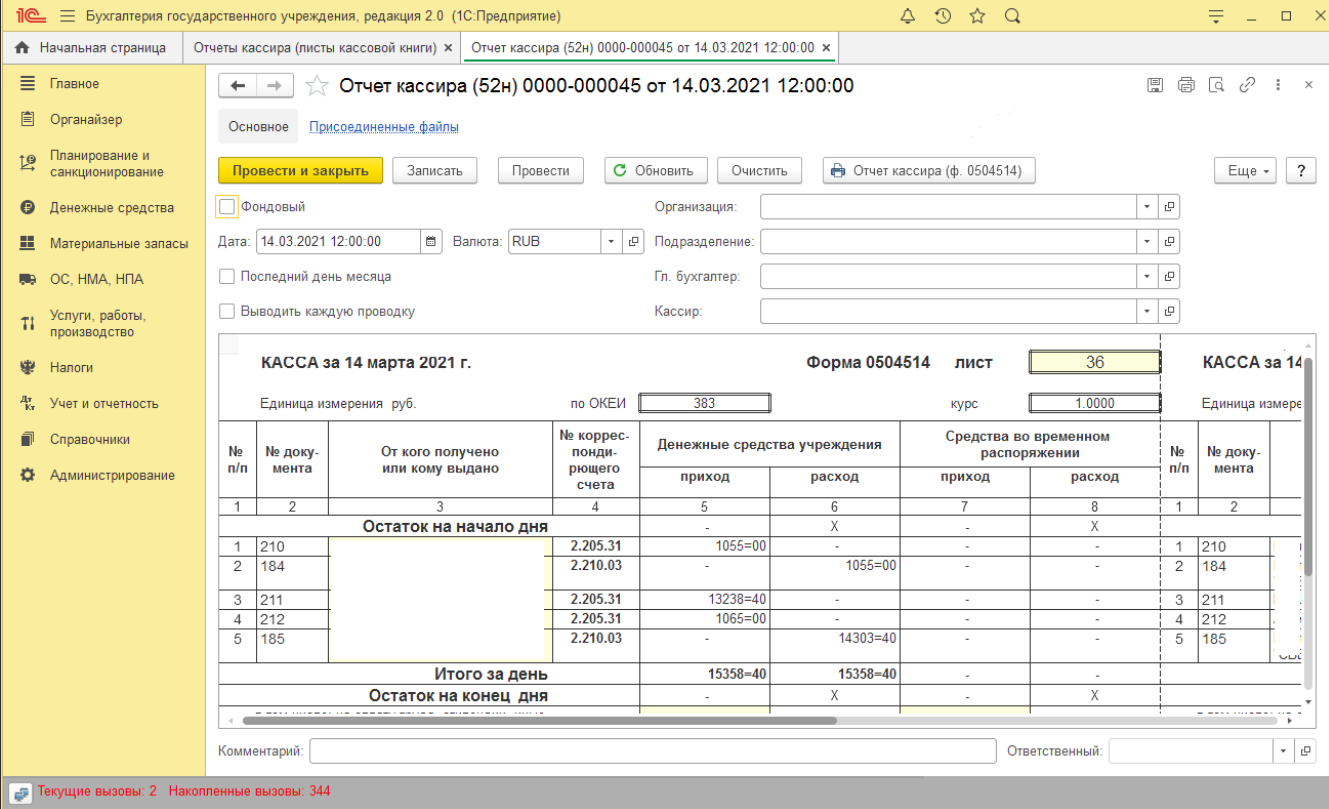


Рисунок 3 – страница формирования отчёта

2 сервис:

GBS.Market - это кассовая программа для малого и среднего бизнеса. Она позволяет следить за остатками на складе, вести учёт продаж, поддерживать в актуальном состоянии весь ассортимент.

Функционал сервиса:

* Поддержка нескольких точек продаж.
* Хранение базы клиентов, с возможностью настройки гибких правил скидок и накопительной системой баллов.
* Возможность подключения доп. оборудования, например, кассовых аппаратов, весов.
* Приемка и торговля алкоголем (ЕГАИС)
* Ведение учета остатков товаров и денежных средств
* Оффлайн-режим.
* Гибкие отчеты о продажах
* Распределение прав доступа сотрудников
* Продажа маркированных товаров
* Ведение учета остатков товаров и денежных средств
* Печать ценников и чеков
* Несколько режимов работы – для кафе, магазина/склада и дома/офиса

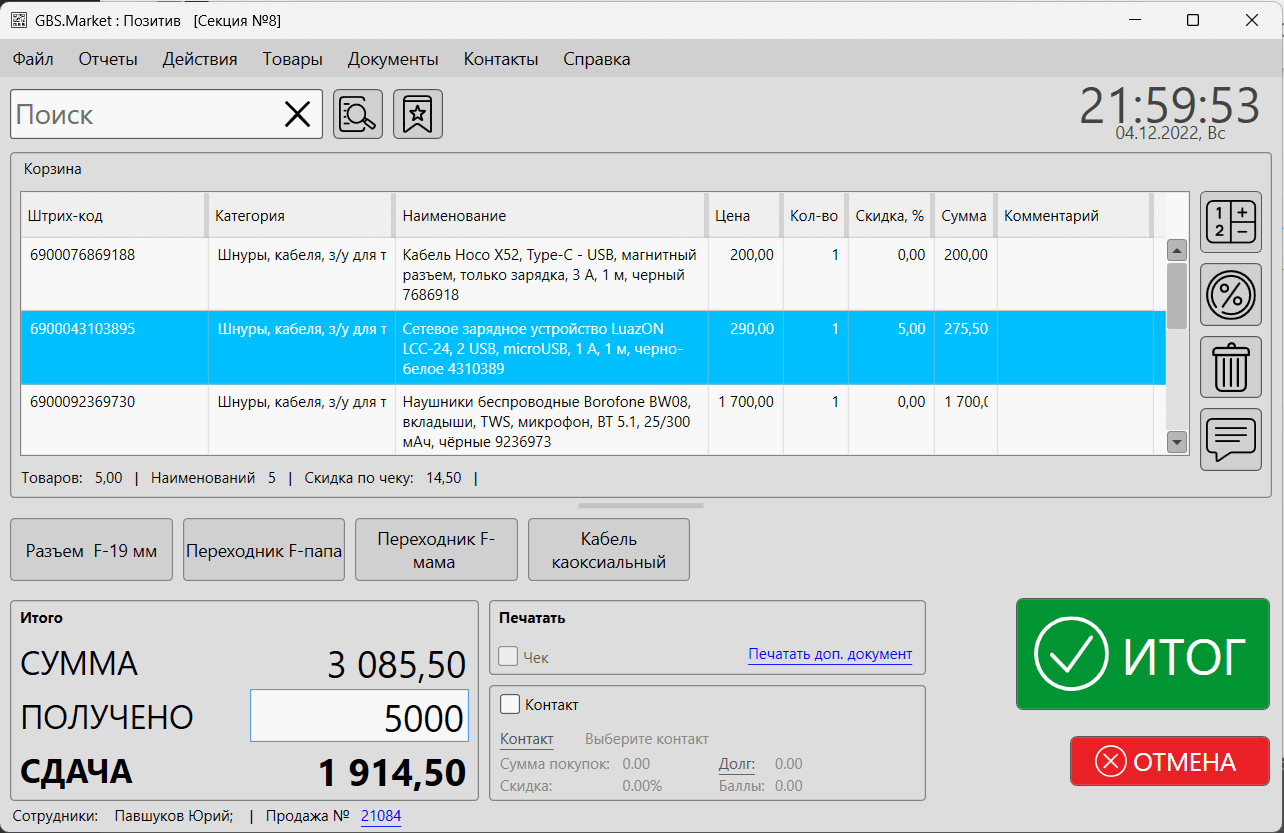


Рисунок 4 – Основная страница приложения

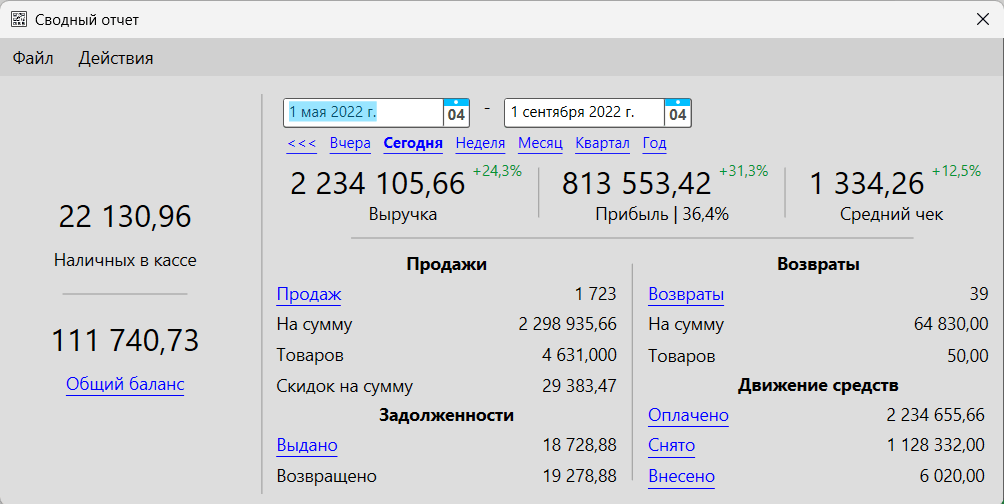


Рисунок 5 – страница формирования отчёта

Сравнение характеристик

|  |  |  |  |
| --- | --- | --- | --- |
| № | Характеристика | 1С:Касса | GBS.Market |
| 1 | Стоимость | 200-600 руб\месяц; 5000 руб\бессрочно; цена за 1 рабочее место | 349 руб\месяц; 999 руб\квартал; 3499 руб\год; 10999 руб\бессрочно; цена за 1 рабочее место |
| 2 | Наличие пробной версии | Есть | Нет |
| 3 | Наличие тех. поддержки | Имеется горячая линия, круглосуточная онлайн служба поддержки | Имеется онлайн служба поддержки |
| 4 | Наглядность | Понятный интерфейс | Простой и понятный интерфейс |
| 5 | Живучесть | Оффлайн режим отсутсвует | Есть оффлайн режим |
| 6 | Производительность | За счёт облачных вычислений практически нетребователен к системе. | При использовании всего функционала может нагружать систему на слабых компьютерах. |
| 7 | Платформы | Облачный сервис, интерфейс | Windows, облачный сервис, интернет |
| 8 | Мобильность | Возможна работа только с другими продуктами 1С. | Не поддерживает работу с другими продуктами |
| 9 | Численность задействованного персонала | 1-10 человек | 1-15 человек |

