СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc100851156)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc100851157)

[1.1. Обоснование актуальности задачи 4](#_Toc100851158)

[1.1.1. Актуальность темы 4](#_Toc100851159)

[1.1.2. Анализ существующих разработок по данной теме 4](#_Toc100851160)

[1.2. Обоснование выбора и описание методологии разработки программы решения задачи 5](#_Toc100851161)

[1.3. Обоснование выбора и описание инструментальных средств разработки программы решения задачи 7](#_Toc100851162)

[1.4. Содержательная постановка задачи. 8](#_Toc100851163)

[1.4.1. Описательная модель задачи 8](#_Toc100851164)

[1.4.2. Математическая модель 8](#_Toc100851165)

[1.4.3. Описание входной информации 8](#_Toc100851166)

[1.4.4. Выходная информация 9](#_Toc100851167)

[1.4.5. Описание бизнес-процессов предприятия 9](#_Toc100851168)

[1.4.6. Описание потоков данных предприятия 9](#_Toc100851169)

[1.4.7. Диаграмма вариантов использования 9](#_Toc100851170)

[1.4.8. Диаграмма последовательности 9](#_Toc100851171)

[1.4.9. Диаграмма классов. 9](#_Toc100851172)

[1.4.10. Диаграмма деятельности 9](#_Toc100851173)

[1.4.11. Диаграмма развертывания 9](#_Toc100851174)

[1.4.12. Архитектура приложения 9](#_Toc100851175)

[2. ПРОЕКТНАЯ ЧАСТЬ 10](#_Toc100851176)

[2.1. Разработка структуры данных программы. 10](#_Toc100851177)

[2.2. Логическая модель программы. 14](#_Toc100851178)

[2.3. Разработка пользовательского интерфейса программы 14](#_Toc100851179)

[2.3.1. Разработка требований к интерфейсу 14](#_Toc100851180)

[2.3.2. Проектирование и описание элементов управления 15](#_Toc100851181)

[2.4. Листинг программы 27](#_Toc100851182)

[2.5.Разработка тестовых данных 41](#_Toc100851183)

[2.5.1. Разработка тестов. 42](#_Toc100851184)

[2.6. Результат работы программы 49](#_Toc100851185)

[2.7. Руководство по использования программы 52](#_Toc100851186)

[2.7.1. Руководство системного программиста 52](#_Toc100851187)

[2.6.2. Руководство программиста 52](#_Toc100851188)

[2.5.3. Руководство пользователя 53](#_Toc100851189)

[2.8. Разработка политики безопасности 58](#_Toc100851190)

[2.9. Эксплуатация и развитие программы 59](#_Toc100851191)

[ЗАКЛЮЧЕНИЕ 60](#_Toc100851192)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 61](#_Toc100851193)

[ПРИЛОЖЕНИЕ 1 – IDEF0 63](#_Toc100851194)

[ПРИЛОЖЕНИЕ 2 – DFD 65](#_Toc100851195)

[ПРИЛОЖЕНИЕ 3 – ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ 67](#_Toc100851196)

[ПРИЛОЖЕНИЕ 4 – ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ 68](#_Toc100851197)

[ПРИЛОЖЕНИЕ 5 – ДИАГРАММА КЛАССОВ 69](#_Toc100851198)

[ПРИЛОЖЕНИЕ 6 – ДИГРАММА ДЕЯТЕЛЬНОСТИ 70](#_Toc100851199)

[ПРИЛОЖЕНИЕ 7 – ДИАГРАММА РАЗВЕРТЫВАНИЯ 73](#_Toc100851200)

[ПРИЛОЖЕНИЕ 8 – АРХИТЕКТУРА ПРИЛОЖЕНИЯ 74](#_Toc100851201)

[ПРИЛОЖЕНИЕ 9 – ЛОГИЧЕСКАЯ МОДЕЛЬ 75](#_Toc100851202)

[ПРИЛОЖЕНИЕ 10 – ФИЗИЧЕСКАЯ МОДЕЛЬ 76](#_Toc100851203)

# ВВЕДЕНИЕ

На предприятии для учета нормативно-справочной информации об изделия, введение идет вручную, занося всю информацию на бумажный носитель. Для повышения эффективности работы предприятия необходимо разработать программу, которая автоматизирует и улучшит данный процесс.

Требуется вести информацию об изготовляемых на производстве изделиях, информацию о цене самого изделия.

Для эффективности работы необходимо ведение базы данных, содержащей информацию о ценах, данные об изделиях.

Объектом исследования является справочная информация об изделии.

Предметом исследования является производство изделий, как система учета информации об произведенных изделиях.

Целью работы является разработка приложения для введения нормативно-справочной информации об изделиях, изготавливаемых на предприятии.

Для реализации данной цели необходимо решить следующие задачи:

* добавление информации о ценах;
* добавление информации об изделиях;
* формирование справочной информации изделий;
* формирование графика.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1. Обоснование актуальности задачи

### 1.1.1. Актуальность темы

В настоящим времени приложения для предприятия являются весьма актуальными. Для любых предприятий нужна разработанная база данных для автоматизации работы по производству и учёту в данной области. В задаче автоматизируется учет нормативно-справочной информации, изготовляемых предприятием. Такой учет необходим для быстрой работы предприятия по отслеживанию полной информации об изделиях, в которую входят данные о самом изделии, его цехе производства, цене изделия и дате установки цены.

Автоматизировав процесс, предприятие с легкостью сможет узнать необходимую информацию. Быстрая коммуникация между отделами упростит работу ремонтной компании.

### 1.1.2. Анализ существующих разработок по данной теме

«ПОЛИНОМ:MDM» - программа по учету производства предприятия. Оно позволяет управлять нормативно-справочной информацией промышленного предприятия, применяя единые инструменты, подходы и методы для работы с данными. Автоматическое заполнение и печать документов.

«Supasoft CRM» - универсальная программа для учета нормативно-справочной информации предприятия. Сетевая версия позволит нескольким пользователям работать в единой базе с разными правами доступа. Автоматическое заполнение и печать документов.

«1С:Предприятие» - справочная программа для предприятия. Разработана компанией 1С. построена по принципу гипертекста, аналогично страницам в Интернете, ссылающимся друг на друга.

Плюсы и минусы этих систем представлены в таблице 1.

Таблица 1 – плюсы и минусы систем автоматизации процесса.

|  |  |  |
| --- | --- | --- |
| Наименование | Плюсы | Минусы |
| «ПОЛИНОМ:MDM» | Широкий спектр информации в рамках единой программной платформы | Является платным программным продуктом |
| Широкий функционал использования | Возможны подвисания |
| Имеет простой интерфейс | Нагружает работу компьютера |
| «Supasoft CRM» | Достаточная проста в эксплуатации | Полное программное обеспечения платное |
| Универсальная программа | Замедление работы СУБД при большом количестве информации |
| Модифицируемое приложение | Возможна нагрузка на работу компьютера |

|  |  |  |
| --- | --- | --- |
| «1С:Предприятие» | Широкий функционал | Платная программа |
| Техподдержка приложения | Сложный в понимании интерфейс |
| Удобная интеграция | Плохая оптимизация на старых устройствах |

Вывод: проанализировав данные представление в таблице, необходимо разработать программный продукт, который будет повторять плюсы представленных программ. Приложение должно иметь широкий функционал, должна быть иметь понятный обычному пользователю интерфейс. Необходимо решить такие минусы, как нагрузку на работу компьютера и отсутствие зависания БД при большом хранении информации.

## 1.2. Обоснование выбора и описание методологии разработки программы решения задачи

Каскадная модель – модель процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.

Следуя каскадной модели, разработчик переходит от одной стадии к другой строго последовательно.

Тем самым, каскадная модель подразумевает, что переход от одной фазы разработки к другой происходит только после полного и успешного завершения предыдущей фазы, и что переходов назад либо вперёд или перекрытия фаз — не происходит.

V-образная модель применима к системам, которым особенно важно бесперебойное функционирование. Особенностью модели можно считать то, что она направлена на тщательную проверку и тестирование продукта, находящегося уже на первоначальных стадиях проектирования. Стадия тестирования проводится одновременно с соответствующей стадией разработки.

В инкрементной модели полные требования к системе делятся на различные сборки. Терминология часто используется для описания поэтапной сборки ПО. Цикл разделен на более мелкие легко создаваемые модули. Каждый модуль проходит через фазы определения требований, проектирования, кодирования, внедрения и тестирования.

В RAD-модели компоненты или функции разрабатываются несколькими высококвалифицированными командами параллельно, будто несколько мини-проектов. Временные рамки одного цикла жестко ограничены. Созданные модули затем интегрируются в один рабочий прототип.

Проанализировав данные методологии по их описанию, была выбрана каскадная модель, так как все этапы происходят в строго последовательном порядке и решение задачи переходит на следующий этап в том случае, если предыдущий был завершен.

## 1.3. Обоснование выбора и описание инструментальных средств разработки программы решения задачи

Microsoft Visual Studio 2019 — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию.

Visual Studio Code (VS Code) — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

В качестве средства разработки клиентской части ПО использовалась среда программирования Microsoft Visual Studio 2019. Среда разработана в соответствии с концепцией визуального программирования.

В качестве языка программирования клиентской части данного проекта был выбран язык C#.

C# является языком программирования, который разработан для создания множества приложений, работающих в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. C# упрощает разработку компонентов программного обеспечения инкапсулированные сигнатуры методов, называемые делегатами, которые поддерживают типобезопасные уведомления о событиях;

Выбор СУБД

MySQL — это реляционная система управления базами данных с открытым исходным кодом. В настоящее время эта СУБД одна из наиболее популярных в веб-приложениях — подавляющее большинство CMS использует именно MySQL, а почти все веб-фреймворки поддерживают MySQL уже на уровне базовой конфигурации (без дополнительных модулей).

Решение Microsoft SQL Server 2019 представляет собой высокопроизводительную платформу обработки и анализа данных для бизнеса любого размера, которая отвечает всем современным требованиям по работе с данными любых типов и поддержке удобной и быстрой разработки приложений. Продукт реализует концепцию комплексной обработки данных и позволяет преобразовывать ценную информацию в знания.

Поэтому в качестве СУБД используется Microsoft SQL Server 2019.

## 1.4. Содержательная постановка задачи.

### 1.4.1. Описательная модель задачи

Приложение для учета нормативно-справочной информации об изделиях, изготавливаемых на предприятии должно хранить в себе определенную информацию по изделиям и их цену. Данная информация необходима инженеру-технологу для формирования справочной информации по готовому продукту.

### 1.4.2. Математическая модель

В данной задачи математическая формула отсутствует.

### 1.4.3. Описание входной информации

Описание входной информации представлены в таблицах 2-3.

Таблица 2 – Входная информация.

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Источник данных |
| Код изделия | CodeProduct | Числовой | «Справочник изделия» |
| Код организации | CodeOrg | Числовой | «Справочник изделия» |
| Код характеристики | CodeCharac | Числовой | «Справочник изделия» |
| Номер по порядку | SerialNum | Числовой | «Справочник изделия» |
| Наименование | NameProduct | Текстовый | «Справочник изделия» |
| Цех | Name | Текстовый | «Справочник изделия» |
| Фото | Photo | Бинарный | «Справочник изделия» |

Таблица 3 – Входная информация

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Источник данных |
| № по порядку | NumPP | Числовой | «Прайс-лист деталей» |
| Дата установки | DateInsert | дата | «Прайс-лист деталей» |
| Деталь | NameProduct | Текстовый | «Прайс-лист деталей» |

Продолжение таблицы 3.

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Источник данных |
| Цена | Cost | Числовой | «Прайс-лист деталей» |
| № договора | NumContract | Числовой | «Прайс-лист деталей» |

### 1.4.4. Выходная информация

Описание выходной информации представлено в таблице 4.

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Источник данных |
| Код организации | CodeOrg | Числовой | «Паспорт изделия» |
| Код характеристики | CodeCharac | Числовой | «Паспорт изделия» |
| Порядковый № | SerialNum | Числовой | «Паспорт изделия» |
| Наименование | NameProduct | Текстовый | «Паспорт изделия» |
| Цена | Cost | Числовой | «Паспорт изделия» |
| Дата установки | DateInsert | Дата | «Паспорт изделия» |
| Цех | Name | Текстовый | «Паспорт изделия» |

### 1.4.5. Описание бизнес-процессов предприятия

Бизнес-процесс IDEF0 с детализацией представлен в приложении 1.

### 1.4.6. Описание потоков данных предприятия

Потоки данных DFD с детализацией представлены в приложении 2.

### 1.4.7. Диаграмма вариантов использования

Диаграмма вариантов использования представлена в приложении 3.

### 1.4.8. Диаграмма последовательности

Диаграмма последовательности представлена в приложении 4.

### 1.4.9. Диаграмма классов.

Диаграмма классов представлена в приложении 5

### 1.4.10. Диаграмма деятельности

Диаграммы деятельности представлена в приложении 6.

### 1.4.11. Диаграмма развертывания

Диаграмма развертывания представлена в приложении 7.

### 1.4.12. Архитектура приложения

Архитектура приложения представлена в приложении 8.

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1. Разработка структуры данных программы.

Описание структуры таблиц представлены в таблицах 5-14

Таблица 5 – Описание таблицы «Единица измерения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| № ед. измерения | NumUnit | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № единицы измерения |
| Значение | NameUnit | Текстовый | Размер – 10 символов  Обязательное поле  Не может быть пустым | Значение единицы измерения |

Таблица 6 – Описание таблицы «Цех»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| № цеха | NumDep | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № цеха |
| Название | Name | Текстовый | Размер – 28 символов  Обязательное поле  Не может быть пустым | Наименование цеха |

Таблица 7 – Описание таблицы «Операции»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код операции | CodeOperation | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код операции |
| Наименование | Name | Текстовый | Размер – 40 символов  Обязательное поле  Не может быть пустым | Наименование операции |

Таблица 8 – Описание таблицы «Должность»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код должности | CodePost | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код должности |
| Название | NamePost | Текстовый | Размер – 28 символов  Обязательное поле  Не может быть пустым | Наименование должности |
| Ставка за час | SalaryPerHours | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Ставка за час |

Таблица 9 – Описание таблицы «Пользователь»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| ID | ID | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | ID пользователя |
| Логин | Login | Текстовый | Размер – 20 символов  Обязательное поле  Не может быть пустым | Логин пользователя |
| Пароль | Password | Текстовый | Размер – 20 символов  Обязательное поле  Не может быть пустым | Пароль пользователя |
| Код должности | CodePost | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код должности |

Таблица 10 – Описание таблицы «Материал»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код материала | CodeMaterial | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код материала |
| Наименование | Name | Текстовый | Размер – 20 символов  Обязательное поле  Не может быть пустым | Наименование материала |
| Цена материала | PriceMaterial | Текстовый | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Цена материала |

Продолжение таблицы 10.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| № ед. измерения | NumUnit | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № единицы измерения |

Таблица 11 – Описание таблицы «Изделие»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код изделия | CodeProduct | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код изделия |
| Код организации | CodeOrg | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код организации |
| Код характеристики | CodeCharac | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код характеристики |
| Порядковый № | SerialNum | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Порядковый № |
| Наименование | NameProduct | Текстовый | Размер – 20 символов  Обязательное поле  Не может быть пустым | Наименование изделия |
| № цеха | NumDep | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № цеха производства |
| Фото | Photo | Бинарный | Размер поля – длинное поле  Может быть пустым | Фото изделия |

Таблица 12 – Описание таблицы «Цена»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| № по порядку | NumPP | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № по порядку |
| Дата установки | DateInsert | Дата | Первичный ключ  Обязательное поле | Дата установки цены |

Продолжение таблицы 12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код изделия | CodeProduct | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код изделия |
| Значение | Cost | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Цена детали |
| № договора | NumContract | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № договора |

Таблица 13 – Описание таблицы «Материальная карта»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| № материальной карты | NumMatCard | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | № материальной карты |
| Код материала | CodeMaterial | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код материала |
| Код изделия | CodeProdct | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код изделия |
| Количество | Quantity | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Количество |

Таблица 14 – Описание таблицы «Карта трудоемкости»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код карты трудоемкости | CodeCardWork | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код карты трудоемкости |

Продолжение таблицы 14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код изделия | CodeProduct | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код изделия |
| Код должности | CodePost | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код должности |
| Код операции | CodeOperation | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код операции |
| Порядок операции | OperarionOrder | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Порядок операции |
| Затраченное время | ElapsedTime | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Затраченное время |

## 2.2. Логическая модель программы.

Логическая модель представлена в приложении 9.

Физическая модель представлена в приложении 10.

## 2.3. Разработка пользовательского интерфейса программы

### 2.3.1. Разработка требований к интерфейсу

Основные требования к пользовательскому интерфейсу:

* понятность и логичность;
* обеспечение высокой скорости работы пользователя;
* обеспечение защиты от ошибок;
* приятная цветовая гамма

### 2.3.2. Проектирование и описание элементов управления

Макет основного окна представлен на рисунке 1.