Вывод: рассматриваемые сервисы имеют во многом схожий функционал и отлично подходят для автоматизации обработки заказов, однако существует ряд отличий. Преимуществом сервиса 1С:Касса является возможность интеграции с другими продуктами 1С, что позволяет автоматизировать многие аспекты бизнеса. Также стоит отметить меньшую стоимость сервиса. Преимуществами сервиса GBS.Market будут выступать наличие базы клиентов и более гибкой настройки системы скидок, а также наличие оффлайн режима.

Для автоматизации был выбран процесс обработки заказа. Перечень объектов автоматизации: кассовые аппараты, персональный компьютер, база данных.

Целью создания подсистемы является Автоматизация процесса обработки заказа, автоматическое ведение учёта проданных товаров, предоставление документации в электронном виде.

Задачи:

– Задача ввода данных о заказе;

– Задача формирования заказа;

– Задача проведения инвентаризации;

– Задача сохранения данных в базе;

– Задача формирования чека.

Функциональные требования:

Требования к «задача ввода данных о заказе»: система должна иметь возможность вводить данные о заказе.

Требования к «задаче формирования заказа»: на основе данных о заказе должен формироваться заказ.

Требования к «задаче инвентаризации»: система должна сохранять информацию о изменениях числах товаров в базе данных, а также информировать о имеющемся количестве товара.

Требования к «задаче сохранения данных в базе»: все необходимые для работы системы данные должны сохраняться в базе данных. Это списки всех заказов, позиций в меню, ингредиентов на складе. Обязательно наличие возможности редактирования, удаления и экспорта произвольной информации из базы данных.

Требования к «задаче редактирования позиций меню»: на основе данных о заказе и доступных ингредиентах на складе должен формироваться заказ.

Требования к «задаче формирования чека»: на основе данных о заказе и клиенте должен формироваться чек.

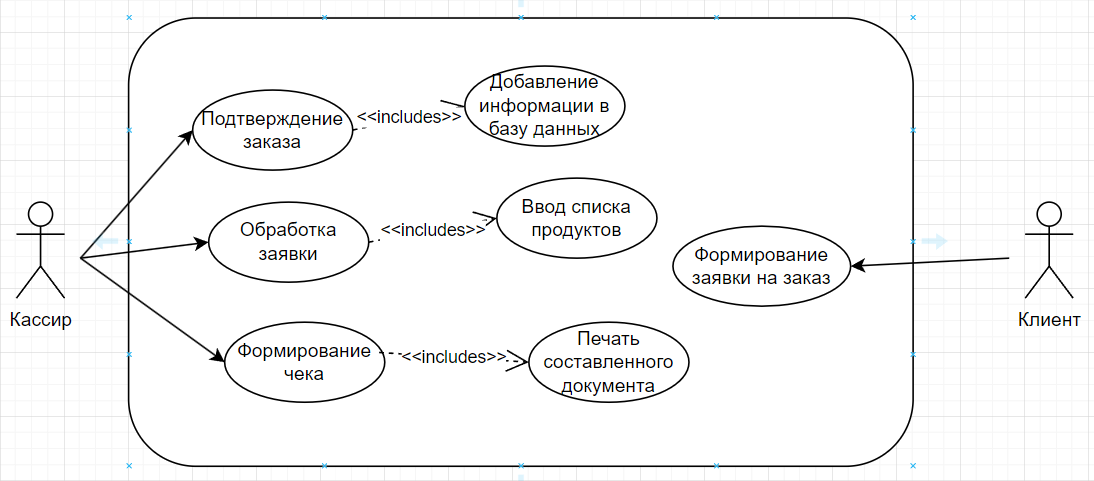


Рисунок 1 – Диаграмма требований UML

База данных проектируемой программной системы должна иметь следующие характеристики:

* Надежность.

База данных должна быть надежной и обеспечивать целостность данных. Данные должны храниться в безопасном месте, чтобы избежать потери информации в результате сбоя системы или внешнего вмешательства.

* Эффективность.

База данных должна обеспечивать высокую скорость доступа к данным и быструю обработку запросов.

* Безопасность.

База данных должна обеспечивать безопасность хранения и передачи данных, чтобы защитить конфиденциальную информацию от несанкционированного доступа.

* Актуальность.

База данных должна содержать актуальную информацию о продажах, чтобы обеспечить точность принятия решений на основе этих данных.

Требования к инструментальному программному обеспечению:

* Необходимо обеспечить поддержку операционных систем Windows 10 и выше;
* Достаточное количество свободного места на жестком диске;
* Среда разработки должна быть многофункциональной и поддерживать многие аспекты разработки программного обеспечения;

Нефункциональные требования к ПО:

* База данных для разрабатываемой системы не должна терять данные в случае аварийного завершения системы;
* Программное обеспечение должно решать поставленные задачи за приемлемое время, не должно быть продолжительных простоев в работе программы;
* Удобный и интуитивно понятный пользовательский интерфейс приложения;
* Должна присутствовать возможность скачать программное обеспечение из интернета или передать с помощью внешнего источника;
* Не должно возникать сбоев в работе программы, не связанных с недостатками технического оборудования;



Рисунок 1 – Диаграмма «Чёрный ящик»

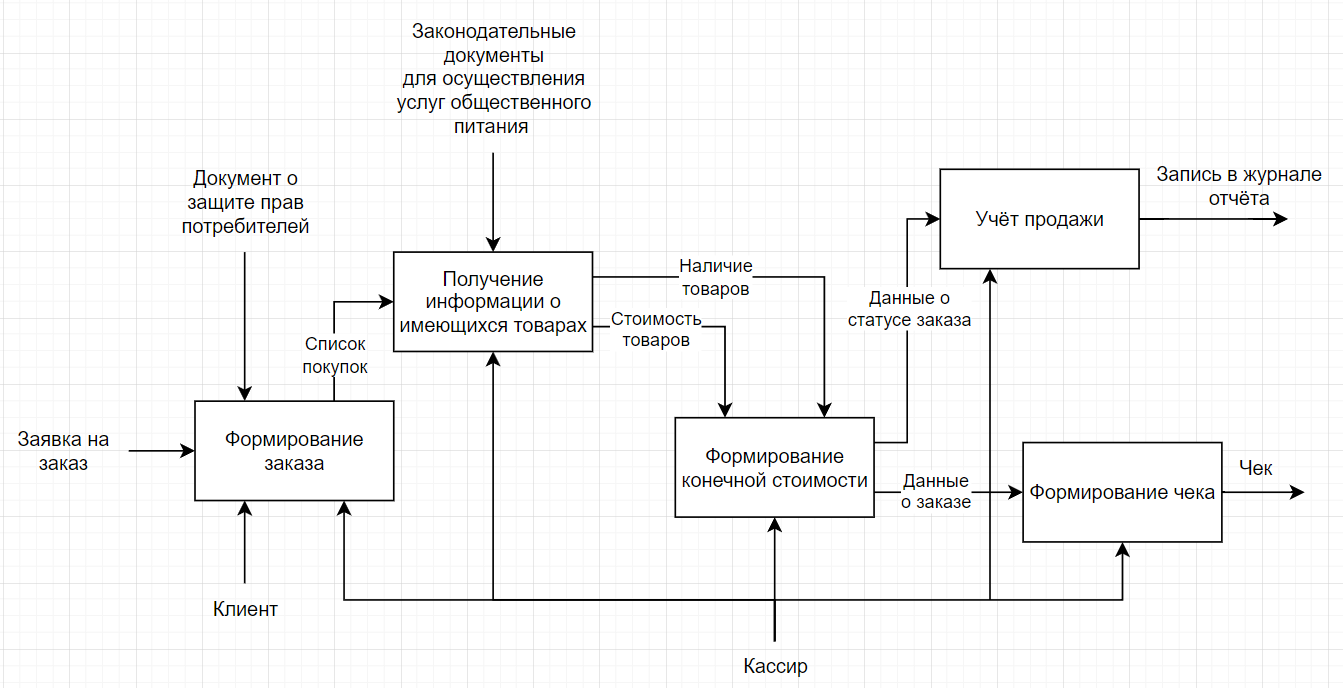


Рисунок 2 – Диаграмма декомпозиции А0

Таблица 1— описание элементов функциональной модели

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приёмник стрелки | Тип стрелки приёмника |
| Кассир | Внешняя граница | Mechanism | Формирование заказа | Mechanism |
| Клиент | Внешняя граница | Mechanism | Формирование заказа | Mechanism |
| Законодательные  документы для осуществления  услуг общественного питания | Внешняя граница | Input | Получение информации о имеющихся товарах | Input |
| Документ о защите прав потребителей | Внешняя граница | Input | Составление заказа | Input |
| Заявка на заказ | Внешняя граница | Input | Составление заказа | Input |
| Список покупок | Формирование заказа | Output | Получение информации о имеющихся товарах | Input |
| Стоимость товаров | Получение информации о имеющихся товарах | Output | Формирование конечной стоимости | Input |
| Наличие товаров | Получение информации о имеющихся товарах | Output | Формирование конечной стоимости | Input |
| Данные о заказе | Формирование конечной стоимости | Output | Формирование чека | Input |
| Данные о статусе заказа | Формирование конечной стоимости | Output | Учёт продажи | Input |
| Запись в журнале отчёта | Учёт продажи | Output | Внешняя граница | Output |
| Чек | Формирование чека | Output | Внешняя граница | Output |