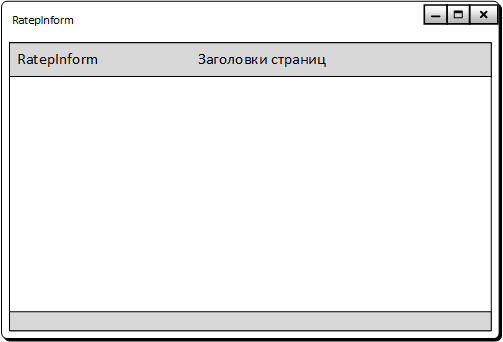


Рис. 1 – Макет основного окна

Интерфейс основного окна представлен на рисунке 2.

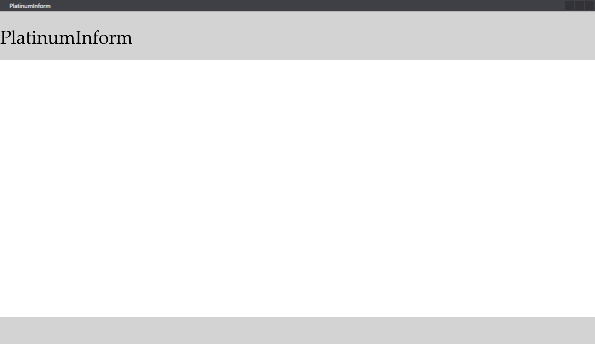


Рис. 2 – Интерфейс основного окна

Таблица 15 – Свойства и компоненты основного окна

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="#FFE9E9FD" |
| Grid | - | Grid.ColumnSpan="3" Background="LightGray" |

Продолжение таблицы 15.

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Image | - | HorizontalAlignment="Left" |
| TextBlock | - | Text="{Binding ElementName=MainFrame, Path=Content.Title}" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="29" |
| Frame | MainFrame | NavigationUIVisibility="Hidden" Grid.Column="1" Grid.Row="1" |

Макет страницы «Авторизация» представлен на рисунке 3

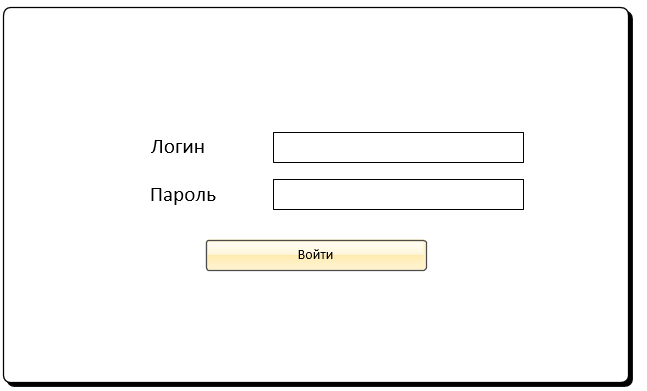


Рис. 3 – макет страницы «Авторизация»

Интерфейс страницы «Авторизация» представлен на рисунке 4.

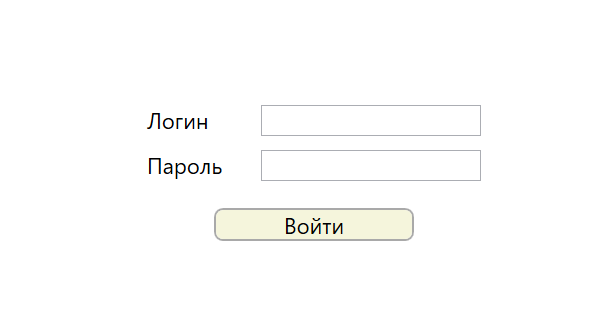


Рис. 4 – интерфейс страницы «Авторизация»

Таблица 16 – Свойства и компоненты страницы «Авторизация»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | HorizontalAlignment="Center" VerticalAlignment="Center" |
| StackPanel | - | Orientation="Horizontal" |
| TextBlock | - | Text="Логин" FontSize="22" Width="100" Margin="7" |
| TextBox | tbLogin | FontSize="22" Width="220" Margin="7" |
| TextBlock | - | Text="Пароль" FontSize="22" Width="100" Margin="7" |
| PasswordBox | tbPassword | FontSize="22" Width="220" Margin="7" |
| Button | btnEnter | Content="Войти" FontSize="22" Width="200" Margin="0,20,0,0" Click="btnEnter\_Click" |

Макет страницы «Меню» представлен на рисунке 5.

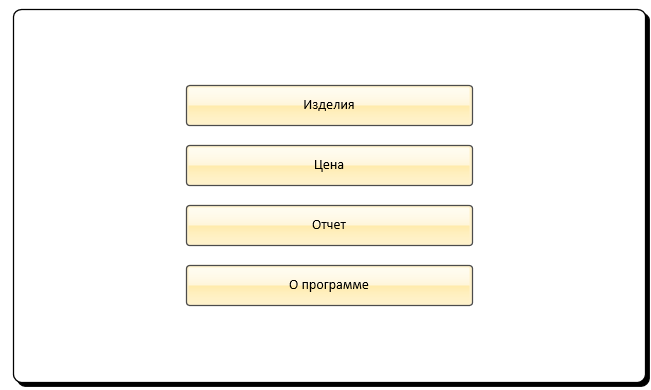


Рис. 5 – макет страницы «Меню»

Интерфейс страницы «Меню» представлен на рисунке 6.

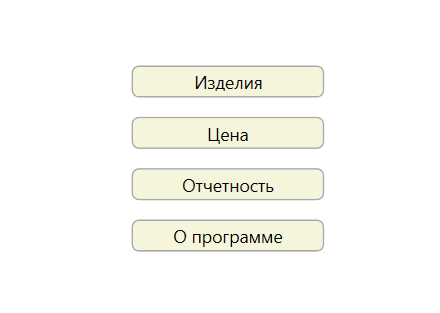


Рис. 6 – интерфейс страницы «Меню»

Таблица 17 – Свойства и компоненты страницы «Меню»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | HorizontalAlignment="Center" VerticalAlignment="Center" |
| Button | btnProduct | Content="Изделия" FontSize="22" Width="240" Height="40" Margin="12" Click="btnProduct\_Click" |
| Button | btnPrice | Content="Цена" FontSize="22" Width="240" Height="40" Margin="12" Click="btnPrice\_Click" |
| Button | btnReport | Content="Отчетность" FontSize="22" Width="240" Height="40" Margin="12" Click="btnReport\_Click" |
| Button | btnAbout | Content="О программе" FontSize="22" Width="240" Height="40" Margin="12" Click="btnAbout\_Click" |

Макет страницы «Изделия» представлен на рисунке 7.

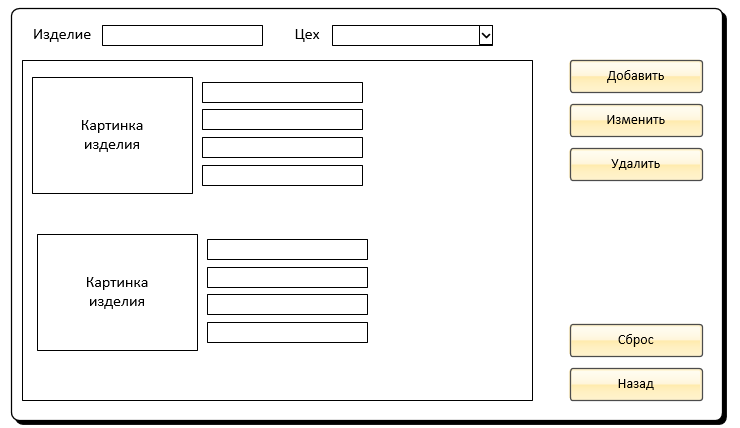


Рис. 7 – макет страницы «Изделия»

Интерфейс страницы «Изделия» представлен на рисунке 8

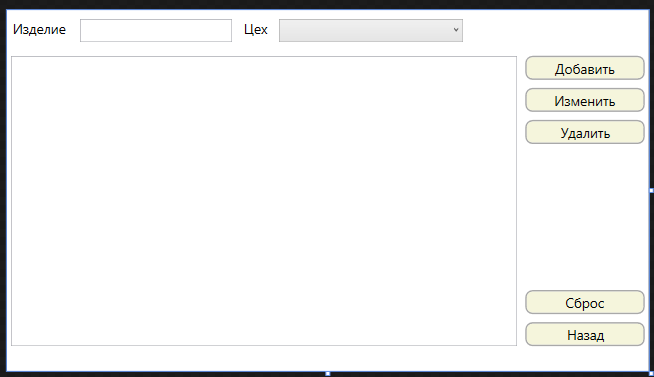


Рис. 8 – Интерфейс страницы «Изделия»

Таблица 18 – Свойства и компоненты страницы «Изделия»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | Orientation="Horizontal" VerticalAlignment="Center" |
| TextBlock | - | Text="№ изделия" FontSize="17" Width="120" Margin="7" |
| TextBox | tbSearch | FontSize="17" Width="190" Margin="7" TextChanged="tbSearch\_TextChanged" |
| ListView | lwProduct | Grid.Row="1" Margin="5" ScrollViewer.VerticalScrollBarVisibility="Hidden" |
| Border | - | BorderBrush="Black" BorderThickness="2" |
| StackPanel | - | Orientation="Horizontal" |
| Border | - | BorderBrush="Black" BorderThickness="1" Margin="5" |
| Image | - | Width="200" |
| TextBlock | - | Text="Код изделия" FontSize="16" Width="140" Margin="5" |
| TextBlock | tbCode | Text="{Binding CodeProduct}" FontSize="16" Width="220" Margin="5" |
| TextBlock | - | Text="Децимальный №" FontSize="16" Width="140" Margin="5" |

Продолжение таблицы 18

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| TextBlock | tbDecNum | FontSize="16" Width="220" Margin="5" |
| TextBlock | - | Text="Наименование" FontSize="16" Width="140" Margin="5" |
| TextBlock | tbName | Text="{Binding NameProduct}" FontSize="16" Width="220" Margin="5" |
| TextBlock | - | Text="Цех" FontSize="16" Width="140" Margin="5" |
| TextBlock | tbManuf | Text="{Binding Departament.Name}" FontSize="16" Width="220" Margin="5" |
| StackPanel | - | Grid.Row="1" Grid.Column="1" VerticalAlignment="Top" |
| Button | btnAdd | Content="Добавить" FontSize="17" Height="30" Margin="5" Click="btnAdd\_Click" |
| Button | btnEdit | Content="Изменить" FontSize="17" Height="30" Margin="5" Click="btnEdit\_Click" |
| Button | btnDel | Content="Удалить" FontSize="17" Height="30" Margin="5" Click="btnDel\_Click" |
| Button | btnBack | Content="Назад" Grid.Column="1" Grid.Row="1" VerticalAlignment="Bottom" FontSize="17" Height="30" Margin="5" Click="btnBack\_Click" |
| Button | btnReset | Content="Сброс" FontSize="17" Height="30" Margin="5" Click="btnReset\_Click" |

Макет страницы «Добавление/изменение данных» представлен на рисунке 9.

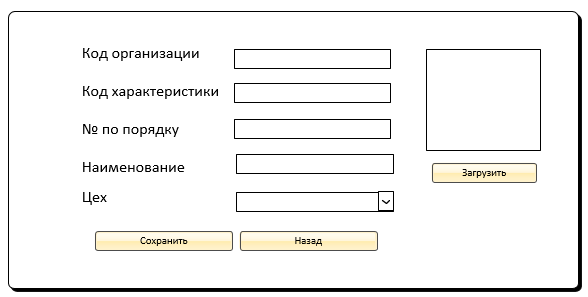


Рис. 9 – макет страницы «Добавление/изменение данных»

Интерфейс страницы «Добавление/изменение данных» представлен на рисунке 10.

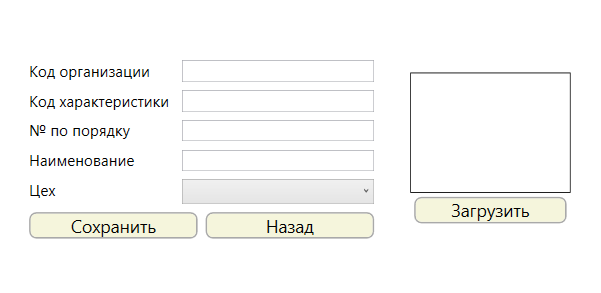


Рис. 10 – интерфейс страницы «Добавление/изменение данных»

Таблица 19 – Свойства и компоненты страницы «Добавление/изменение данных».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | HorizontalAlignment="Center" VerticalAlignment="Center" Orientation="Horizontal" |
| StackPanel | - | Orientation="Vertical" |
| StackPanel | - | Orientation="Horizontal" |
| TextBlock | - | Text="Код организации" Width="180" Margin="5" FontSize="19" |

Продолжение таблицы 19

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| TextBox | tbOrganization | Width="240" Margin="5" FontSize="19" PreviewTextInput="tbOrganization\_PreviewTextInput" |
| TextBlock | - | Text="Код характеристики" Width="180" Margin="5" FontSize="19" |
| TextBox | tbCharacteristic | Width="240" Margin="5" FontSize="19" PreviewTextInput="tbOrganization\_PreviewTextInput" |
| TextBlock | - | Text="№ по порядку" Width="180" Margin="5" FontSize="19" |
| TextBox | tbNumPP | Width="240" Margin="5" FontSize="19" PreviewTextInput="tbOrganization\_PreviewTextInput" |
| TextBlock | - | Text="Наименование" Width="180" Margin="5" FontSize="19" |
| TextBox | tbDetail | Margin="5" FontSize="19" PreviewTextInput="tbDetail\_PreviewTextInput" |
| TextBlock | - | Text="Цех" Width="180" Margin="5" FontSize="19" |
| ComboBox | cbDepartament | Width="240" Margin="5" FontSize="19" DisplayMemberPath="Name" |
| Button | btnSave | Content="Сохранить" Width="210" FontSize="22" Margin="5" Click="btnSave\_Click" |
| Button | btnCancel | Content="Назад" Width="210" FontSize="22" Margin="5" Click="btnCancel\_Click" |
| Image | imgPhoto | - |
| Button | btnLoad | Content="Загрузить" FontSize="22" Margin="5" Click="btnLoad\_Click" |

Макет страницы «Цена» представлена на рисунке 11.

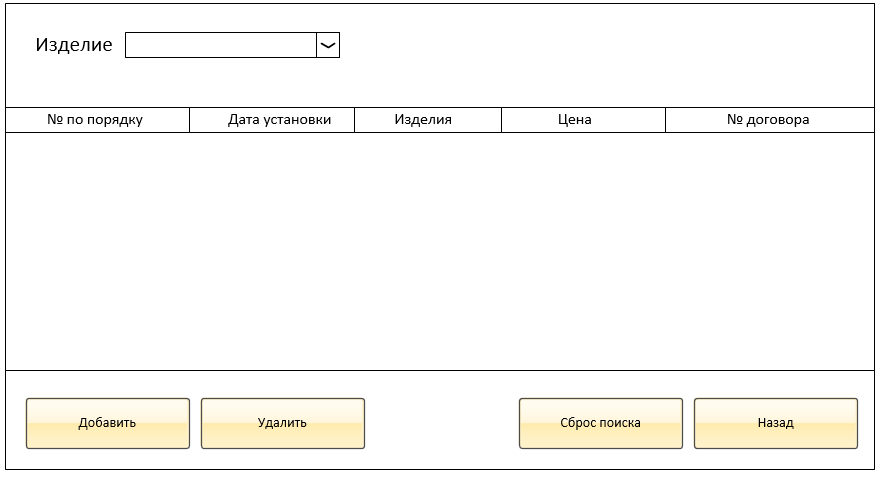


Рис. 11 – макет страницы «Цены»

Интерфейс страницы «Цена» представлен на рисунке 12.

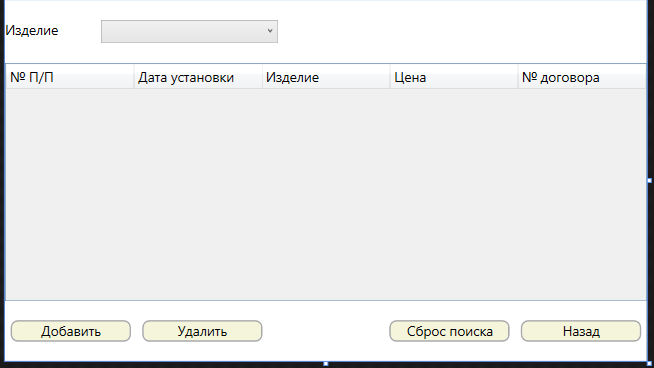


Рис. 12 – интерфейс страницы «Цена»

Таблица 20 – Свойства и компоненты страницы «Цена»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | VerticalAlignment="Center" Orientation="Horizontal" HorizontalAlignment="Left" |
| TextBlock | - | Text="Изделие" FontSize="17" Width="120" |

Продолжение таблицы 20.