Таблица 1 – Абстракции подсистемы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Тип | Описание |
| 1 | Кассир | Сущность | Пользователь системы, взаимодействующий с ней |
| 2 | Продукт | Сущность | Информация о имеющихся продуктах для продажи |
| 3 | Подсчёт конечной стоимости | Поведение | Подсчёт стоимости заказа, с учётом скидки, стоимости и количества продуктов |
| 4 | Формирование заказа | Поведение | Проведение самой системой определенных действий, связанных с формированием заказа. |
| 5 | Авторизация кассира | Поведение | Введение кассиром идентификационных данных для получения доступа к системе |
| 6 | Получение информации об остатках продуктов | Поведение | Получение информации о наличии продуктов от системы |
| 7 | Заказ | Сущность | Список продуктов в заявке |
| 8 | Запись о проведённой продаже в журнал отчёта | Поведение | Формирование записи в соответствии с проведённой операцией продажи |
| 9 | Чек | Сущность | Информация о суммарной цене товаров, статусе заказа, дате, способу оплаты. |

Таблица 2. Классификация абстракции

|  |  |  |
| --- | --- | --- |
| № | Класс | Список абстракций |
| 1 | Люди | Кассир |
| 2 | Предметы | Продукт, заказ, чек |
| 3 | События | Авторизация кассира, запись о проведённой продаже в журнал отчёта, формирование заказа, подсчёт конечной стоимости, получение информации об остатков продуктов |

Таблица 3. Абстракции подсистемы и их поведение

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Поведение | Описание поведения |
| 1 | Кассир | Внести, добавить, удалить, подтвердить | Вносит в систему информацию о заказе, добавляет, удаляет позиции заказа, подтверждает заказ |
| 2 | Чек | Выдать | Выдать результат работы сервиса - чек |
| 3 | Формирование заказа | Сформировать | Сформировать заказ на основе введённых в систему данных |
| 4 | Позиции заказа | Добавить, удалить, изменить | Редактирование позиций заказа |
| 5 | Конечная стоимость | Подсчитать | Рассчитывает конечную стоимость заказа, с учётом скидки, стоимости и количества продуктов |
| 6 | Статус заказа | Изменить | Изменение статуса заказа при внесении оплаты, отмене заказа |
| 7 | Запись о заказе | Внести | Внесение записи о заказе в систему |
| 8 | Аккаунт | Проверяет | Проверяет правильность введённых авторизационных данных |

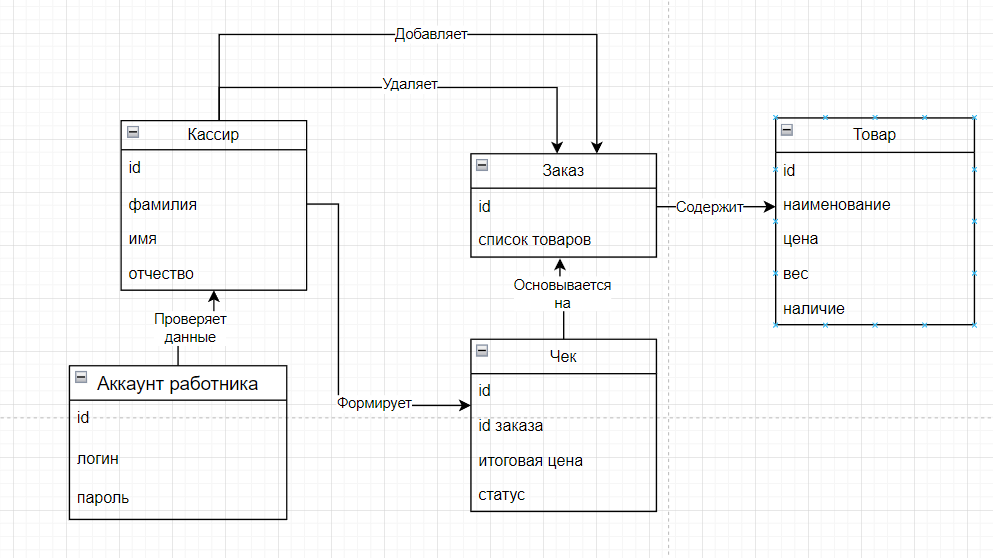


Рисунок 1 – диаграмма классов

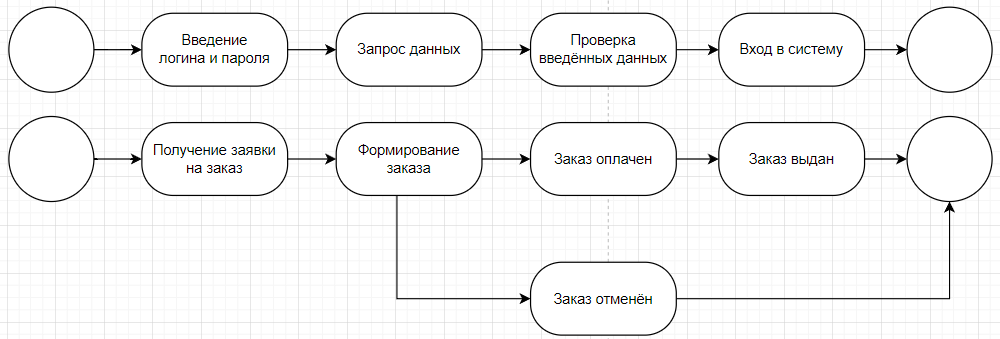


Рисунок 2 – Диаграммы состояния

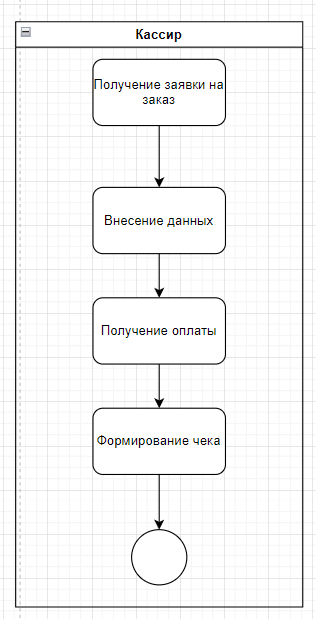


Рисунок 3 – Диаграмма деятельности

Таблица 4. Список объектов для каждого потока событий

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Прецедент | Объект | Описание объекта |
| 1 | Авторизация кассира | Модуль авторизации | Часть системы, ответственная за авторизацию кассира |
| 2 | Сформировать заказ для клиента | Модуль формирования заказа | Часть системы, ответственная за формирование заказа для клиента |

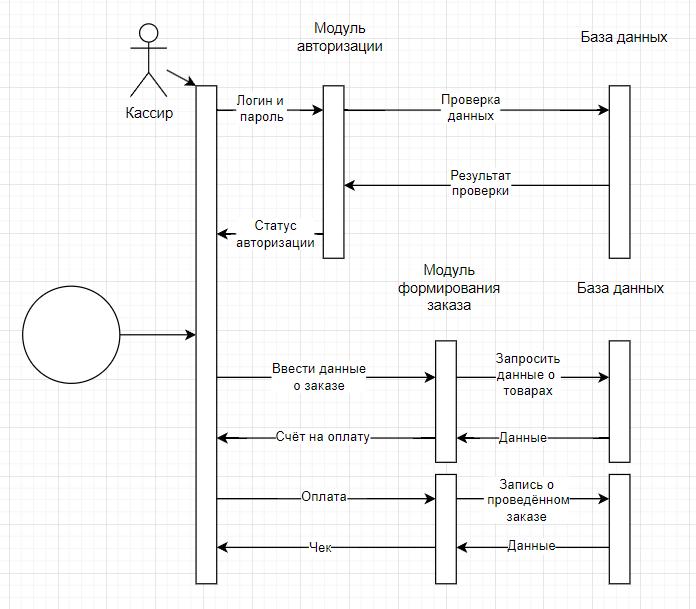


Рисунок 4 – Диаграмма последовательности

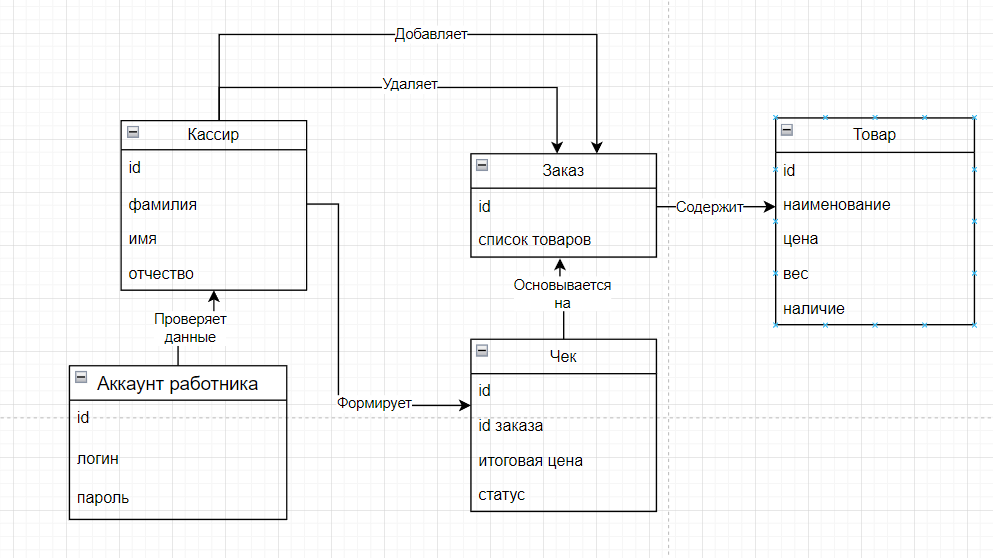


Рисунок 1 – диаграмма классов

Таблица 1 – Идентификация сущностей

|  |  |
| --- | --- |
| Документ | Сущность |
| Кассир | Кассир |
| Чек | Чек |
| Аккаунт |  |
| Список товаров в заказе | Заказ |
| Товар | Товар |

Таблица 2 – Сущности физической модели данных

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Описание |
| Справочные | | |
| 1 Кассир | ID кассира, ФИО, номер телефона | Информация о кассире |
| 2 Аккаунт | ID кассира, логин, пароль | Данные для входа кассира |
| 3 Товар | ID товара, название, цена, наличие, описание, ограничения | Информация о продаваемых товарах |
| Оперативные | | |
| 5 Чек | ID чека, обслуживающий кассир, общая стоимость, статус, дата, способ оплаты | Информация о заказе |
| 5 Заказ | ID чека, ID продукта, количество, промежуточная стоимость | Информация о товарах в заказе |

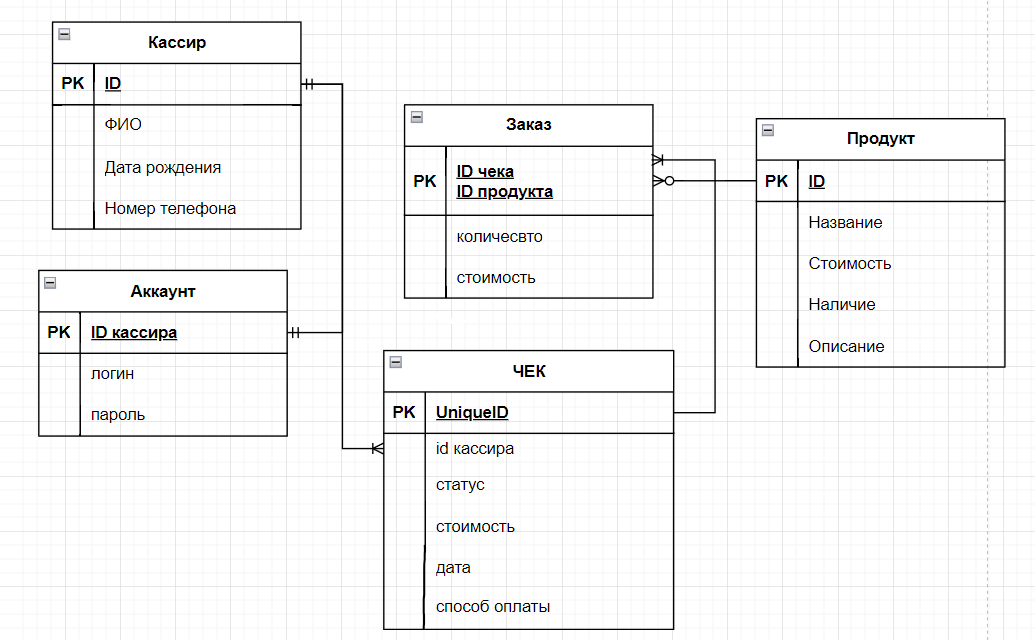


Рисунок 2 – Логическая модель базы данных

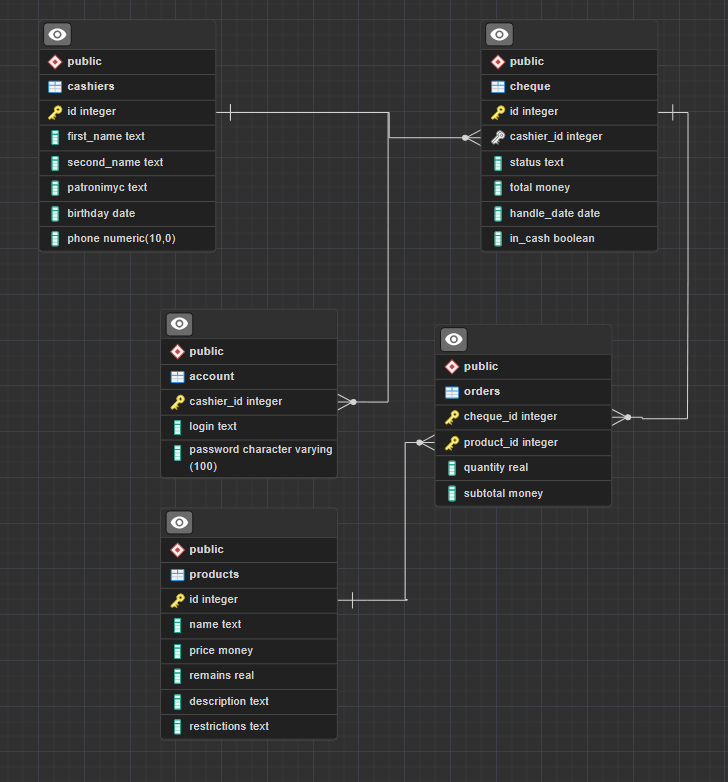


Рисунок 3 – Физическая модель базы данных

Сгенерированный SQL-код для создания реляционной базы данных ПС

CREATE TABLE cashier(

id SERIAL PRIMARY KEY,

first\_name text NOT NULL,

second\_name text NOT NULL,

patronimyc text,

birthdate date NOT NULL,

phone numeric(10,0) NOT NULL

);

CREATE TABLE account(

cashier\_id SERIAL PRIMARY KEY REFERENCES cashier(id),

login text NOT NULL,

password VARCHAR(100) NOT NULL

);

CREATE TABLE products(

id SERIAL PRIMARY KEY,

name text NOT NULL,

price MONEY NOT NULL,

remains REAL,

description text,

restrictions text

);

CREATE TABLE cheque(

id SERIAL PRIMARY KEY,

cashier\_id INT NOT NULL REFERENCES cashiers(id),

status text NOT NULL,

CHECK(status in('created', 'rejected', 'succesfull')),

total MONEY,

handle\_date DATE NOT NULL,

in\_cash BOOLEAN

);

CREATE TABLE orders(

cheque\_id INT NOT NULL REFERENCES cheque(id) ,

product\_id INT NOT NULL REFERENCES products(id),

quantity REAL NOT NULL,

subtotal MONEY,

CONSTRAINT order\_pkey PRIMARY KEY(cheque\_id, product\_id)

);

Для разработки ПС была выбрана СУБД PostgreSQL по следующим причинам:

* поддержка БД неограниченного размера;
* мощные и надёжные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования и поддержка загрузки C‑совместимых модулей;
* наследование;
* легкая расширяемость.

Таблица 3 – структура таблицы cashier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Примечания |
| id | integer | – | NOT NULL, UNIQUE | PRIMARY KEY |
| firstname | text | – | NOT NULL | – |
| secondname | text | – | NOT NULL | – |
| patronimyc | text | – | – | – |
| birthdate | date | – | NOT NULL | – |
| phone | numeric | 10 | NOT NULL | – |

Таблица 4 – структура таблицы account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Примечания |
| cashier\_id | integer | – | NOT NULL, UNIQUE | PRIMARY KEY, FOREIGN KEY |
| login | text | – | NOT NULL | – |
| password | varchar | 100 | NOT NULL | Содержит хэш пароля |

Таблица 5 – структура таблицы product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Примечания |
| id | integer | – | NOT NULL, UNIQUE | PRIMARY KEY |
| name | text | – | NOT NULL | – |
| price | money | – | NOT NULL | – |
| remains | real | – | – | – |
| description | text | – | – | – |
| restrictions | text | – | – | – |

Таблица 6 – структура таблицы cheque

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Примечания |
| id | integer | – | NOT NULL, UNIQUE | PRIMARY KEY |
| cashier\_id | integer | – | NOT NULL | FOREIGN KEY |
| status | text | – | NOT NULL,  status in ('created', 'rejected', 'succesfull') | – |
| total | money | – | – | – |
| handle\_date | date | – | NOT NULL | – |
| in\_cash | boolean | – | – | – |

Таблица 7 – структура таблицы order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Примечания |
| cheque\_id | integer | – | NOT NULL, UNIQUE | PRIMARY KEY  FOREIGN KEY |
| product\_id | integer | – | NOT NULL | PRIMARY KEY  FOREIGN KEY |
| quantity | real | – | NOT NULL,  status in ('created', 'rejected', 'succesfull') | – |
| subtotal | money | – | – | – |

Таблица 8 – Список разработанных таблиц

|  |  |  |
| --- | --- | --- |
| № п/п | Имя таблицы | Описание |
| 1 | cashier | Описывает сущность кассира |
| 2 | account | Содержит данные для входа кассира в систему |
| 3 | product | Описывает сущность продукт |
| 4 | cheque | Описывает сущность чек |
| 5 | order | Содержит список продуктов в заказе |