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| ComboBox | cbProduct | FontSize="17" Width="220" DisplayMemberPath="NameProduct" SelectionChanged="cbProduct\_SelectionChanged" |
| DataGrid | dgPrice | Grid.Row="1" FontSize="17" AutoGenerateColumns="False" |
| StackPanel | - | Orientation="Horizontal" Grid.Row="2" HorizontalAlignment="Left" VerticalAlignment="Center" |
| Button | btnAdd | Content="Добавить" FontSize="17" Width="150" Margin="7" Click="btnAdd\_Click" |
| Button | btnDel | Content="Удалить" FontSize="17" Width="150" Margin="7" Click="btnDel\_Click" |
| StackPanel | - | Orientation="Horizontal" Grid.Row="2" HorizontalAlignment="Right" VerticalAlignment="Center" |
| Button | btnReset | Content="Сброс поиска" FontSize="17" Width="150" Margin="7" Click="btnReset\_Click" |
| Button | btnBack | Content="Назад" FontSize="17" Width="150" Margin="7" Click="btnBack\_Click" |

Макет страницы «Добавление данных» представлена на рисунке 13.

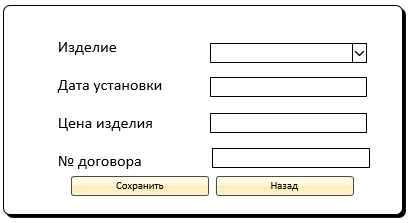


Рис. 13 – макет страницы «Добавление данных»

Интерфейс страницы «Добавление данных» представлен на рисунке 14.

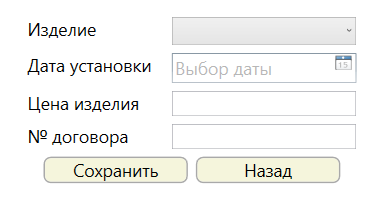


Рис. 14 – интерфейс страницы «Добавление данных»

Таблица 21 – Свойства и компоненты страницы «Добавление данных»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | HorizontalAlignment="Center" VerticalAlignment="Center" Orientation="Horizontal" |
| StackPanel | - | Orientation="Horizontal" |
| TextBlock | - | Text="Изделие" Width="170" FontSize="22" Margin="5" |
| ComboBox | cbDetail | Width="230" FontSize="22" Margin="5" DisplayMemberPath="NameProduct" |
| TextBlock | - | Text="Дата установки" Width="170" FontSize="22" Margin="5" |
| DatePicker | dpDateInsert | Width="230" FontSize="22" Margin="5" IsEnabled="False" |
| TextBlock | - | Text="Цена изделия" Width="170" FontSize="22" Margin="5" |
| TextBox | tbCost | Width="230" FontSize="22" Margin="5" PreviewTextInput="tbCost\_PreviewTextInput" TextChanged="tbCost\_TextChanged" |
| TextBlock | - | Text="№ договора" Width="170" FontSize="22" Margin="5" |
| TextBox | tbNumContract | Width="230" FontSize="22" Margin="5" PreviewTextInput="tbCost\_PreviewTextInput" |

Продолжение таблицы 21.

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Button | btnSave | Content="Сохранить" Width="180" Margin="5" FontSize="22" Click="btnSave\_Click" |
| Button | btnCancel | Content="Назад" Width="180" Margin="5" FontSize="22" Click="btnCancel\_Click" |

Макет страницы «Отчетность» представлена на рисунке 15.

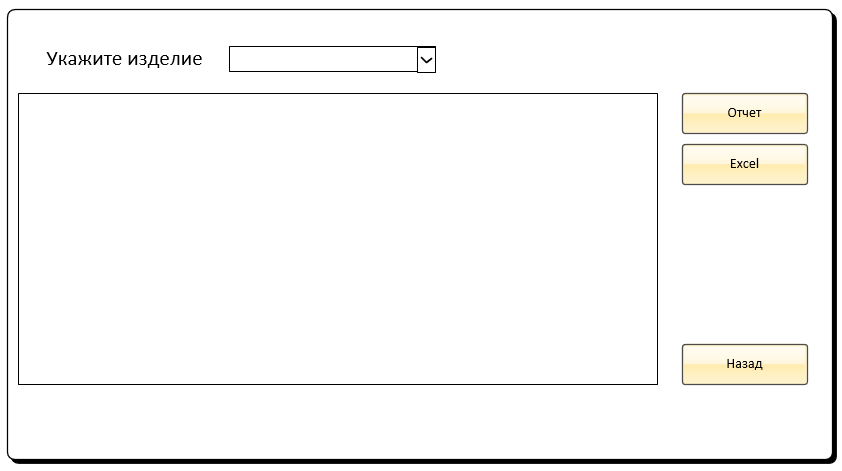


Рис. 15 – макет страницы «Отчетность»

Интерфейс страницы «Отчетность» представлена на рисунке 16.

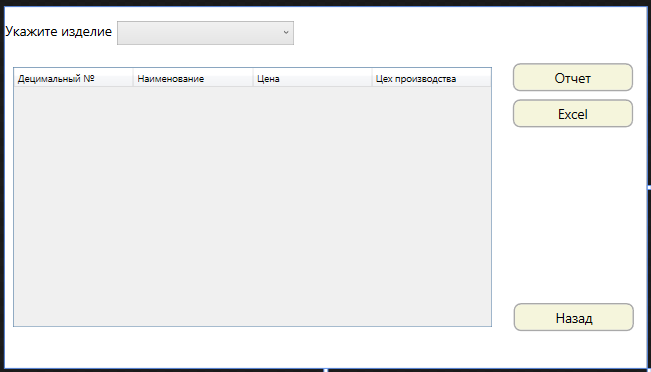


Рис. 16 – интерфейс страницы «Отчетность»

Таблица 22 – Свойства и компоненты страницы «Отчетность»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | - |
| StackPanel | - | Orientation="Horizontal" VerticalAlignment="Center" |
| TextBlock | - | Text="Укажите изделие" FontSize="17" Width="140" |
| ComboBox | cbProduct | FontSize="17" Width="220" DisplayMemberPath="NameProduct" SelectionChanged="cbProduct\_SelectionChanged" |
| DataGrid | dgReport | Grid.Row="1" Margin="10" AutoGenerateColumns="False" |
| StackPanel | - | Grid.Column="1" Grid.Row="1" VerticalAlignment="Top" |
| Button | btnRepWord | Content="Отчет" FontSize="17" Width="150" Height="35" Margin="5" Click="btnRepWord\_Click" |
| Button | btnExcel | Content="Excel" FontSize="17" Width="150" Height="35" Margin="5" Click="btnExcel\_Click" |
| Button | btnBack | Content="Назад" Grid.Column="1" Grid.Row="1" VerticalAlignment="Bottom" FontSize="17" Width="150" Height="35" Margin="5" Click="btnBack\_Click" |

## 2.4. Листинг программы

Логика основного окна представлена в листинге 1.

using PlatinumInform.Pages;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace PlatinumInform.Windows

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new AuthorizationPage());

}

}

}

Листинг 1 – логика основного окна

Логика страницы «Авторизация» представлена в листингах 2-3.

using PlatinumInform.Entities;

using PlatinumInform.Windows;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для AuthorizationPage.xaml

/// </summary>

public partial class AuthorizationPage : Page

{

public AuthorizationPage()

{

Листинг 2 – логика страницы «Авторизация»

InitializeComponent();

}

public static string Encrypt(string value)

{

StringBuilder Sb = new StringBuilder();

using (SHA256 hash = SHA256Managed.Create())

{

Encoding enc = Encoding.UTF8;

Byte[] result = hash.ComputeHash(enc.GetBytes(value));

foreach (Byte b in result)

Sb.Append(b.ToString("x2"));

}

return Sb.ToString();

}

private void btnEnter\_Click(object sender, RoutedEventArgs e)

{

string EncryptPassword = Encrypt(tbPassword.Password);

try

{

Users users = ConectDB.inform.Users.Where(c => c.Login == tbLogin.Text && c.Password == EncryptPassword).FirstOrDefault();

if (users != null)

{

NavigationService.Navigate(new MenuPage());

}

else

{

MessageBox.Show("Введен неправильно логин или пароль!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception)

{

MessageBox.Show("Произошла ошибка при попытке подключения к базе. Повторите попытка входа позже", "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

}

}

Листинг 3 – логика страницы «Авторизация»

Логика страницы «Меню» представлена в листингах 4-5.

using PlatinumInform.Entities;

using PlatinumInform.Windows;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

Листинг 4 – логика страницы «Меню»

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для MenuPage.xaml

/// </summary>

public partial class MenuPage : Page

{

public MenuPage()

{

InitializeComponent();

}

private void btnProduct\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ProductPage());

string Dir = $@"{Directory.GetCurrentDirectory()}\Фото\";

foreach (var item in ConectDB.inform.Product.ToList())

{

if (File.Exists(Dir + item.SerialNum + ".png"))

{

Bitmap bitmap = new Bitmap(Dir + item.SerialNum + ".png");

MemoryStream memoryStream = new MemoryStream();

bitmap.Save(memoryStream, System.Drawing.Imaging.ImageFormat.Png);

Product prod = ConectDB.inform.Product.FirstOrDefault(c=>c.SerialNum == item.SerialNum);

prod.Photo = memoryStream.ToArray();

ConectDB.inform.SaveChanges();

}

}

}

private void btnPrice\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new PricePage());

}

private void btnReport\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ReportPage());

}

private void btnAbout\_Click(object sender, RoutedEventArgs e)

{

AboutAppWindow about = new AboutAppWindow();

about.ShowDialog();

}

}

}

Листинг 5 – логика страницы «Меню»

Логика страницы «Изделие» представлено в листингах 6-8.

using PlatinumInform.Entities;

Листинг 6 – логика страницы «Изделие»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для ProductPage.xaml

/// </summary>

public partial class ProductPage : Page

{

Product prod;

Departament departament;

List<Product> SavedValues;

public ProductPage()

{

InitializeComponent();

lwProduct.ItemsSource = ConectDB.inform.Product.ToList();

cbDep.ItemsSource = ConectDB.inform.Departament.ToList();

}

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new AddEditProd());

}

private void btnEdit\_Click(object sender, RoutedEventArgs e)

{

prod = lwProduct.SelectedItem as Product;

if (prod != null)

{

NavigationService.Navigate(new AddEditProd(prod));

}

else

{

MessageBox.Show("Выберите изделие для редактирования", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

private void btnDel\_Click(object sender, RoutedEventArgs e)

{

prod = lwProduct.SelectedItem as Product;

if (prod != null)

{

if (MessageBox.Show("Вы уверены, что хотите удалить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ConectDB.inform.Product.Remove(prod);

ConectDB.inform.SaveChanges();

Листинг 7 – логика страницы «Изделие»

lwProduct.ItemsSource = ConectDB.inform.Product.ToList();

}

}

else

{

MessageBox.Show("Выберите запись для удаления!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

NavigationService.GoBack();

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

lwProduct.ItemsSource = ConectDB.inform.Product.ToList();

SavedValues = ConectDB.inform.Product.ToList();

}

public static List<Product> FilterData(List<Product> data, string nameProduct = null, int numDep = -1)

{

List<Product> FilterValues = data;

if (nameProduct != null)

{

FilterValues = FilterValues.Where(c => c.NameProduct.Contains(nameProduct)).ToList();

}

if (numDep != -1)

{

FilterValues = FilterValues.Where(c => c.NumDep == numDep).ToList();

}

if (FilterValues.Count() == 0)

{

MessageBox.Show("Запись не найдена!");

return data;

}

return FilterValues;

}

private void tbSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

departament = (Departament)cbDep.SelectedItem;

int numDep = departament == null ? -1 : departament.NumDep;

lwProduct.ItemsSource = FilterData(SavedValues, tbSearch.Text, numDep);

}

private void cbDep\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

departament = (Departament)cbDep.SelectedItem;

int numDep = departament == null ? -1 : departament.NumDep;

lwProduct.ItemsSource = FilterData(SavedValues, tbSearch.Text, numDep);

}

private void btnReset\_Click(object sender, RoutedEventArgs e)

{

tbSearch.Text = "";

cbDep.SelectedIndex = -1;

}

}

}

Листинг 8 – логика страницы «Изделие»

Логика страницы «Добавление/изменение» представлена в листингах 9-11.

using Microsoft.Win32;

using PlatinumInform.Entities;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для AddEditProd.xaml

/// </summary>

public partial class AddEditProd : Page

{

Product \_product;

bool Change;

private byte[] mainImage;

public AddEditProd()

{

InitializeComponent();

Change = false;

tbOrganization.IsEnabled = false;

tbCharacteristic.IsEnabled = false;

tbNumPP.IsEnabled = false;

cbDepartament.ItemsSource = ConectDB.inform.Departament.ToList();

int newSerialNum = ConectDB.inform.Product.Max(c => c.SerialNum) + 1;

tbOrganization.Text = (1001).ToString();

tbCharacteristic.Text = (40974).ToString();

tbNumPP.Text = (newSerialNum).ToString();

tbDetail.Text = null;

}

public AddEditProd(Product product)

{

InitializeComponent();

Change = true;

this.DataContext = product;

imgPhoto.DataContext = product;

\_product = product;

tbOrganization.IsEnabled = false;

tbCharacteristic.IsEnabled = true;

tbNumPP.IsEnabled = true;

tbOrganization.Text = (product.CodeOrg).ToString();

tbCharacteristic.Text = (product.CodeCharac).ToString();

tbNumPP.Text = (product.SerialNum).ToString();

tbDetail.Text = product.NameProduct;

cbDepartament.ItemsSource = ConectDB.inform.Departament.ToList();

cbDepartament.SelectedItem = product.Departament;

Листинг 9 – логика страница «Добавление/изменение»

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (!Change)

{

try

{

Product newProduct = new Product();

int newCodeDetail = ConectDB.inform.Product.Max(c => c.CodeProduct) + 1;

newProduct.CodeProduct = newCodeDetail;

newProduct.CodeOrg = Convert.ToInt32(tbOrganization.Text);

newProduct.CodeCharac = Convert.ToInt32(tbCharacteristic.Text);

newProduct.SerialNum = Convert.ToInt32(tbNumPP.Text);

newProduct.NameProduct = tbDetail.Text;

newProduct.NumDep = (cbDepartament.SelectedItem as Departament).NumDep;

newProduct.Photo = mainImage;

if (MessageBox.Show("Вы уверены, что хотите добавить новую запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ConectDB.inform.Product.Add(newProduct);

ConectDB.inform.SaveChanges();

MessageBox.Show("Данные успешно сохранены!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.GoBack();

}

}

catch (Exception)

{

MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

else

{

try

{

Product product = ConectDB.inform.Product.FirstOrDefault(c => c.CodeProduct == \_product.CodeProduct);

product.CodeOrg = Convert.ToInt32(tbOrganization.Text);

product.CodeCharac = Convert.ToInt32(tbCharacteristic.Text);

product.SerialNum = Convert.ToInt32(tbNumPP.Text);

product.NameProduct = tbDetail.Text;

product.NumDep = (cbDepartament.SelectedItem as Departament).NumDep;

if (mainImage != null)

{

product.Photo = mainImage;

}

if (MessageBox.Show("Вы уверены, что хотите изменить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ConectDB.inform.SaveChanges();

MessageBox.Show("Данные успешно изменены!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.GoBack();

}

}

Листинг 10 – логика страницы «Добавление/изменение»

catch (Exception)

{

MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

}

private void btnCancel\_Click(object sender, RoutedEventArgs e)

{

NavigationService.GoBack();

}

private void tbDetail\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = (Char.IsDigit(e.Text, 0));

}

private void tbOrganization\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = !(Char.IsDigit(e.Text, 0));

}

private void btnLoad\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "Image |\*.png; \*.jpg; \*.jpeg";

if (ofd.ShowDialog() == true)

{

mainImage = File.ReadAllBytes(ofd.FileName);

imgPhoto.Source = (ImageSource)new ImageSourceConverter().ConvertFrom(mainImage);

}

}

}

}

Листинг 11 – логика страницы «Добавление/изменение»

Логика страницы «Цена» представлена в листингах 12-14.

using PlatinumInform.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для PricePage.xaml

Листинг 12 – логика страницы «Цена»

/// </summary>

public partial class PricePage : Page

{

Product detail;

List<Price> SavedValues;

public PricePage()

{

InitializeComponent();

dgPrice.ItemsSource = ConectDB.inform.Price.ToList();

cbProduct.ItemsSource = ConectDB.inform.Product.ToList();

}

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new AddPrice());

}

private void btnDel\_Click(object sender, RoutedEventArgs e)

{

Price price = dgPrice.SelectedItem as Price;

if (price != null)

{

if (MessageBox.Show("Вы уверены, что хотите удалить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ConectDB.inform.Price.Remove(price);

ConectDB.inform.SaveChanges();

}

}

else

{

MessageBox.Show("Выберите запись для удаления!","Ошибка",MessageBoxButton.OK,MessageBoxImage.Error);

}

dgPrice.ItemsSource = ConectDB.inform.Price.ToList();

}

private void btnReset\_Click(object sender, RoutedEventArgs e)

{

cbProduct.SelectedIndex = -1;

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

NavigationService.GoBack();

}

public static List<Price> FilterData(List<Price> data, int numProd = -1)

{

List<Price> FilterValues = data;

if (numProd != -1)

{

FilterValues = FilterValues.Where(c => c.CodeProduct == numProd).ToList();

}

if (FilterValues.Count() == 0)

{

MessageBox.Show("Запись не найдена!");

return data;

}

return FilterValues;

Листинг 13 – логика страницы «Цена»

}

private void cbProduct\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

detail = (Product)cbProduct.SelectedItem;

int numProd = detail == null ? -1 : detail.CodeProduct;

dgPrice.ItemsSource = FilterData(SavedValues, numProd);

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

dgPrice.ItemsSource = ConectDB.inform.Price.ToList();

SavedValues = ConectDB.inform.Price.ToList();

}

}

}

Листинг 14 – логика страницы «Цена»

Логика страницы «Добавление данных» представлен в листингах 15-16.

Using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для AddPrice.xaml

/// </summary>

public partial class AddPrice : Page

{

public AddPrice()

{

InitializeComponent();

DateTime DateNow = DateTime.Today;

cbDetail.ItemsSource = ConectDB.inform.Product.ToList();

dpDateInsert.SelectedDate = DateNow;

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

try

{

Price price = new Price();

int newId = ConectDB.inform.Price.Max(c => c.NumPP) + 1;

price.NumPP = newId;

price.DateInsert = dpDateInsert.SelectedDate.Value;

price.CodeProduct = (cbDetail.SelectedItem as Product).CodeProduct;

price.Cost = Convert.ToDecimal(tbCost.Text);

price.NumContract = Convert.ToInt32(tbNumContract.Text);

if (MessageBox.Show(“Вы уверены, что хотите добавить новую запись?”, “Внимание”, MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

Листинг 15 – логика страницы «Добавление данных»

{

ConectDB.inform.Price.Add(price);

ConectDB.inform.SaveChanges();

MessageBox.Show(“Данные успешно сохранены!”, “Внимание”, MessageBoxButton.OK, MessageBoxImage.Information);

}

NavigationService.GoBack();

}

catch (Exception)

{

MessageBox.Show(“Введены некорректные данные или остались незаполненные поля!”, “Внимание”, MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void btnCancel\_Click(object sender, RoutedEventArgs e)

{

NavigationService.GoBack();

}

private void tbCost\_TextChanged(object sender, TextChangedEventArgs e)

{

if (tbCost.Text != “”)

{

int Cost = Convert.ToInt32(tbCost.Text);

if (Cost <= 0)

{

MessageBox.Show(“Цена должна быть больше 0”, “Внимание”, MessageBoxButton.OK, MessageBoxImage.Warning);

tbCost.Text = null;

}

}

}

private void tbCost\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = !(Char.IsDigit(e.Text, 0));

}

}

}

Листинг 16 – логика страницы «Добавление данных»

Логика страницы «Отчетность» представлена в листингах 17-20.

using PlatinumInform.Entities;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using Word = Microsoft.Office.Interop.Word;

Листинг 17 – логика страницы «Отчетность»

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Word;

using Page = System.Windows.Controls.Page;

using Microsoft.Win32;

using System.Windows.Shapes;

namespace PlatinumInform.Pages

{

/// <summary>

/// Логика взаимодействия для ReportPage.xaml

/// </summary>

public partial class ReportPage : Page

{

Price report;

public ReportPage()

{

InitializeComponent();

cbProduct.ItemsSource = ConectDB.inform.Product.ToList();

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

NavigationService.GoBack();

}

private void btnRepWord\_Click(object sender, RoutedEventArgs e)

{

DateTime date = DateTime.Now;

report = dgReport.SelectedItem as Price;

if (report != null)

{

try

{

string source = $@"{Directory.GetCurrentDirectory()}\Информация по детали.docx";

Word.Application app = new Word.Application();

Word.Document doc = app.Documents.Open(source);

doc.Activate();

try

{

var res = ConectDB.inform.Price.Join(ConectDB.inform.Product, price => price.CodeProduct, detail => detail.CodeProduct, (price, detail) => new

{

price.Cost,

price.DateInsert,

detail.CodeOrg,

detail.CodeCharac,

detail.SerialNum,

detail.CodeProduct,

detail.NameProduct,

detail.NumDep

}).Join(ConectDB.inform.Departament, price => price.NumDep, departament => departament.NumDep, (price, departament) => new

{

price.Cost,

price.DateInsert,

price.CodeOrg,

price.CodeCharac,

price.SerialNum,

price.CodeProduct,

price.NumDep,

Листинг 18 – логика страницы «Отчетность»

price.NameProduct,

departament.Name

}).Where(c => c.CodeProduct == report.CodeProduct).FirstOrDefault();

Word.Bookmarks wB = doc.Bookmarks;

wB["ДетальГлавная"].Range.Text = res.NameProduct.ToString();

wB["ДецНомер"].Range.Text = res.CodeOrg.ToString() + "." + res.CodeCharac.ToString() + "." + res.SerialNum.ToString();

wB["Деталь"].Range.Text = res.NameProduct.ToString();

wB["Цех"].Range.Text = res.Name.ToString();

wB["Цена"].Range.Text = res.Cost.ToString();

wB["ДатаУстановки"].Range.Text = res.DateInsert.ToShortDateString();

wB["ДатаДоговора"].Range.Text = date.ToShortDateString();

doc.SaveAs2($@"{Directory.GetCurrentDirectory()}\Информация по деталям\Информация по детали {res.NameProduct}.docx");

doc.Close();

doc = null;

app.Quit();

MessageBox.Show("Файл успешно создан!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

doc.Close();

doc = null;

app.Quit();

}

}

catch (Exception)

{

MessageBox.Show("Отсутствует макет документа!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

else

{

MessageBox.Show("Выберите данные для создания файла!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void btnExcel\_Click(object sender, RoutedEventArgs e)

{

int count = 0;

var app = new Excel.Application();

try

{

SaveFileDialog svd = new SaveFileDialog();

svd.Title = "Сохранить как...";

svd.Filter = "Excel file(\*.xlsx)|.\*xslx";

svd.OverwritePrompt = true;

svd.CheckPathExists = true;

if (svd.ShowDialog() == false)

{

app.Quit();

}

else

{

Листинг 19 – логика страницы «Отчетность»

List<Price> specializationList = ConectDB.inform.Price.ToList();

app.Workbooks.Add(Type.Missing);

var worksheet = (Excel.Worksheet)app.Worksheets[1];

count = specializationList.Count;

for (int i = 0; i < count; i++)

{

worksheet.Cells[i + 1, 1] = specializationList[i].Product.NameProduct;

worksheet.Cells[i + 1, 2] = specializationList[i].Cost;

}

worksheet.Columns.AutoFit();

var xlCharts = (Excel.ChartObjects)worksheet.ChartObjects(Type.Missing);

var myChart = (Excel.ChartObject)xlCharts.Add(120, 0, 350, 250);

var chart = myChart.Chart;

var seriesCollection = (Excel.SeriesCollection)chart.SeriesCollection(Type.Missing);

var series = seriesCollection.NewSeries();

series.XValues = worksheet.get\_Range("A1", "A" + count);

series.Values = worksheet.get\_Range("B1", "B" + count);

chart.ChartType = Excel.XlChartType.xlColumnStacked;

app.Application.ActiveWorkbook.SaveAs(svd.FileName);

app.Quit();

MessageBox.Show("Отчет сформирован!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

app.Quit();

}

}

private void cbProduct\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Product detail = (Product)cbProduct.SelectedItem;

if (detail != null)

{

dgReport.ItemsSource = ConectDB.inform.Price.ToList().Where(c => c.CodeProduct == detail.CodeProduct);

}

}

}

}

Листинг 20 – логика страницы «Отчетность»

## 2.5.Разработка тестовых данных

### 2.5.1. Разработка тестов.

Информация по тесту №1 представлена на рисунке 17.

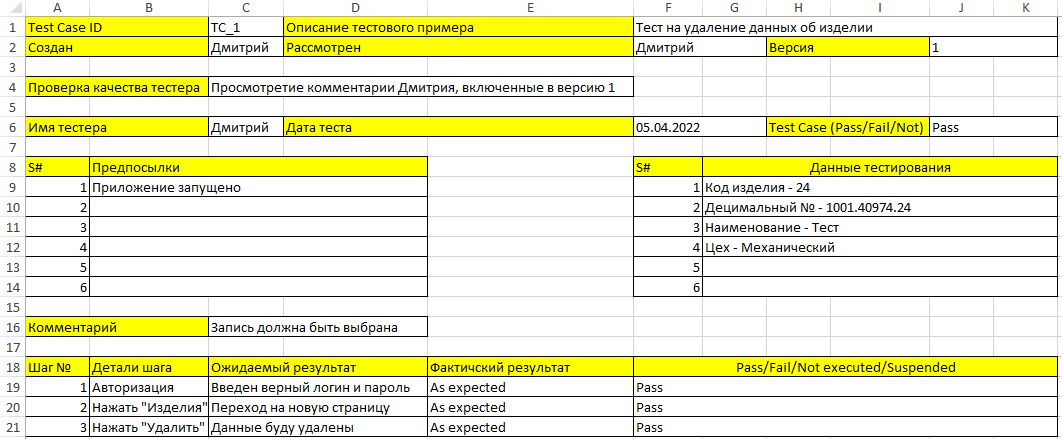


Рис. 17 – Тест №1

Информацию по тесту №2 представлена на рисунке 18.

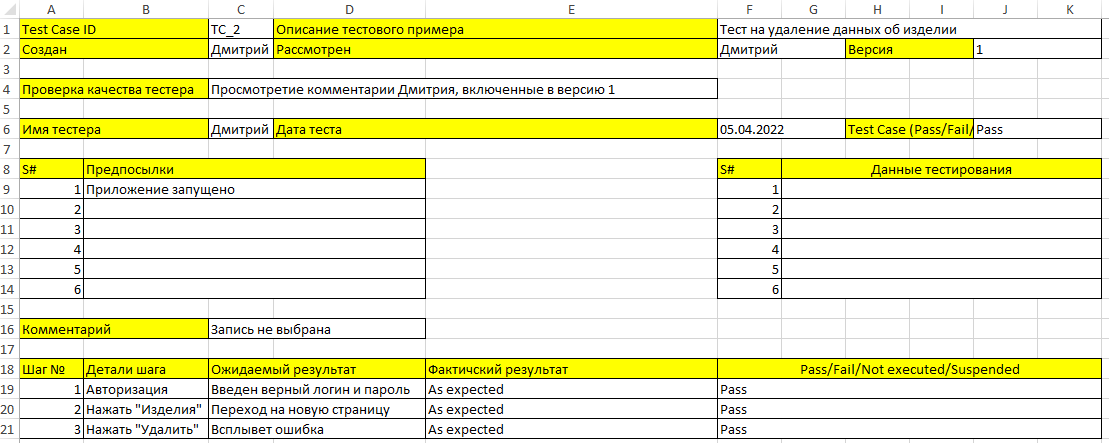


Рис. 18 – Тест №2

Информация по тесту №3 представлена на рисунке 19.

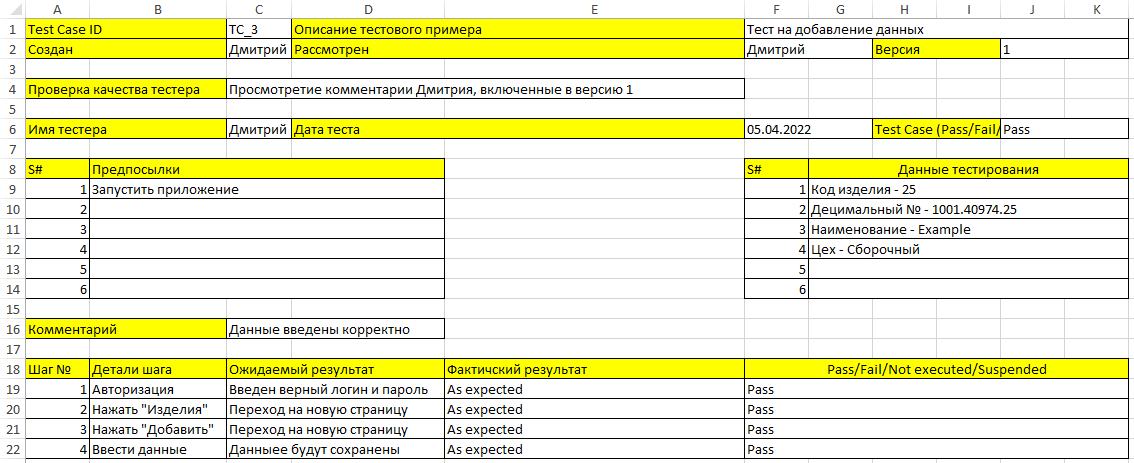


Рис. 19 – Тест №3

Информация по тесту №4 представлена на рисунке 20.

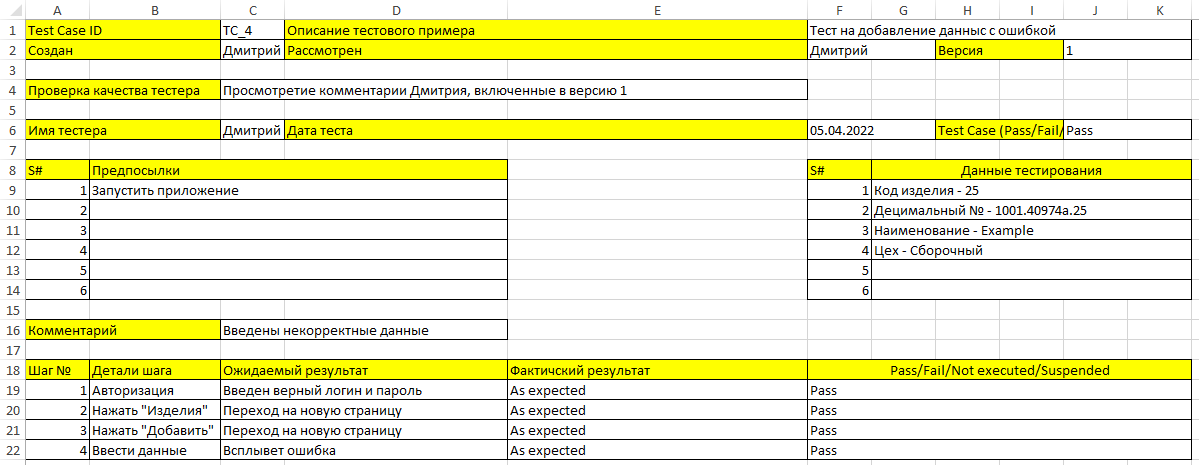


Рис. 20 – Тест №4

Информация по тесту №5 представлена на рисунке 21.

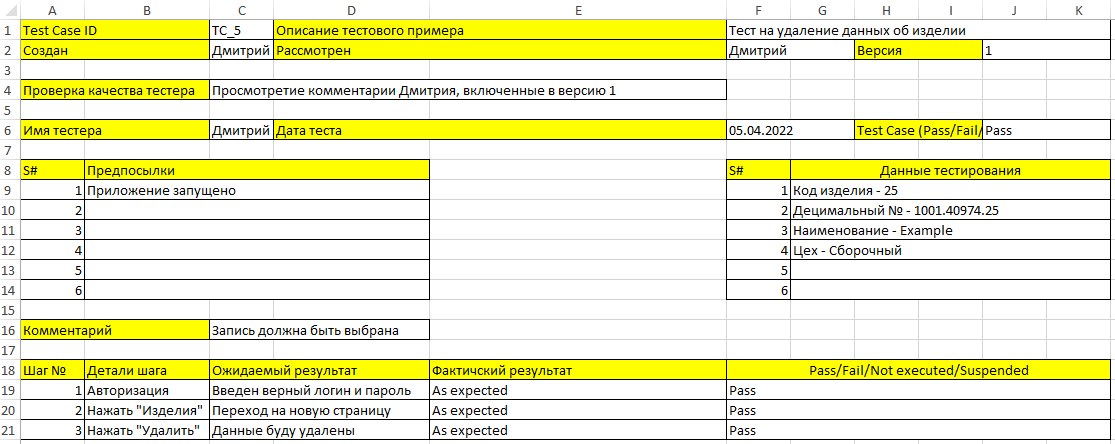


Рис. 21 – Тест №5

Информация по тесту №6 представлена на рисунке 22.