Таблица 9 – Связи между таблицами БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут | Название | Атрибут |
| cashier | id | account | cashier\_id | один-к-одному |
| cashier | id | cheque | cashier\_id | один-ко-многим |
| cheque | id | order | cheque\_id | один-ко-многим |
| product | id | order | order\_id | один-ко-многим |

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 1 – Структурная схема ПО

Таблица 1 – Перечень разработанных компонентов приложения

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 1 | Program.cs | source | Системный файл, содержащий точку входа в приложение |
| 2 | Account.cs | source | Системный файл, содержащий окно входа в приложение |
| 3 | Account.Designer.cs | form | Системный файл, содержащий настройки внешнего вида и прикрепление событий Account.cs |
| 4 | MainForm.cs | source | Системный файл, содержащий главное окно приложения |
| 5 | MainForm.Designer.cs | form | Системный файл, содержащий настройки внешнего вида и прикрепление событий MainForm.cs |
| 6 | Payment.cs | source | Системный файл, содержащий окно оплаты |
| 7 | Payment.Designer.cs | form | Системный файл, содержащий настройки внешнего вида и прикрепление событий Payment.cs |
| 8 | DBConnection.cs | source | Системный файл, содержащий данные и методы для взаимодействия с базой данных |
| 9 | Cashier.cs | view | Системный файл, содержащий представление для сущности «кассир» |
| 10 | Product.cs | view | Системный файл, содержащий представление для сущности «товар» |
| 11 | ListOfProducts.cs | view | Системный файл, содержащий представление для сущности «список товаров» |
| 12 | Cassa.exe | executable | Исполняемый файл |
| 13 | Npgsql | library | Библиотека, использующая для работы с PostreSQL |

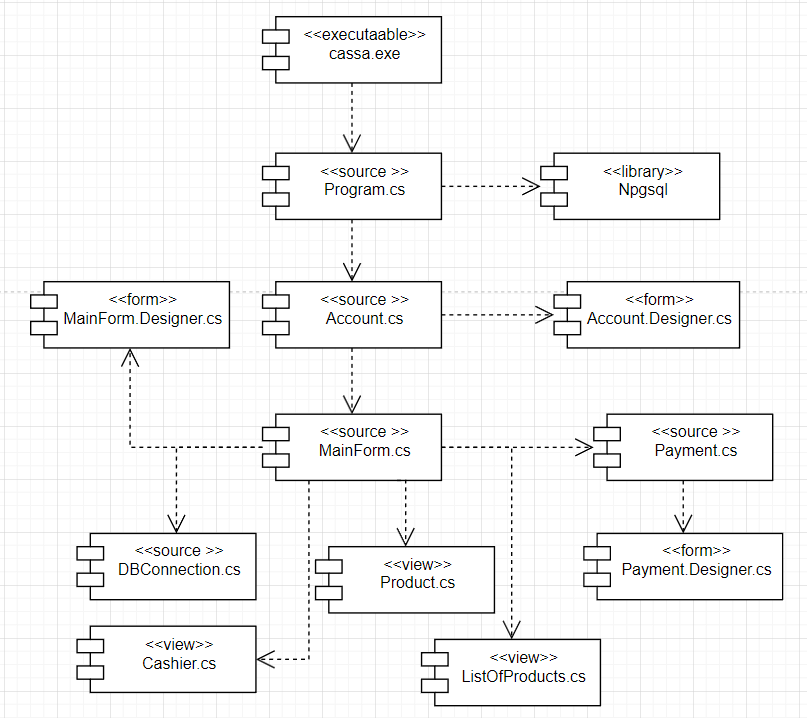


Рисунок 2 – Диаграмма компонентов

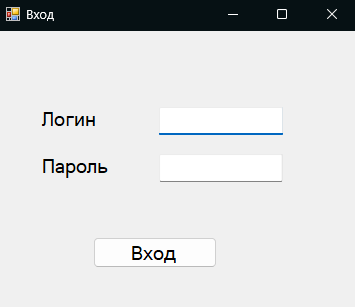


Рисунок 3 – Окно входа в программу

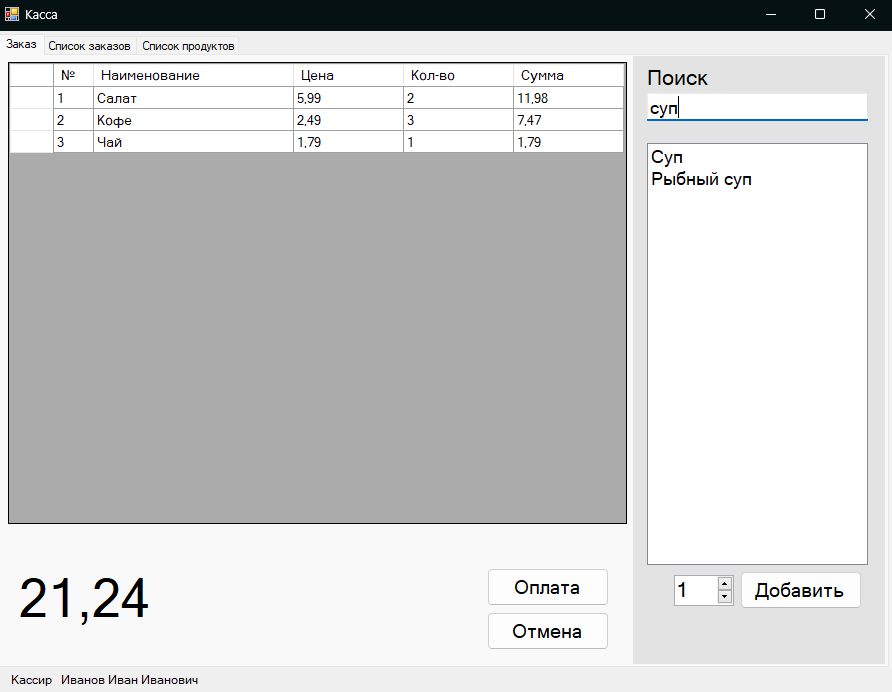


Рисунок 4 – основное окно приложения

В результате анализа кода были получены следующие результаты:

* 8 классов
* 54 функции
* 154 переменные
* 387 операторов

Таблица 2 – Перечень узлов программной системы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 1 | Рабочая станция | processor | Основная рабочая станция пользователя |
| 2 | База данных | database | База данных на рабочей станции |

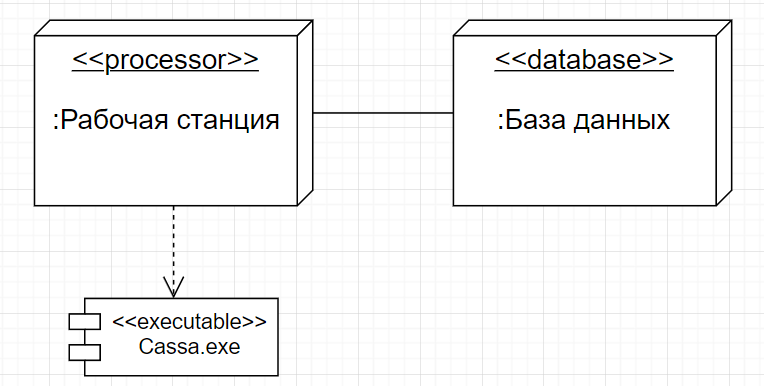


Рисунок 5 – Диаграмма развертывания

**Сценарное тестирование**

Сценарий 1 «Авторизация в системе»

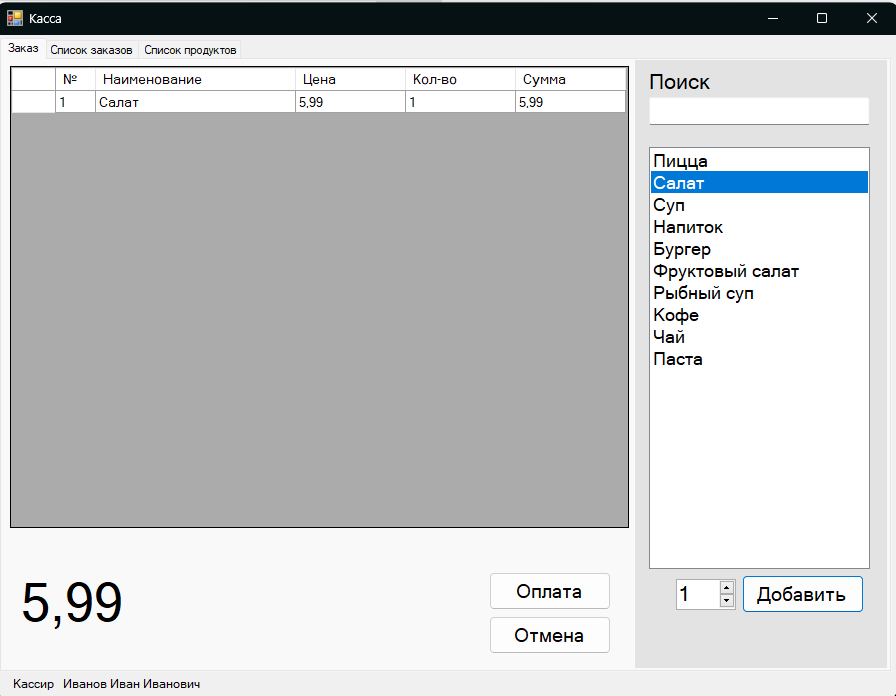
*Предусловие: система содержит данные для авторизации.*

1. Вводим логин и пароль в поля ввода
2. Нажимаем кнопку «войти»
3. Получаем доступ к системе
4. При неправильном вводе логина и/или пароля система должна вывести сообщение об ошибке
5. Если хотя бы одно из полей ввода пустое, система должна вывести сообщение об ошибке

Сценарий 2 «Добавить товар в заказ»

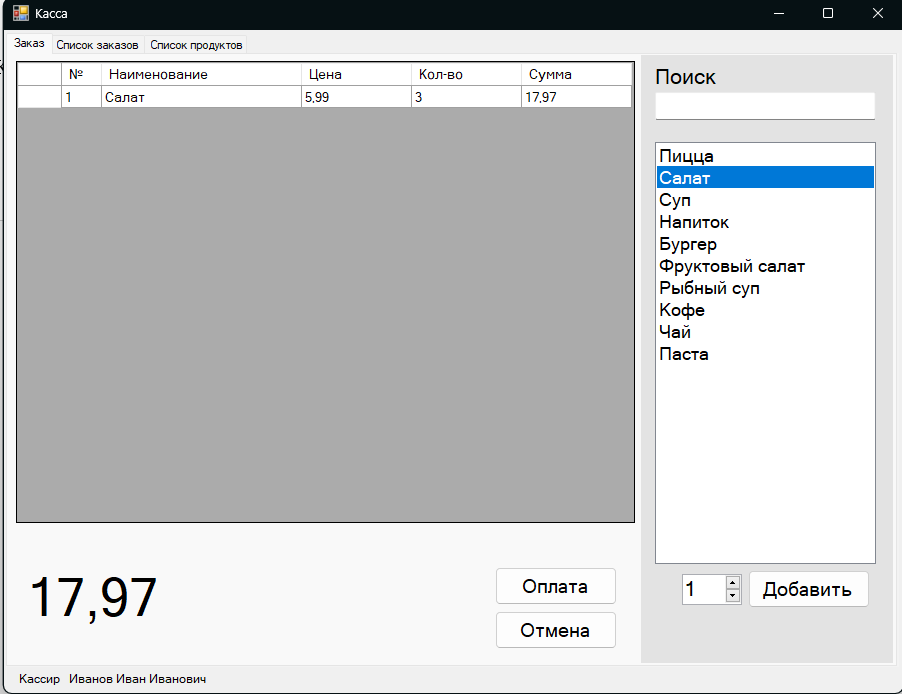
1. Выбираем товар из списка. При необходимости можно использовать строку поиска
2. Вводим количество товара.
3. Нажимаем кнопку «добавить»

Ожидаемый результат: строчка с информацией о товаре добавлена в основную таблицу, итоговая сумма была пересчитана.



Сценарий 3 «Изменить количество товаров»

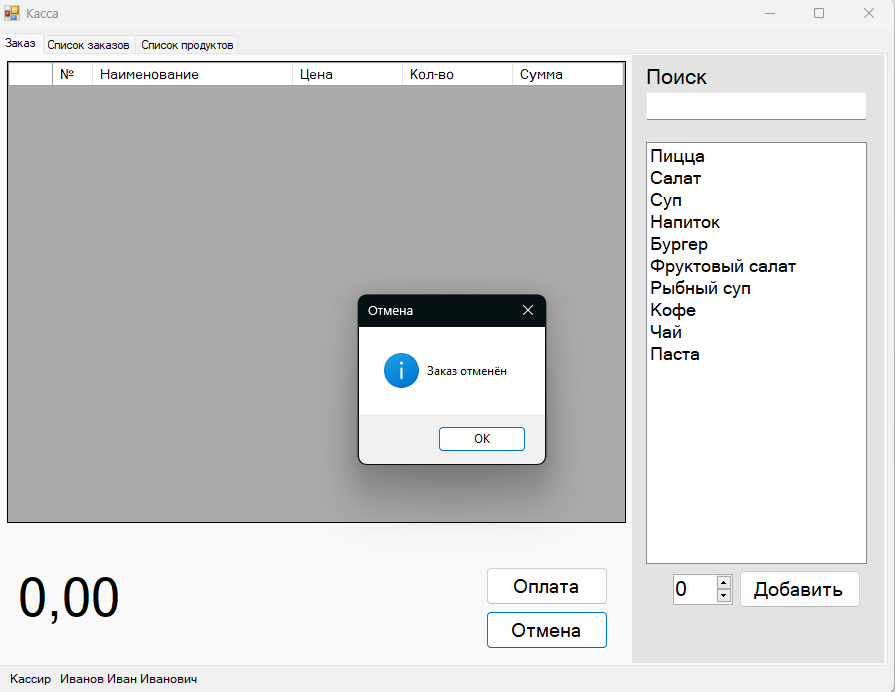
1. Добавляем один или несколько товаров в заказ.
2. Дважды нажимаем на ячейку, содержащую информацию, о количестве товара, которое мы хотим изменить.
3. Вводим новое значение
4. Нажимаем клавишу enter



Сценарий 4 «Отмена заказа»

1. Добавляем один или несколько товаров в заказ.
2. Нажимаем кнопку «отмена».

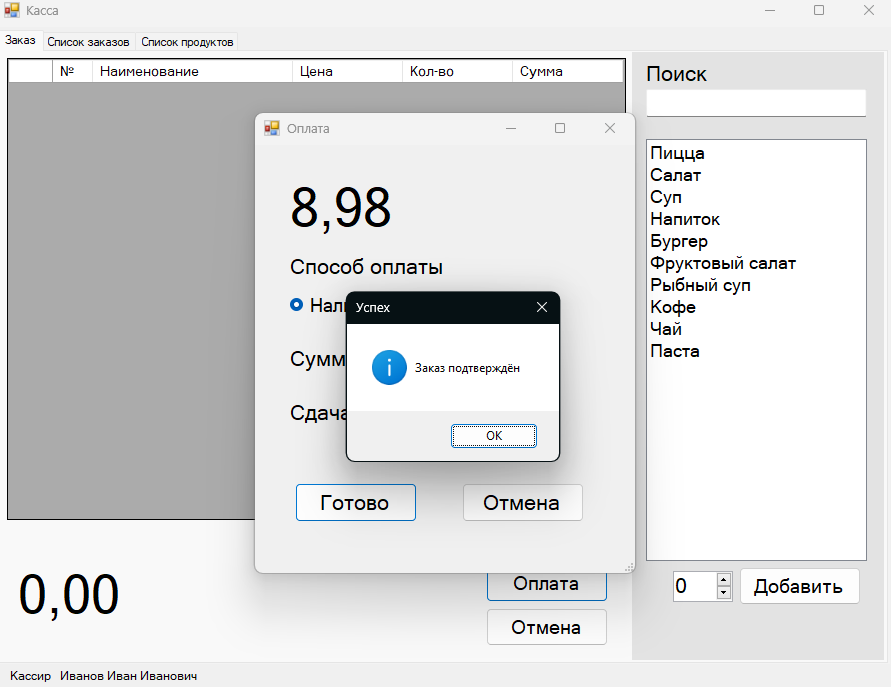
Ожидаемый результат: в базу данных добавляется запись о заказе со статусом «rejected». Выводится сообщение об отмене товара. Основное окно приложения возвращается к первоначальному виду (введённые данные очищаются).



Сценарий 4 «Оплата заказа»

1. Добавляем один или несколько товаров в заказ.
2. Нажимаем кнопку «оплата».
3. Выбираем способ оплаты.
4. При выборе способа оплаты «наличными» вводим внесённую сумму и получаем информацию о размере сдачи.
5. Нажимаем кнопку «Готово».

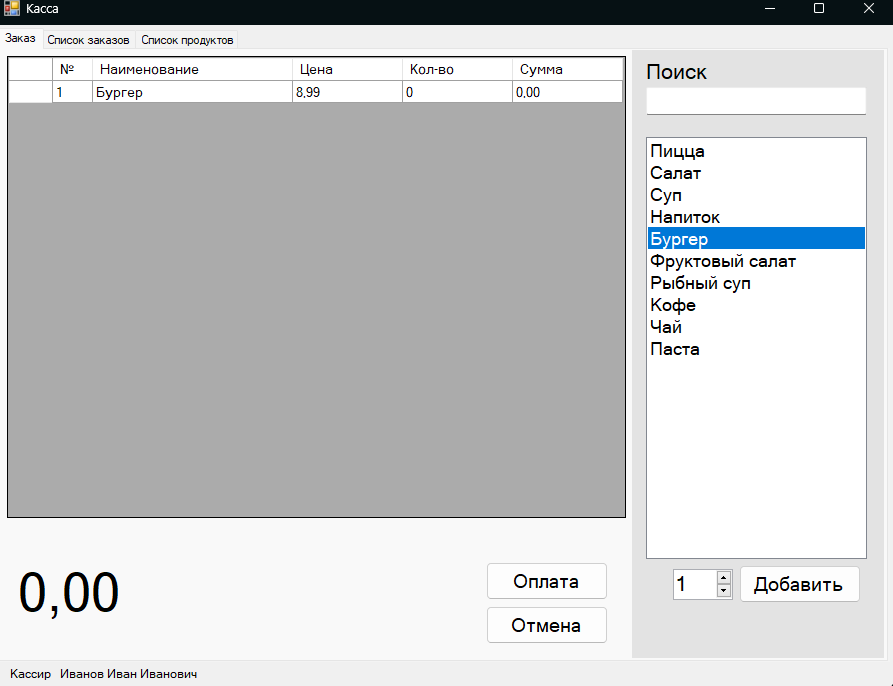
Ожидаемый результат: в базу данных добавляется запись о заказе со статусом «sucsessfull». Выводится сообщение об оплате товара. Основное окно приложения возвращается к первоначальному виду (введённые данные очищаются).