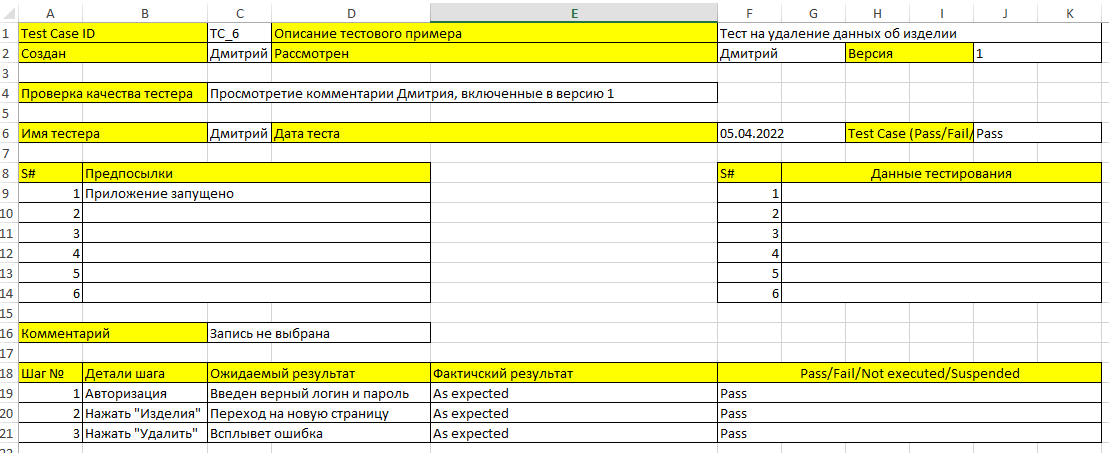


Рис. 22 – Тест №6

Информация по тесту №7 представлена на рисунке 23.

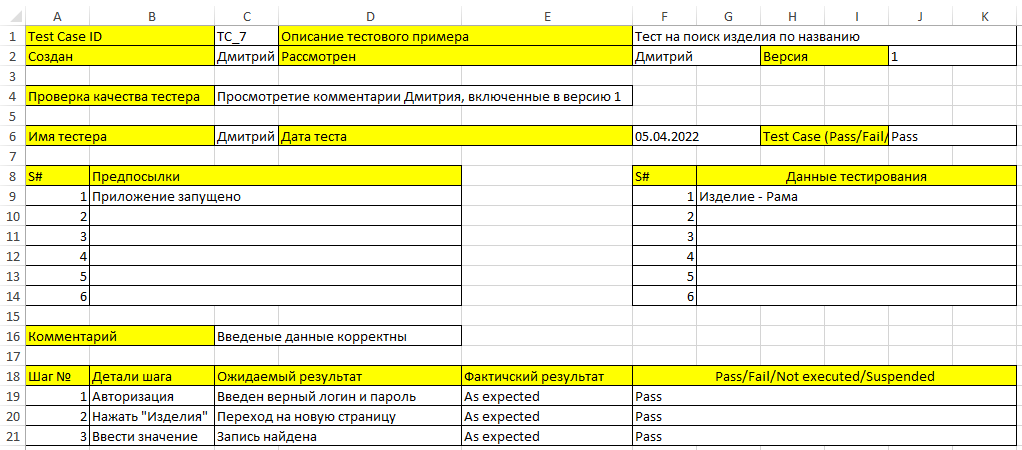


Рис. 23 – Тест №7

Информация по тесту №8 представлена на рисунке 24.

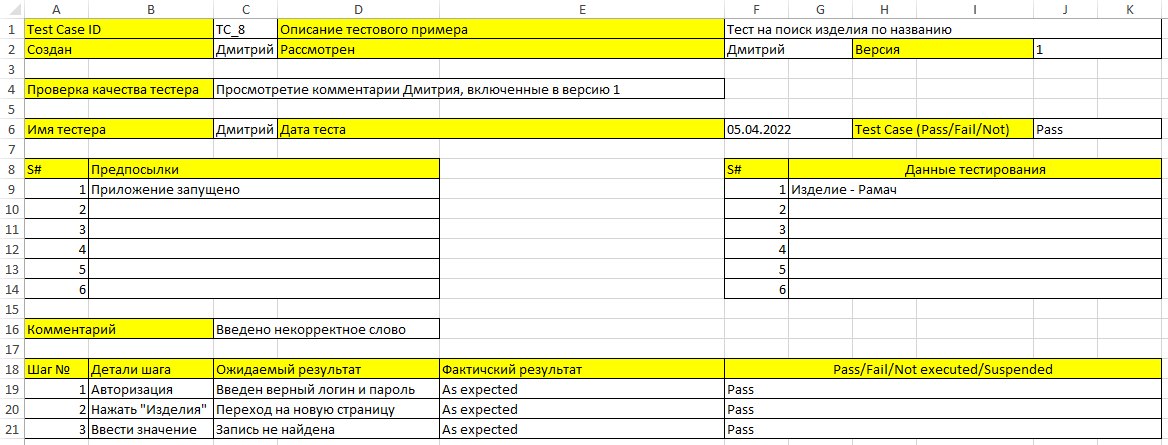


Рис. 24 – Тест №8

Информация по тесту №9 представлена на рисунке 25.

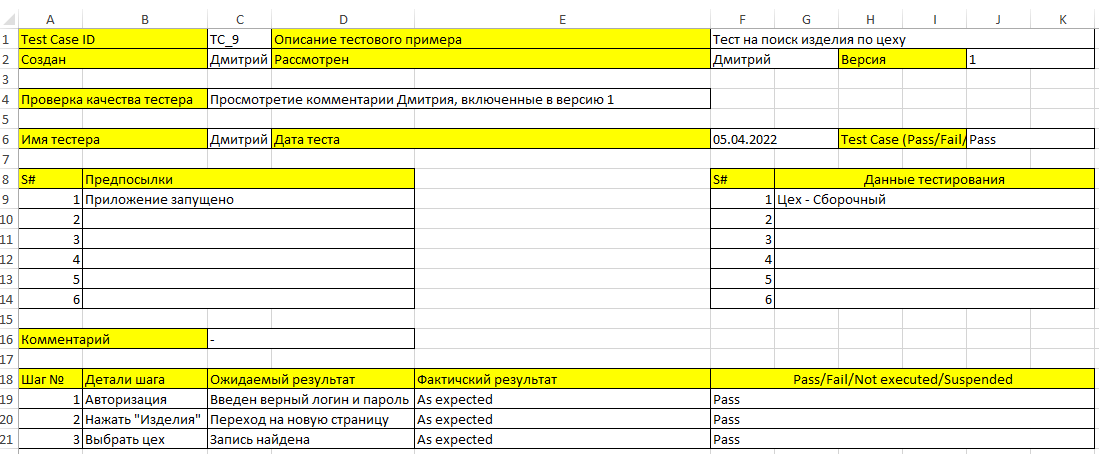


Рис. 25 – Тест №9

Информация по тесту №10 представлена на рисунке 26.

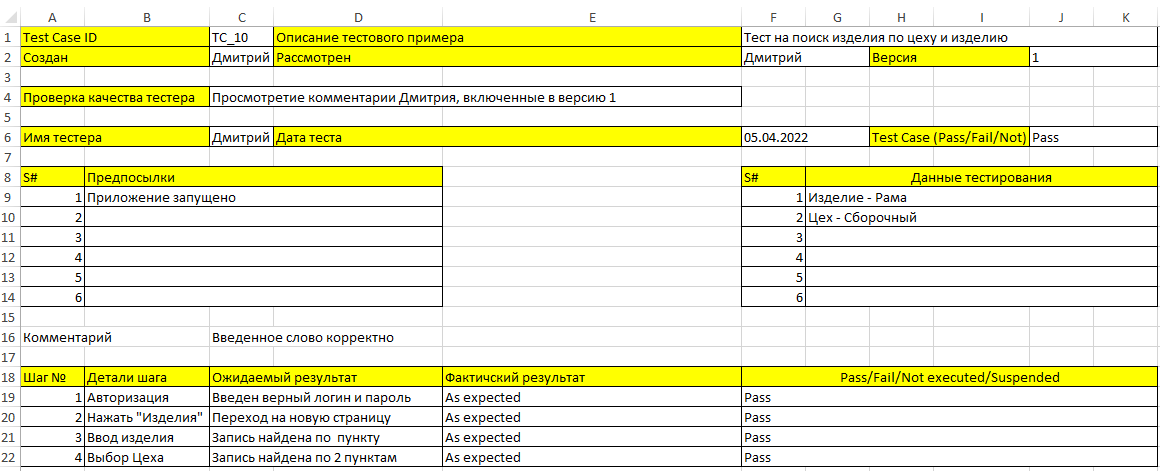


Рис. 26 – Тест №10

Информация по тесту №11 представлена на рисунке 27.

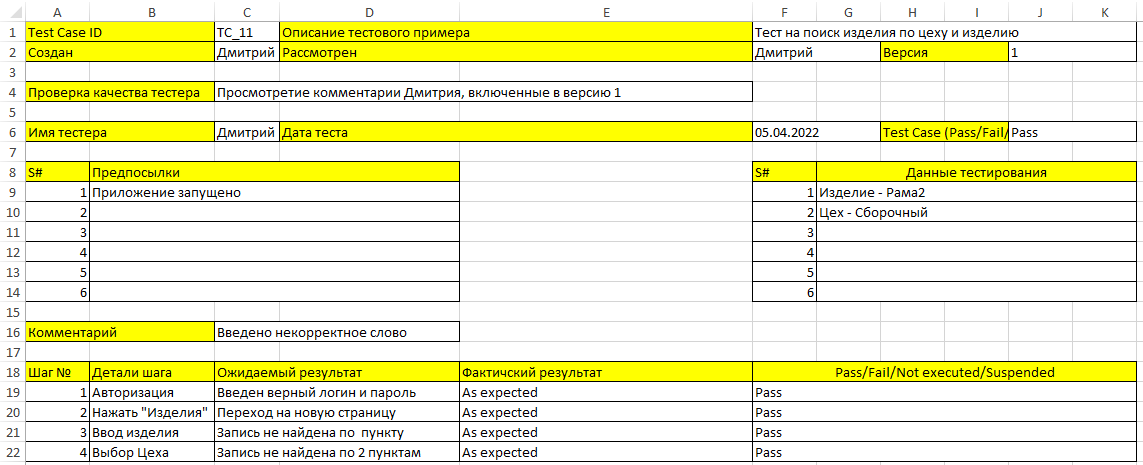


Рис. 27 – Тест №11

Информация по тесту №12 представлена на рисунке 28.

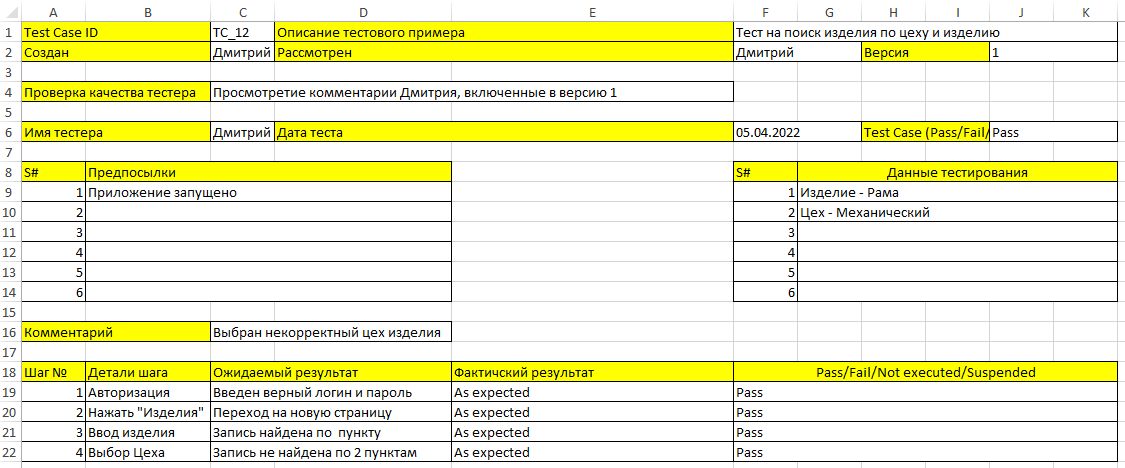


Рис. 28 – Тест №12

Информация по тесту №13 представлена на рисунке 29.

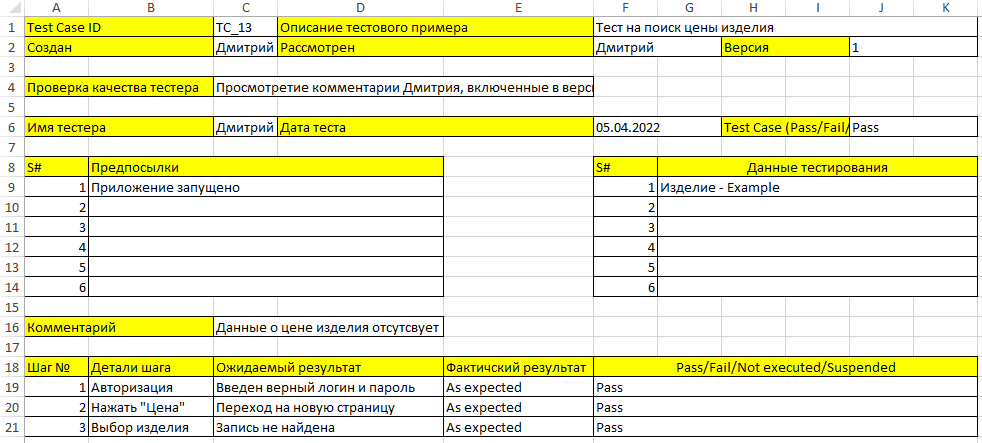


Рис. 29 –Тест №13

Информация по тесту №14 представлена на рисунке 30.

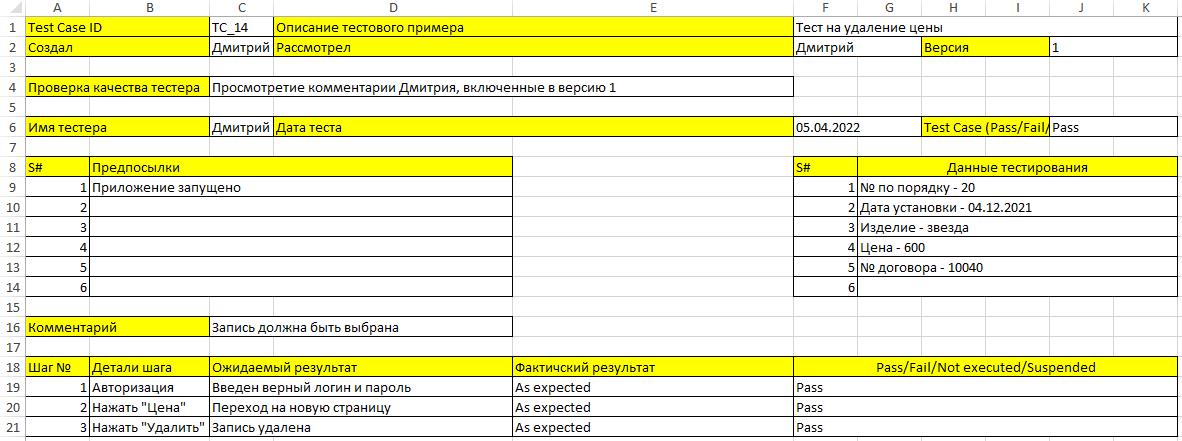


Рис. 30 – Тест №14

Информация по тесту №15 представлена на рисунке 31.

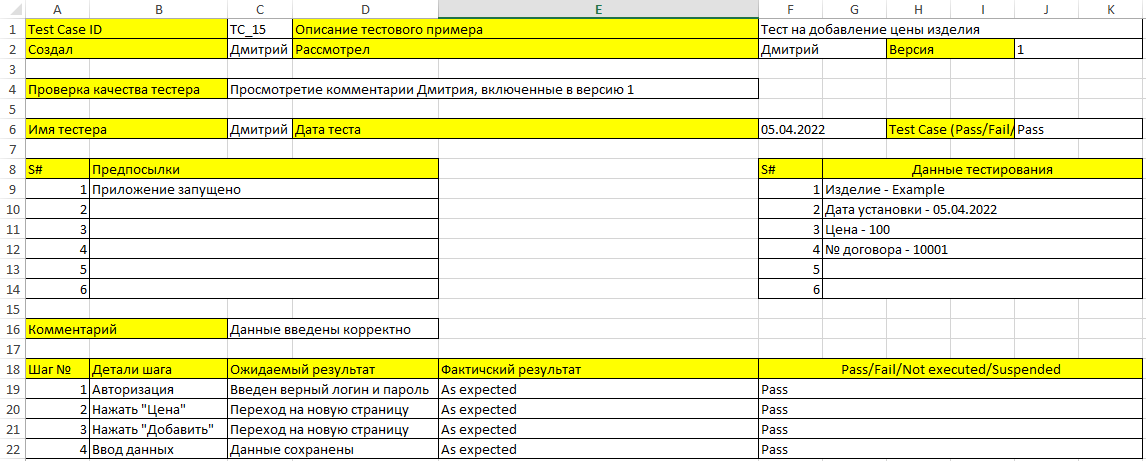


Рис. 31 – Тест №15

Информация по тесту №16 представлена на рисунке 32.