**Документирование логических ошибок**

Логическая ошибка 1:

* Сущность: при изменении количества продуктов на значение 0, в таблице остаётся информация о выбранном продукте. Другими словами, таким образом невозможно удалить позицию из заказа полностью
* Тестовая последовательность манипуляций: в сценарии 3 на шаге 2 вводим значение 0.
* Форма проявления:



* Форма проявления в коде:

private void dataGridView1\_CellValueChanged(object sender, DataGridViewCellEventArgs e)

{

if(dataGridView1.Rows.Count == 0 || e.ColumnIndex != 3)

return;

string name = dataGridView1[1, e.RowIndex].Value.ToString();

double quantity = Convert.ToDouble(dataGridView1[e.ColumnIndex, e.RowIndex].Value);

foreach (var product in products)

{

if (product.Name == name)

{

decimal subtotal = (decimal)quantity \* product.Price;

decimal prevSubtotal = Convert.ToDecimal(dataGridView1[4, e.RowIndex].Value);

dataGridView1[4, e.RowIndex].Value = subtotal;

label4.Text = (Decimal.Parse(label4.Text) - prevSubtotal + subtotal).ToString();

conn.changeOrder(quantity, subtotal, chequeId, product.ID);

dataGridView1.CurrentCell = null;

break;

}

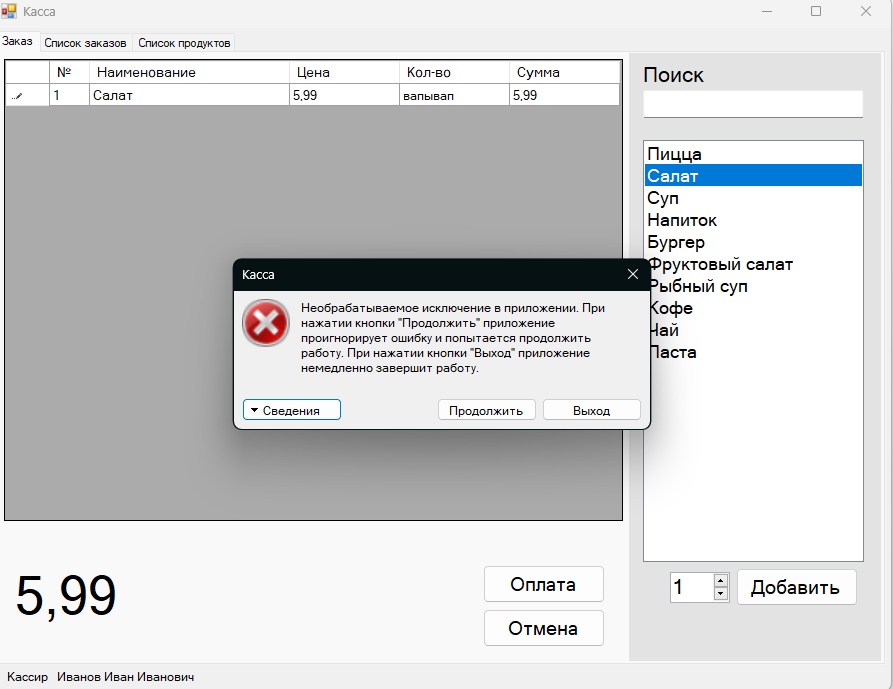
}

}

* Способ устранения: добавить проверку на ввод числа 0. При введение этого числа, удалять всю строку из таблицы.

Логическая ошибка 2:

* Сущность: при изменении количества продуктов можно ввести любую строку в любом формате.
* Тестовая последовательность манипуляций: в сценарии 3 на шаге 2 вводим значение «вапывап».
* Форма проявления:



* Форма проявления в коде:

private void dataGridView1\_CellValueChanged(object sender, DataGridViewCellEventArgs e)

{

if(dataGridView1.Rows.Count == 0 || e.ColumnIndex != 3)

return;

string name = dataGridView1[1, e.RowIndex].Value.ToString();

double quantity = Convert.ToDouble(dataGridView1[e.ColumnIndex, e.RowIndex].Value);

foreach (var product in products)

{

if (product.Name == name)

{

decimal subtotal = (decimal)quantity \* product.Price;

decimal prevSubtotal = Convert.ToDecimal(dataGridView1[4, e.RowIndex].Value);

dataGridView1[4, e.RowIndex].Value = subtotal;

label4.Text = (Decimal.Parse(label4.Text) - prevSubtotal + subtotal).ToString();

conn.changeOrder(quantity, subtotal, chequeId, product.ID);

dataGridView1.CurrentCell = null;

break;

}

}

}

* Способ устранения: добавить проверку на формат вводимой строки. Если вводится не число, не менять значение ячейки.

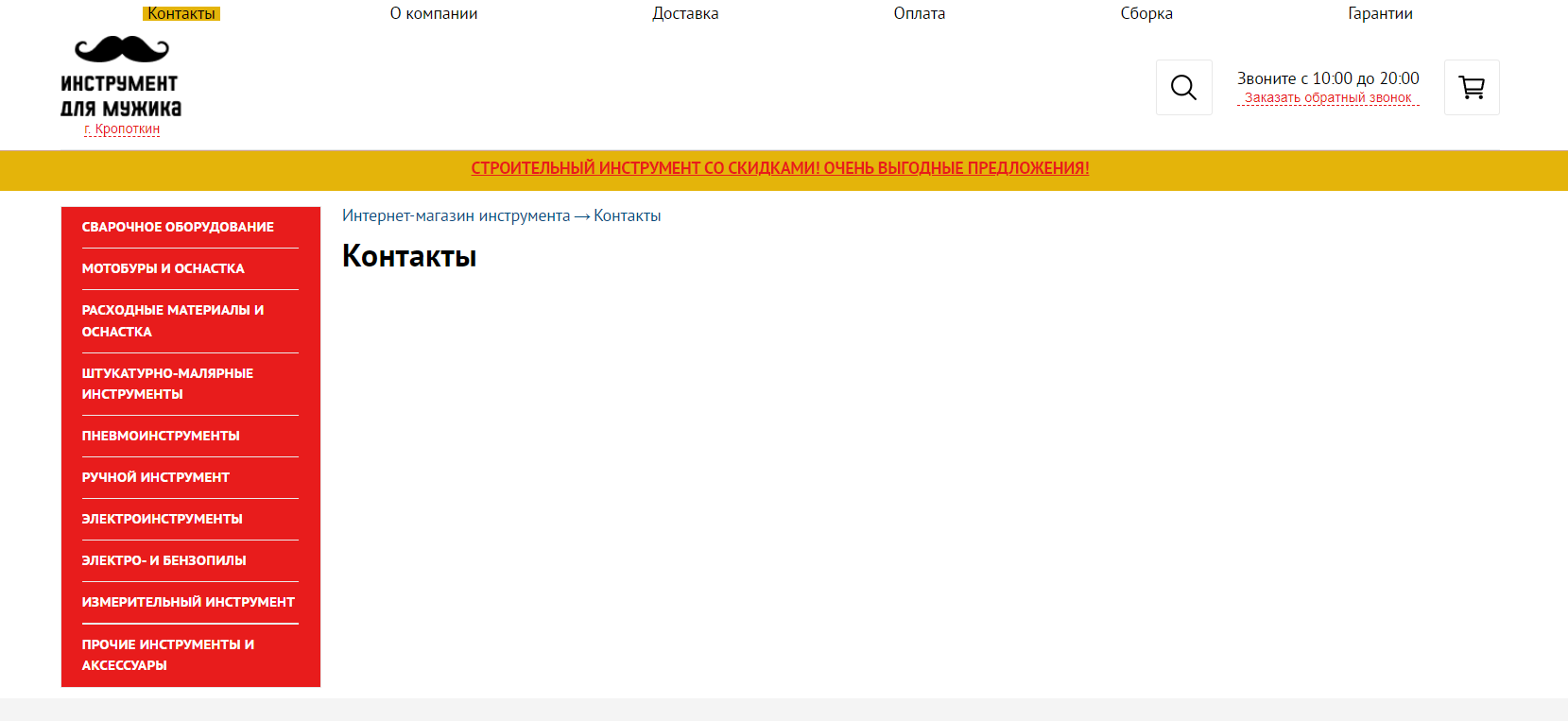
**Сценарное тестирование сайта kropotkin.sale-instrument.ru**

Сценарий 1 «Отображение вкладки контакты»:

1. Открыть сайт https://kropotkin.sale-instrument.ru/
2. Перейти на вкладку «Контакты»

Ожидаемый результат: отображение контактных данных.