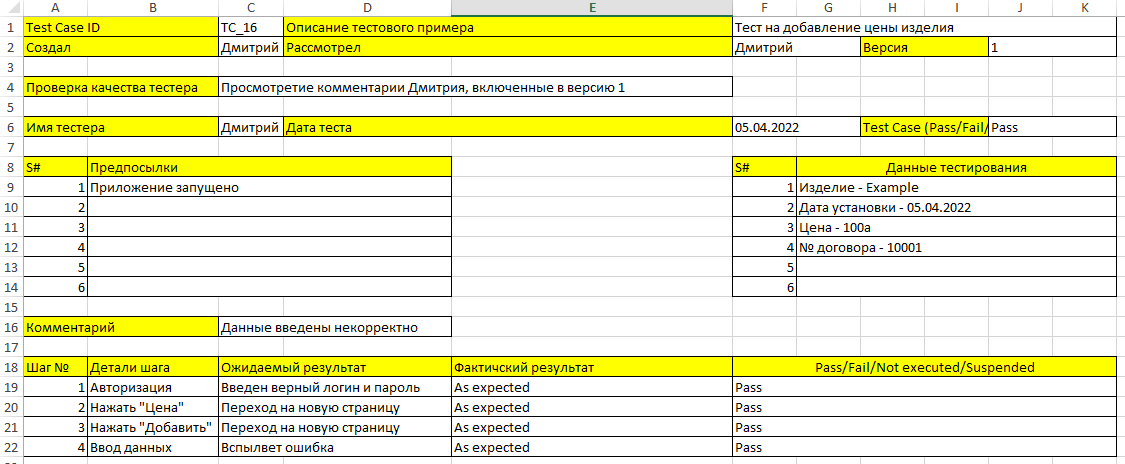


Рис. 32 – Тест №16

Информация по тесту №17 представлена на рисунке 33.

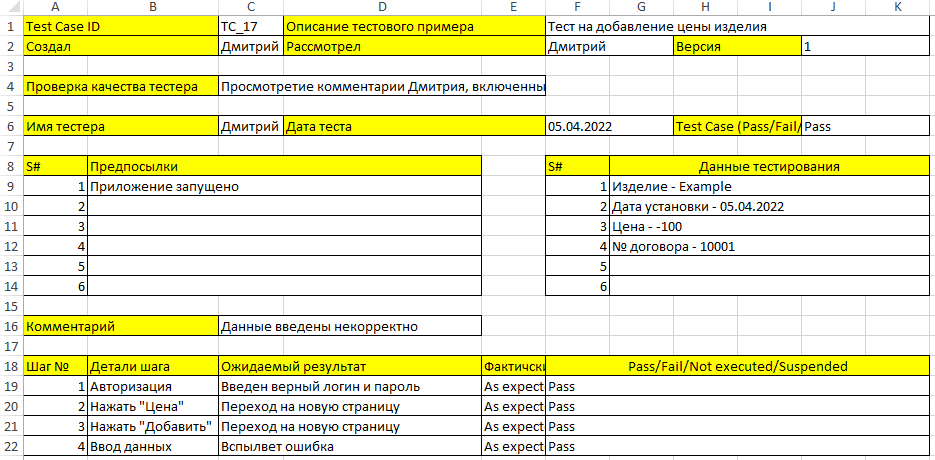


Рис. 33 – тест №17

Информация по тесту №18 представлена на рисунке 34.

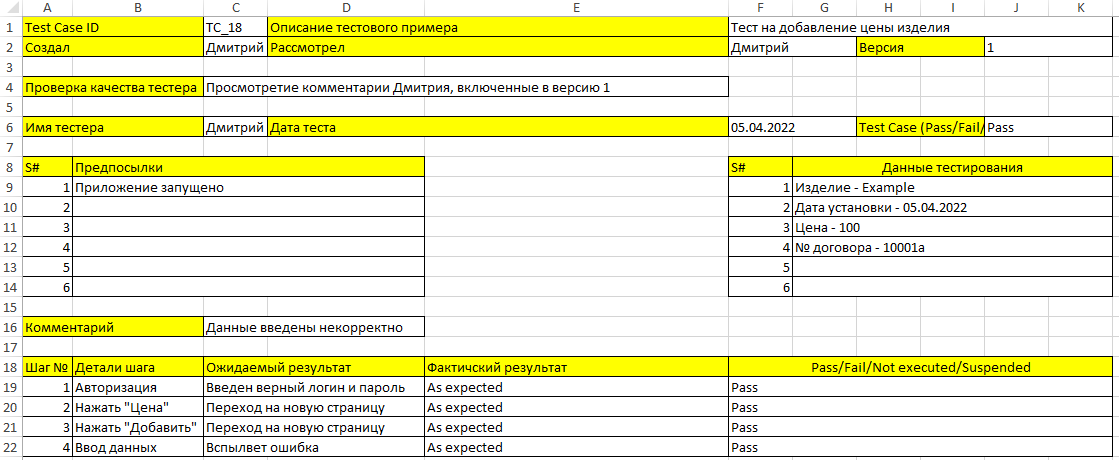


Рис. 34 – Тест №18

Информация по тесту №19 представлена на рисунке 35.

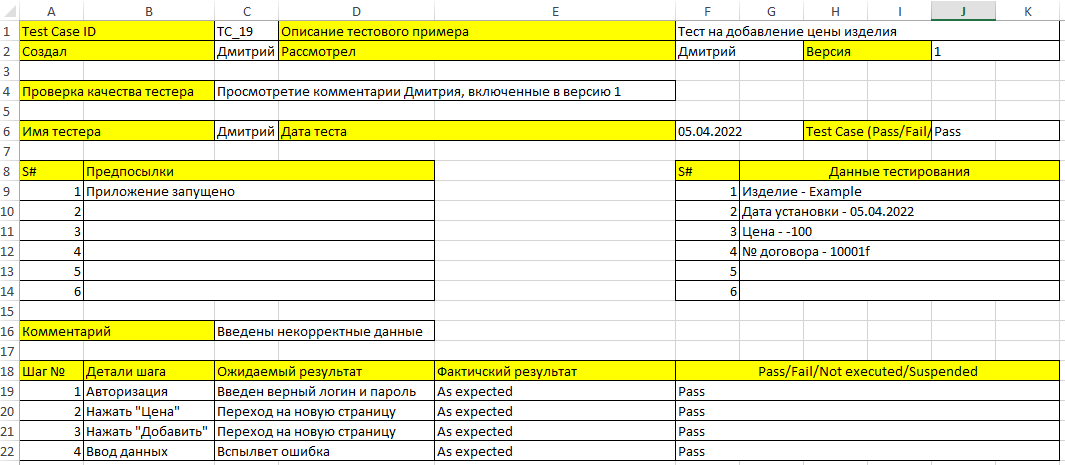


Рис. 35 – Тест №19

Информация по тесту №20 представлена на рисунке 36.

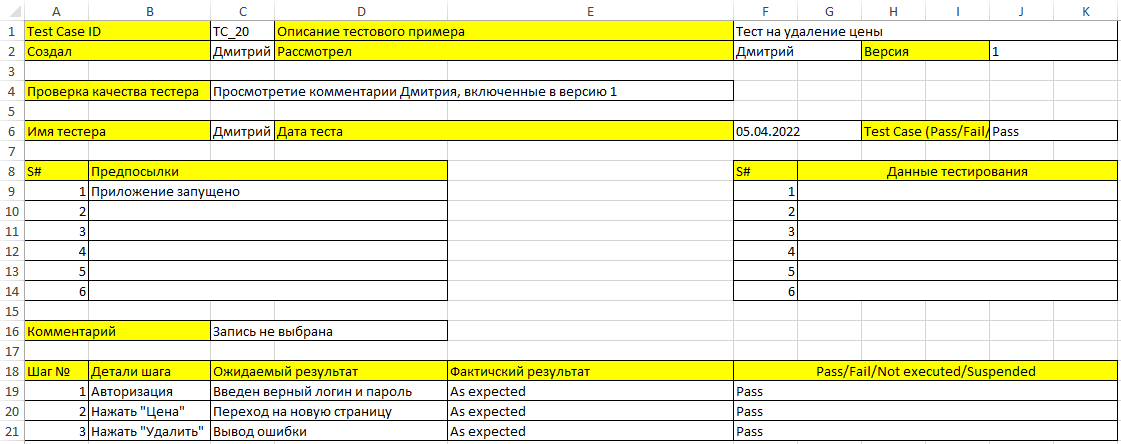


Рис. 36 – Тест №20

Результат работы тестов представлен на рисунке 37.

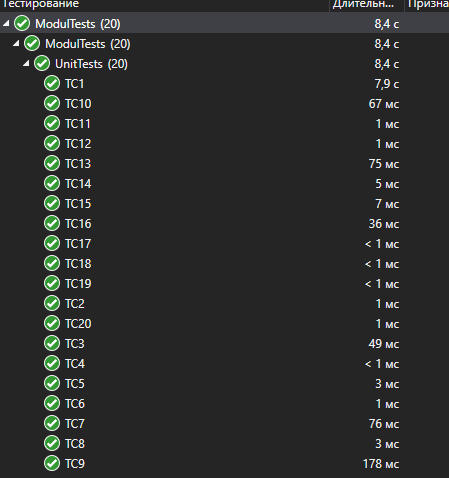


Рис. 37 – результат тестов

## 2.6. Результат работы программы

Результат работы программы представлен на рисунках 38-44

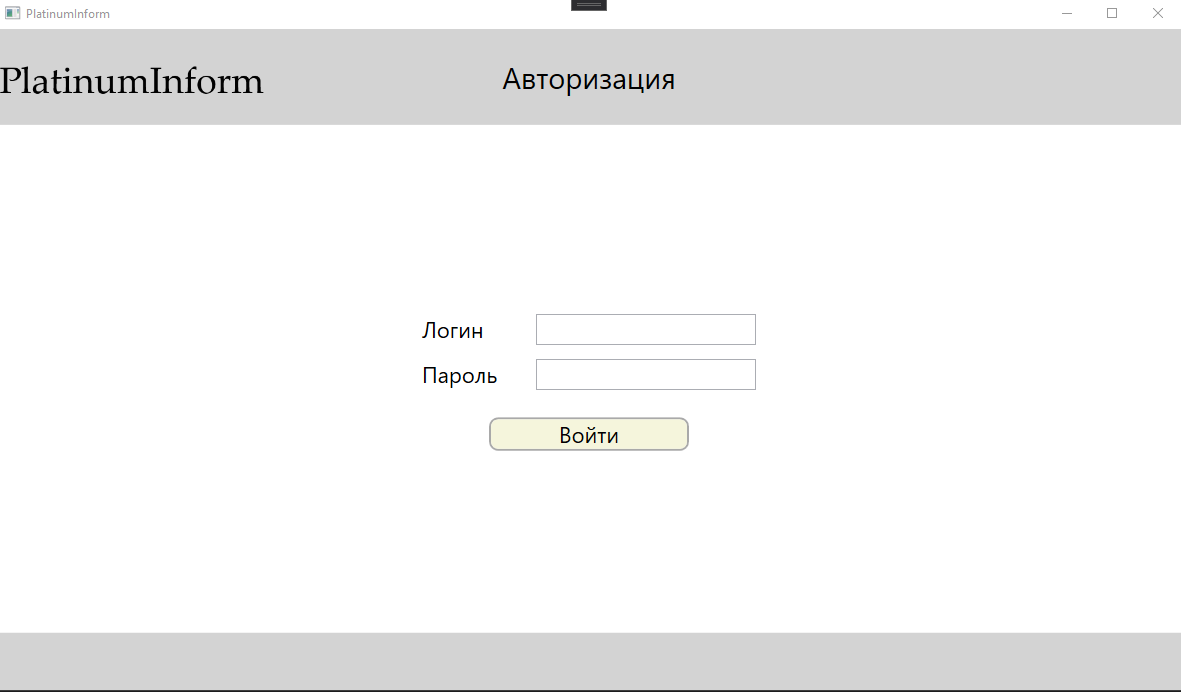


Рис. 38 – результат работы программы «Авторизация»

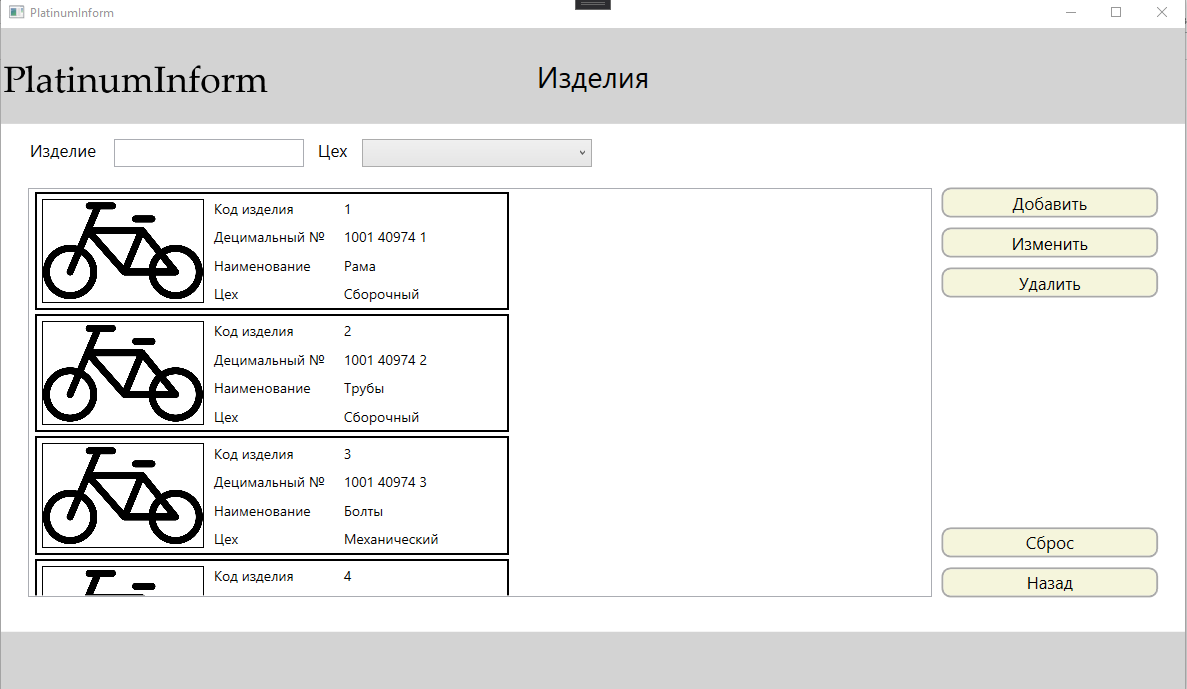


Рис. 39 – результат работы программы «Изделия»

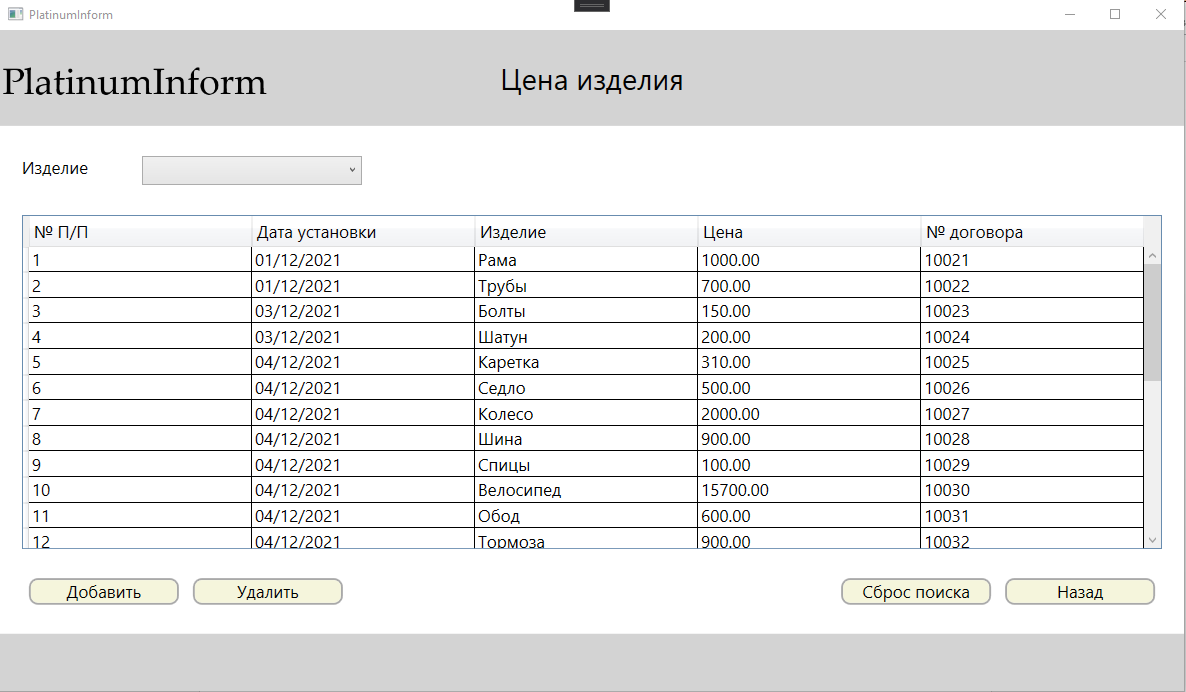


Рис. 40 – результат работы программы «Цена изделия»

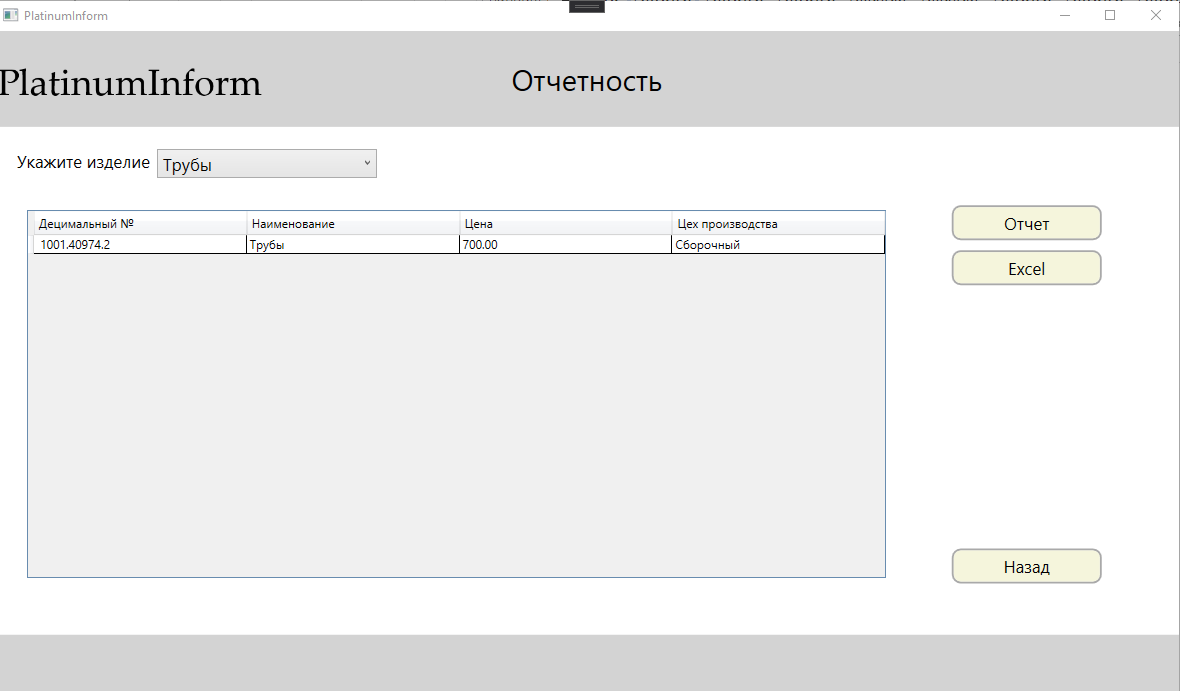


Рис. 41 – результат работы программы «Отчетность»

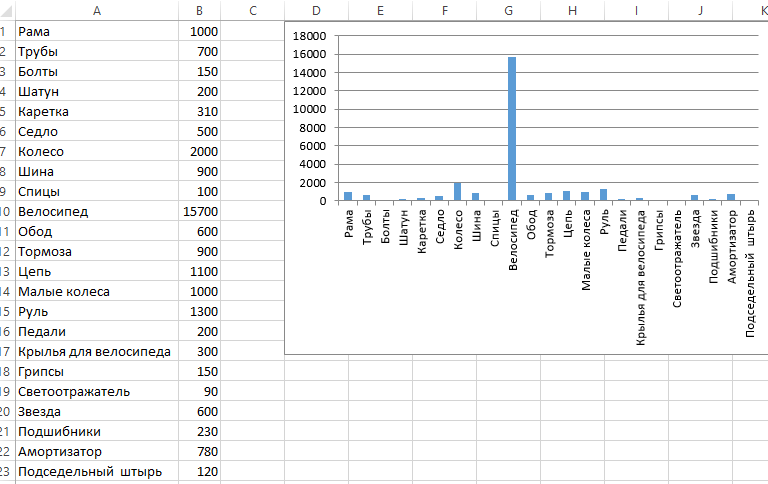


Рис. 42 – результат работы программы «График Excel»



Рис. 43 – результат работы программы «Отчет»

## 2.7. Руководство по использования программы

### 2.7.1. Руководство системного программиста

Данная программа предназначена для ведения нормативно-справочной информации по изделиям, которое выпускает завод «РАТЕП».

Для корректной работы данной программы требуются следующее аппаратное и программное обеспечение:

* процессор AMD A8-5500with Radeon(tm) HD Graphics (или выше) иди совместимый с ним;
* оперативная память: 4 Гб;
* видеоадаптер AMD Radeon HD;
* операционная система Windows 7/8/10
* .NET Framework 4.7.1.

Данная программа состоит из следующих файлов:

* Исполняемого файла TranspotCall.exe;

Приложение поставляется путем установочного файла, содержащий все необходимое для работы программы.

Сообщение системному программисту представлены в таблице 23.

|  |  |
| --- | --- |
| Сообщение | Описание действий при получении сообщений |
| Произошла ошибка подключения к базе данных | Проверьте введенные данные в строке «Имя сервера» |

### 2.6.2. Руководство программиста

Данная программа предназначена для ведения нормативно-справочной информации по изделиям, которое выпускает завод «РАТЕП».

Программа предусматривает авторизацию для одной роли «Инженер-технолог»

Для корректной работы данной программы требуется следующее аппаратное и программное обеспечение:

* процессор AMD A8-5500with Radeon(tm) HD Graphics (или выше) иди совместимый с ним;
* оперативная память: 4 Гб;
* видеоадаптер AMD Radeon HD;
* операционная система Windows 7/8/10;
* Microsoft Word 2013 и выше;
* Microsoft Excel 2013 b выше.

Настройка программы.

Для начала установки программы вам необходимо установить Visual Studio 2019. Скачать установочные файлы можно на официальном сайте Microsoft: https://visualstudio.microsoft.com/ru/downloads/. Далее нужно запустить установочный файл Setup. Следуйте инструкции мастера установки. Выберите нужный путь расположения файлов на вашем компьютере, либо установите в уже заданную директорию по умолчанию.

Важно: не забудьте куда вы установили программу!

После успешной установки, в заданной директории при установке, вы найдете файлы приложения и саму программу. Файлы приложения можно оставить там же или перенести их в стандартную директорию файлов Visual Studio 2019.

Внимание: обычно все файлы Visual Studio 2019 находятся в директории: C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\

Запускаем VisualStudio 2019. Для начала необходимо найти программу «RepairTail». Для этого при открытии Visual Studio 2019 написать в поиске «RepairTail» или найти при нажатии «Открыть проект или решение» программу, которую вы расположили.

Сообщения программисту представлены в таблице 24.

|  |  |
| --- | --- |
| Сообщение | Описание действий при получении сообщений |
| Попытка обращения к элементу массива или коллекции с индексом,  который находится вне границ | Убедитесь, что указанный индекс существует в коллекции. |
| Ошибках операций  преобразования | Убедитесь, что приведение объекта к данному типу возможно. |

### 2.5.3. Руководство пользователя

Запускаем приложение. В открывшемся окне необходимо авторизоваться. Интерфейс листа «Авторизация» представлен на рисунке 44.

Внимание: Логин и пароль для авторизации необходимо получить у системного администратора

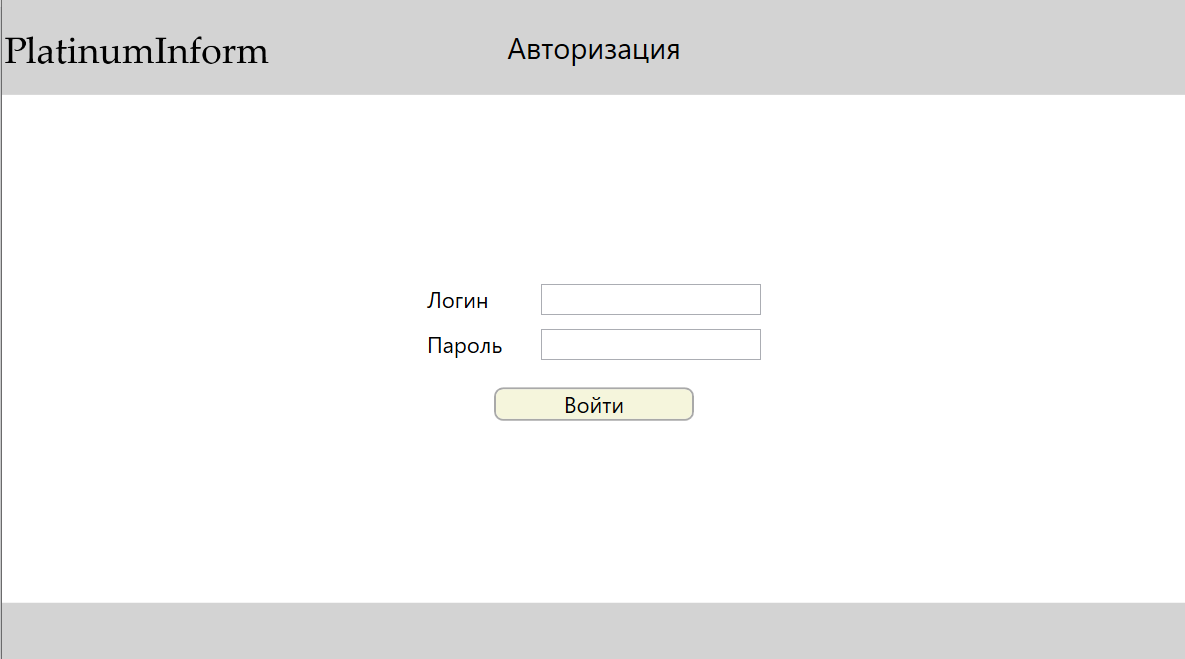


Рис. 44 – страница «Авторизация»

После успешной авторизации Вам откроет главное меню приложения. Интерфейс страницы «Меню» представлен на рисунке 45.

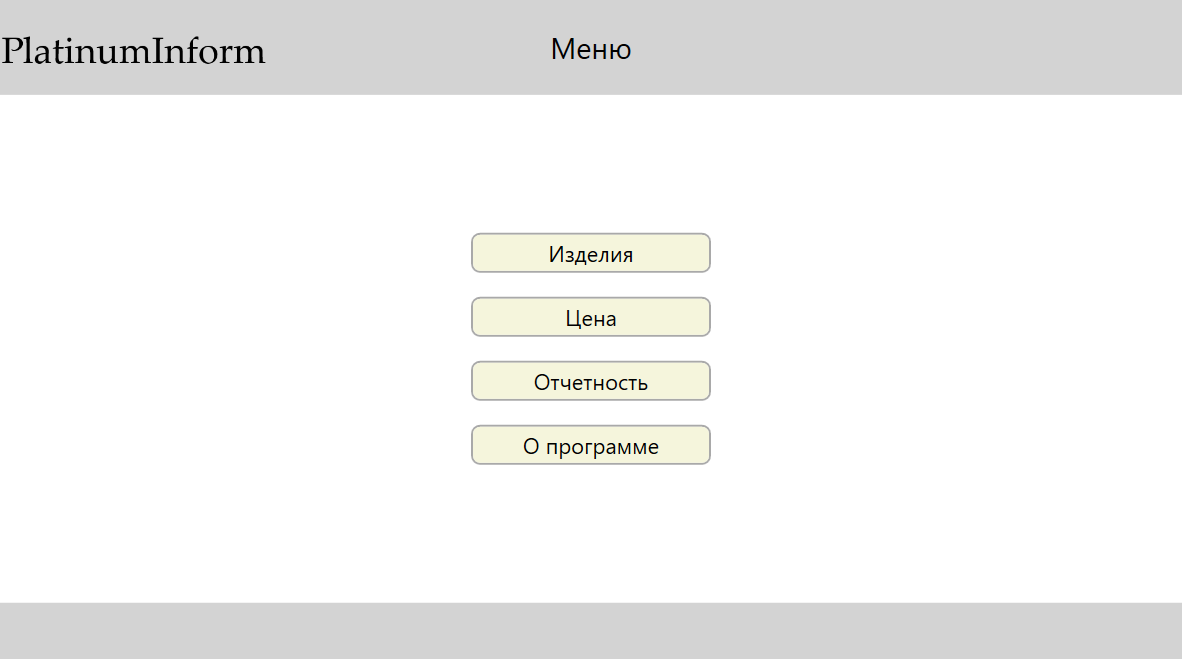


Рис. 45

Кнопка «Изделия» открывает новую страницу с данными по изделиям.

Кнопка «Цена» открывает новую страницу с данными о ценах на изделия.

Кнопка «Отчетность» открывает новую страницу отчетности.

Кнопка «О программе» открывает новое окно с информацией о приложении.

Страница «Изделия»

Для того чтобы перейти на данную страницу, необходимо в главном меню нажать на кнопку «Изделия». Интерфейс страницы представлен на рисунке 46.

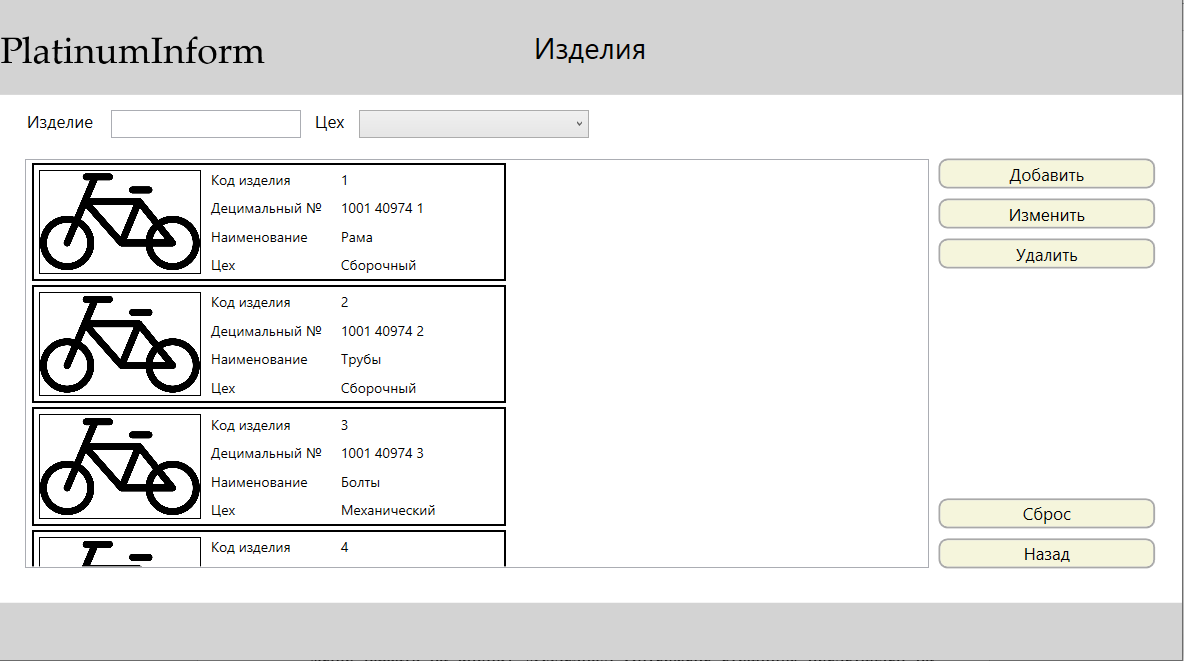


Рис. 46 – страница «Изделия»

Здесь Вы можете просматривать информацию, добавлять, изменять и удалять данные об изделиях, а также осуществлять поиск по двум критериям: наименованию изделия и цеху производства. Поиск может осуществляться совместно, при необходимости моментального сброса поиска, необходимо нажать на кнопку «Сброс».

Страница «Добавление/изменение данных»

Для того чтобы прейти к данной странице, необходимо в изделия нажать на кнопку «Добавить» или выбрать запись и нажать на кнопку «Изменить». Интерфейс страницы представлен на рисунке 47.