Фактический результат:



Сценарий 2 «Поиск по сайту»:

1. Открыть сайт https://kropotkin.sale-instrument.ru/
2. Нажать на кнопку поиска

Ожидаемый результат: пользователю станет доступна строка поиска, не перекрывающая остальной интерфейс сайта

Фактический результат: Ввод в строку поиска недоступен

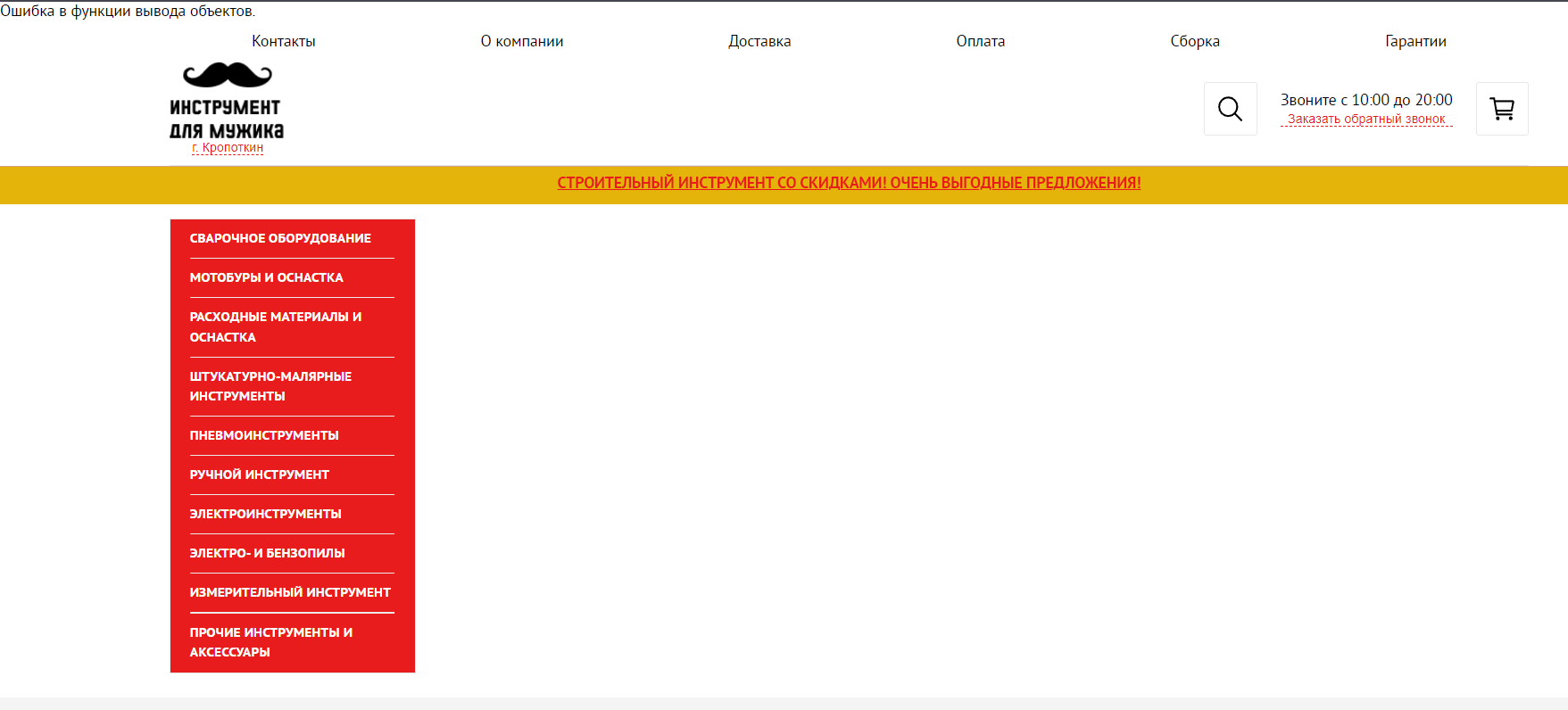


Сценарий 3 «Ввод сортируемой цены»:

1. Открыть сайт https://kropotkin.sale-instrument.ru/
2. Прокрутить страницу до блока «Каталог»
3. Выбрать любую из категорий
4. Выбирать любую из подкатегорий до тех пор, пока не отобразится страница со списком товаров
5. В поле для сортировки по цене ввести любой нечисловой символ
6. Нажать клавишу enter

Ожидаемый результат: сайт обработает ошибочный ввод и не изменит сортировку цены

Фактический результат:



**Нагрузочное тестирование ПС**

Тестирование БД.

В ходе тестирования было измерено время выполнения запросов INSERT, SELECT и TRUNCATE для следующих таблиц: cheque c 100, 500, 1500, 5000 и 10000 строками.

Запрос INSERT тестировался с помощью функций:

CREATE OR REPLACE FUNCTION insert\_random\_cheques(num\_rows INT)

RETURNS VOID AS $$

DECLARE

i INT := 1;

statuses TEXT[] := ARRAY['rejected', 'succesfull', 'created'];

BEGIN

WHILE i <= num\_rows LOOP

INSERT INTO cheque (cashier\_id, status, total, handle\_date, in\_cash)

VALUES (

1,

statuses[1 + (i % 2)],

ROUND((random() \* 1000)::NUMERIC, 2)::MONEY,

current\_date - (random() \* 365)::INT,

random() < 0.5

);

i := i + 1;

END LOOP;

END;

$$ LANGUAGE plpgsql; $$ LANGUAGE plpgsql;

Таблица 1 – время выполнения запросов INSERT

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 4,626 |
| 500 | 17,309 |
| 1500 | 45,808 |
| 5000 | 173,438 |
| 10000 | 340,906 |

Запрос SELECT тестировался с помощью функции:

EXPLAIN ANALYZE SELECT \* FROM cheque

Таблица 2 – время выполнения запросов SELECT

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 0,034 |
| 500 | 0,064 |
| 1500 | 0,156 |
| 5000 | 0,411 |
| 10000 | 1,124 |

Запрос TRUNCATE тестировался с помощью функции:

EXPLAIN ANALYZE TRUNCATE cheque CASCADE

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 20,426 |
| 500 | 25,058 |
| 1500 | 26,843 |
| 5000 | 22,390 |
| 10000 | 23,533 |

Тестирование приложения

В ходе тестирования было измерено время полного запуска приложения в зависимости от количества строк в таблице tourist: 10, 50, 150, 500 и 1000.

Таблица 4 – время полного запуска приложения

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 1238 |
| 500 | 1241 |
| 1500 | 1352 |
| 5000 | 1840 |
| 10000 | 2943 |

# Поиск дефектов ПС

|  |  |
| --- | --- |
| **Шапка** | |
| Короткое описание | Игнорирование интерфейсом изменение размеров окна |
| Проект | Касса |
| Компонент приложения | MainForm.Designer.cs |
| Номер версии | 1.78.420 |
| Серьёзность | S4 Незначительная |
| Приоритет | Р3 Низкий |
| Статус | На стадии исправления |
| Автор | Дакукин И.Е. |
| **Окружение** | |
| ОС | Windows 11 |
| **Описание** | |
| Шаги к воспроизведению | 1. Запустите приложение  --> Приложение открылось  2. Уменьшите или увеличте размер окна |
| Фактический результат | При сжатии окна, интерфейс программы частично скрывается, вплоть до состояния, когда им невозможно пользоваться. При расширении окна, интерфейс программы никак не реагирует, образуется лишнее пустое пространство |
| Ожидаемый результат | Сжатие окна доступа лишь до определённого предела. При расширении окна интерфейс должен адаптивно расширяться. |
| **Дополнения** | |
| Прикрепленный файл |  |

|  |  |
| --- | --- |
| **Шапка** | |
| Короткое описание | Невозможность сортировки по количеству товаров, после изменения одного из значений |
| Проект | Касса |
| Компонент приложения | MainForm.Designer.cs |
| Номер версии | 1.78.420 |
| Серьёзность | S2 Критическая |
| Приоритет | Р1 Высокий |
| Статус | На стадии исправления |
| Автор | Дакукин И.Е. |
| **Окружение** | |
| ОС | Windows 11 |
| **Описание** | |
| Шаги к воспроизведению | 1. Запустите приложение  --> Приложение открылось  2. Нажмите на любой продукт из списка  3. Введите любое количество продукта в поле ввода  4. Нажмите кнопку «Добавить»  --> Продукт добавлен в таблицу  5. Нажмите на любой *другой* продукт из списка  6. Введите любое количество продукта в поле ввода  7. Нажмите кнопку «Добавить»  --> Второй продукт добавлен в таблицу  8. Дважды нажмите на одну из ячеек столбца количество  9. Введите новое значение  10. Нажмите кнопку enter  --> Количество продукта изменено  11. Нажмите на заголовок столбца «количество» |
| Фактический результат | Программа вылетает с ошибкой |
| Ожидаемый результат | Таблица отсортируется относительно столбца «количество» |
| **Дополнения** | |
| Прикрепленный файл |  |

**Поиск дефектов сайта ld-shop.ru**

|  |  |
| --- | --- |
| **Шапка** | |
| Короткое описание | При вводе нечислового значения в поле количества товаров в корзине удаляет товар из корзины |
| Серьёзность | S3 Значительная |
| Приоритет | P2 Средний |
| Автор | Дакукин И. Е. |
| **Окружение** | |
| Браузер + версия | Firefox Версия 120.0.1 |
| ОС | Windows 11 |
| **Описание** | |
| Шаги к воспроизведению | 1. Открыть главную страницу <https://ld-shop.ru/>   --> Сайт открылся   1. Добавить любой товар в корзину   --> Товар добавлен  3. Перейти в корзину  4. Ввести в поле вода «количество» любое нечисловое или же отрицательное значение  5. Нажать клавишу enter |
| Фактический результат | Товар удалится из корзины |
| Ожидаемый результат | Сайт обработает ошибочный ввод и не изменит количество товара |
| **Дополнения** | |
| Прикрепленные файлы |  |

|  |  |
| --- | --- |
| **Шапка** | |
| Короткое описание | Поиск не находит товары которые есть в каталоге по названию |
| Серьёзность | S3 Значительная |
| Приоритет | P1 Высокий |
| Автор | Дакукин И. Е. |
| **Окружение** | |
| Браузер + версия | Firefox Версия 120.0.1 |
| ОС | Windows 11 |
| **Описание** | |
| Шаги к воспроизведению | 1. Открыть главную страницу <https://ld-shop.ru/>   --> Сайт открылся   1. Введите в строку поиска «смеситель» 2. Нажмите клавишу enter |
| Фактический результат | Товары не найдены |
| Ожидаемый результат | Вывод списка смесителей |
| **Дополнения** | |
| Прикрепленные файлы |  |