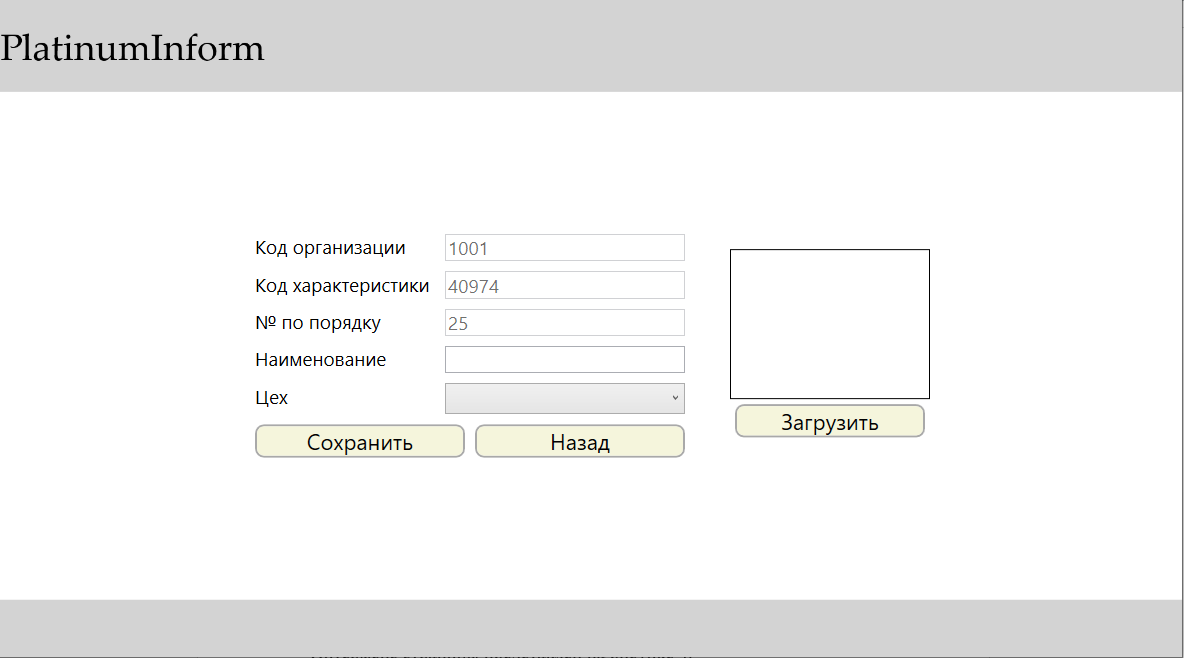


Рис. 47 – страница «Добавление/изменение данных»

Здесь необходимо указать наименование изделия, цех производства и фотографию изделия, для этого надо нажать на кнопку «Загрузить». Для того чтобы сохранить данные, необходимо нажать на кнопку «Сохранить». Если необходимо выйти с данной страницы, нужно нажать на кнопку «Назад».

Страница «Цена»

Для того чтобы перейти к данной странице, необходимо в главном меню нажать на кнопку «Цена». Интерфейс страницы представлен на рисунке 48.

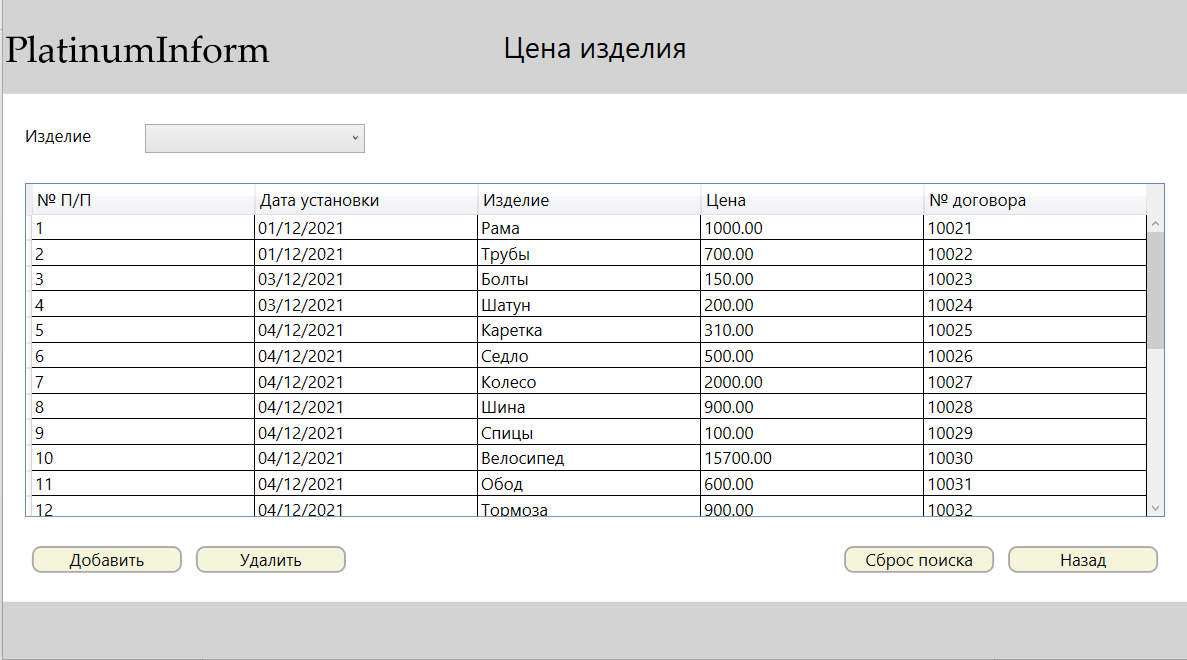


Рис. 48 – страница «Цена»

Здесь Вы можете просматривать информацию по ценам изделий, добавлять и удалять данные, а также произвести поиск по изделию. Для сброса поиска необходимо нажать на кнопку «Сброс поиска».

Страница «Добавление данных»

Для того чтобы перейти к данной странице, необходимо на странице «Цена» нажать на кнопку «Добавить». Интерфейс страницы представлен на рисунке 49.

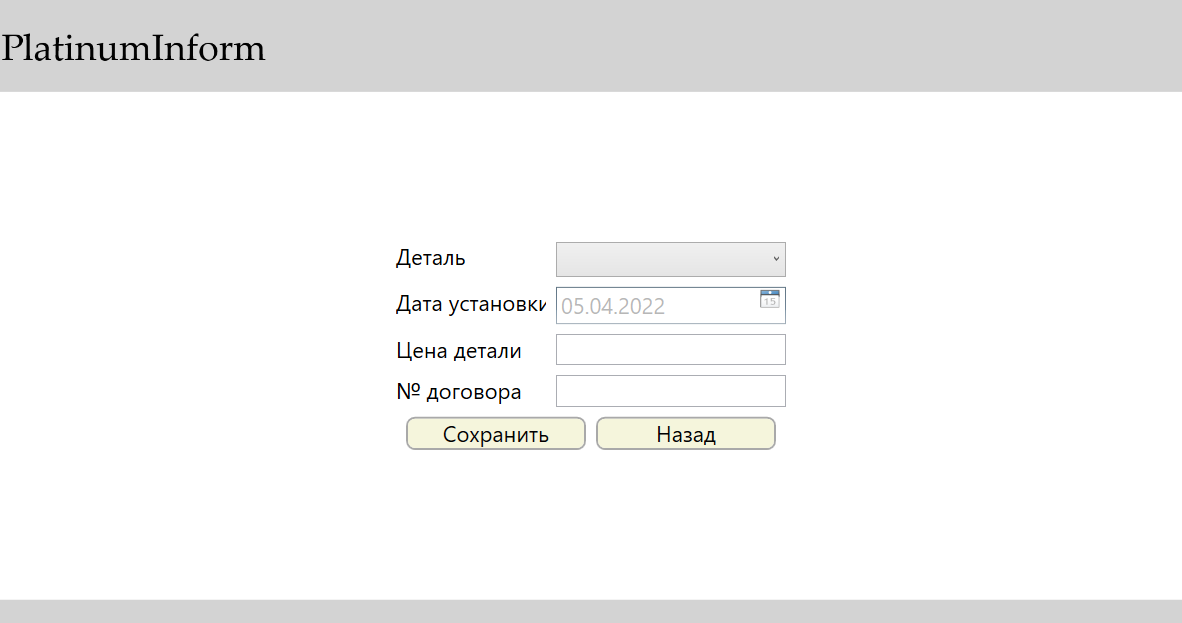


Рис. 49 – страница «Добавление данных»

Здесь необходимо указать изделие, дату установку цены, цену изделия, а также номер договора, по которому осуществляется добавление новой цены. Для того чтобы сохранить данные, то надо нажать на кнопку «Сохранить». Если необходимо выйти с данной страницы, то надо нажать на кнопку «Назад».

Страница «Отчётность»

Для того чтобы перейти к данной страницы, необходимо в главном меню нажать на кнопку «Отчетность». Интерфейс страницы представлен на рисунке 50.

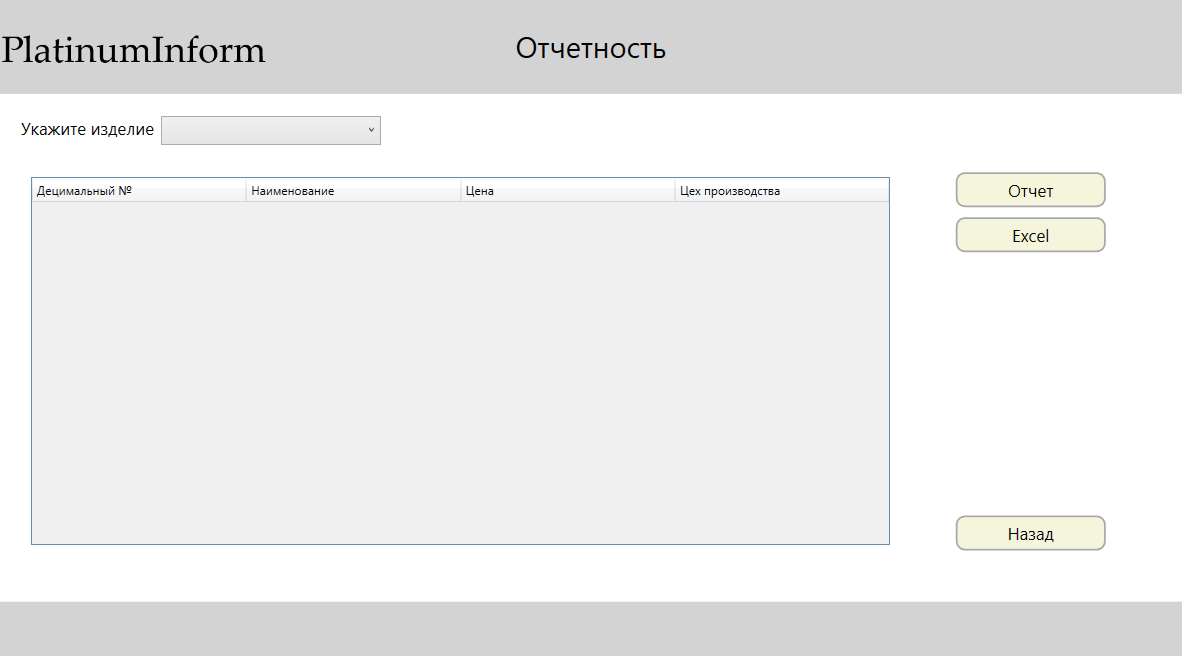


Рис. 50 – страница «Отчетность»

Здесь Вы можете сформировать информационные документ в формате .docx, а также выгрузить статистику по ценам изделий в формате .xlsx. Для того чтобы сформировать документ в Word, необходимо выбрать изделие, нажать на него и затем нажать на кнопку «Отчет». Чтобы сформировать документ в Excel, необходимо нажать на кнопку «Excel».

Окно «О программе»

Для того чтобы перейти к данному окну, необходимо в главном меню нажать на кнопку «О программе». Интерфейс окна представлен на рисунке 51.

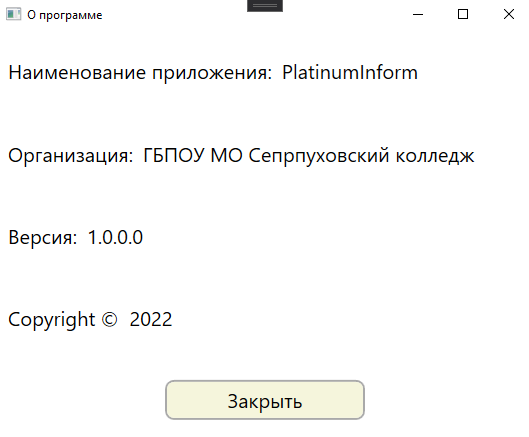


Рис. 52 – окно «О программе»

Здесь Вы можете просмотреть информацию по данному приложению.

Сообщения пользователю представлены в таблице 25.

|  |  |
| --- | --- |
| Сообщение | Описание действий при получении сообщения |
| Выберите запись | Необходимо выбрать запись для дальнейшей работы |
| Не удалось удалить запись | Убедитесь, что выбранная запись больше нигде не используется |
| Произошла ошибка подключения к базе данных | Убедитесь, что соединение с базой данных осуществлено |
| Запись не найдена | Проверить введенные данные при поиске. |
| Отсутствует шаблон | Проверить наличие шаблона документа |

## 2.8. Разработка политики безопасности

Основной целью, на достижение которой направлены все положения настоящей Политики, является защита информационных ресурсов от возможного нанесения им материального, физического, морального или иного ущерба, посредством случайного или преднамеренного взаимодействия на информацию, её носители, процессы обработки и передачи, а также минимизация рисков ИБ.

В данной программе была использована авторизация пользователя, данные для которой хранятся в базе, а также производится шифрование пароля.

Метод шифрования представлен в листинге 21.

public static string Encrypt(string value)

{

StringBuilder Sb = new StringBuilder();

using (SHA256 hash = SHA256Managed.Create())

{

Encoding enc = Encoding.UTF8;

Byte[] result = hash.ComputeHash(enc.GetBytes(value));

foreach (Byte b in result)

Sb.Append(b.ToString("x2"));

}

return Sb.ToString();

}

Листинг 21 – метод шифрования

## 2.9. Эксплуатация и развитие программы

Данное приложение было разработано непосредственно для организации АО «РАТЕП» для улучшения автоматизации процесса.

До этого момента работа по введению нормативно-справочной информации, изготавливаемых предприятием, происходила вручную, без каких-либо ПО. Вся информация по изделиям заносилась на бумажный носитель, не имея дополнительного экземпляра в электронном виде. В связи с этим, данное приложение улучшит процесс введения информации и упростит работу сотрудников.

Таким образом можно сделать вывод, что данная программа будет иметь дальнейшее сопровождение и в случае успеха сможет стать программой для массового использования малых торговых организаций.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной производственной практики было разработано приложения для учета нормативно-справочной информации по изделиям, изготавливаемых предприятием.

Первым этапом был произведен анализ актуальности темы и сравнения уже существующих разработок по данной теме, а именно «ПОЛИНОМ:MDM», «Supasoft CRM», «1С:Предприятие». Сравнивались функциональные особенности приложений, их архитектуры и технологии, при помощи которых разработаны данные приложения.

Для организации хранения данных приложения были рассмотрены основные системы управления базами данных и основываясь на используемых технологиях для написания приложения и интеграции с уже имеющимися сервисами было принято использовать Microsoft SQL Server.

Клиентское приложение написано с использованием языка C#. Программа используется только одним пользователем. Согласно роли – инженер-технолог, ему доступен следующий функционал:

* просмотр информации по изделиям;
* добавление новой информации по изделиям;
* возможность редактирования и удаления данных;
* просмотр информации по цене изделия;
* добавление новой цены изделия;
* возможность удаления информации;
* формирования справочника по изделию;
* формирование графика.

Благодаря использованию различных пользовательских элементов управления, было разработано приложение с удобным и интуитивно понятным пользовательским интерфейсом.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. CyberForum [Электронный ресурс] (Дата обращения 02.01.2022) URL: https://www.cyberforum.ru/
2. ProfessorWeb C# [Электронный ресурс] (Дата обращения 05.01.2022) URL: https://professorweb.ru/my/csharp/charp\_theory/level1/index.php
3. ProfessorWeb WPF [Электронный ресурс] (Дата обращения 05.01.2022) URL: https://professorweb.ru/my/WPF/base\_WPF/level1/base\_WPF\_index.php
4. ProfessorWeb Entity Framework [Электронный ресурс] (Дата обращения 07.01.2022) URL: https://professorweb.ru/my/entity-framework/6/level1/
5. Habr C# [Электронный ресурс] (Дата обращения 10.01.2022)

URL: https://habr.com/ru/hub/csharp/

1. Metanit [Электронный ресурс] (Дата обращения 10.01.2022) URL: https://metanit.com/
2. SQL видеокурс [Электронный ресурс] (Дата обращения 03.01.2022) URL: <https://www.youtube.com/watch?v=9UlJzmaNCYI>
3. CodeRoad [Электронный ресурс] (Дата обращения 22.02.2022) URL: https://coderoad.ru/
4. Документация по SQL [Электронный ресурс] (Дата обращения: 9.02.2021) URL: <https://docs.microsoft.com>
5. SQL Задачи и решения [Электронный ресурс] (Дата обращения 10.02.2022) URL: <http://www.sql-tutorial.ru/>
6. Официальная документация Microsoft по языку T-SQ [Электронный ресурс] (Дата обращения 15.01.2022)

URL: https://docs.microsoft.com/ru-ru/sql/?view=sql-server-ver15

1. Официальная документация Microsoft по языку C# [Электронный ресурс] (Дата обращения: 20.01.2022) URL: https://docs.microsoft.com/ru-ru/
2. Vscode [Электронный ресурс] (Дата обращения 25.01.2022) URL: https://vscode.ru/
3. Try-catch (Справочник по C#) [Электронный ресурс] (Дата обращения 10.01.2022) URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/try-catch
4. Habr Q&A. Ограничение символов [Электронный ресурс] (Дата обращения 17.02.2022) URL: https://qna.habr.com/q/739221

ПРИЛОЖЕНИЕ 1 – IDEF0

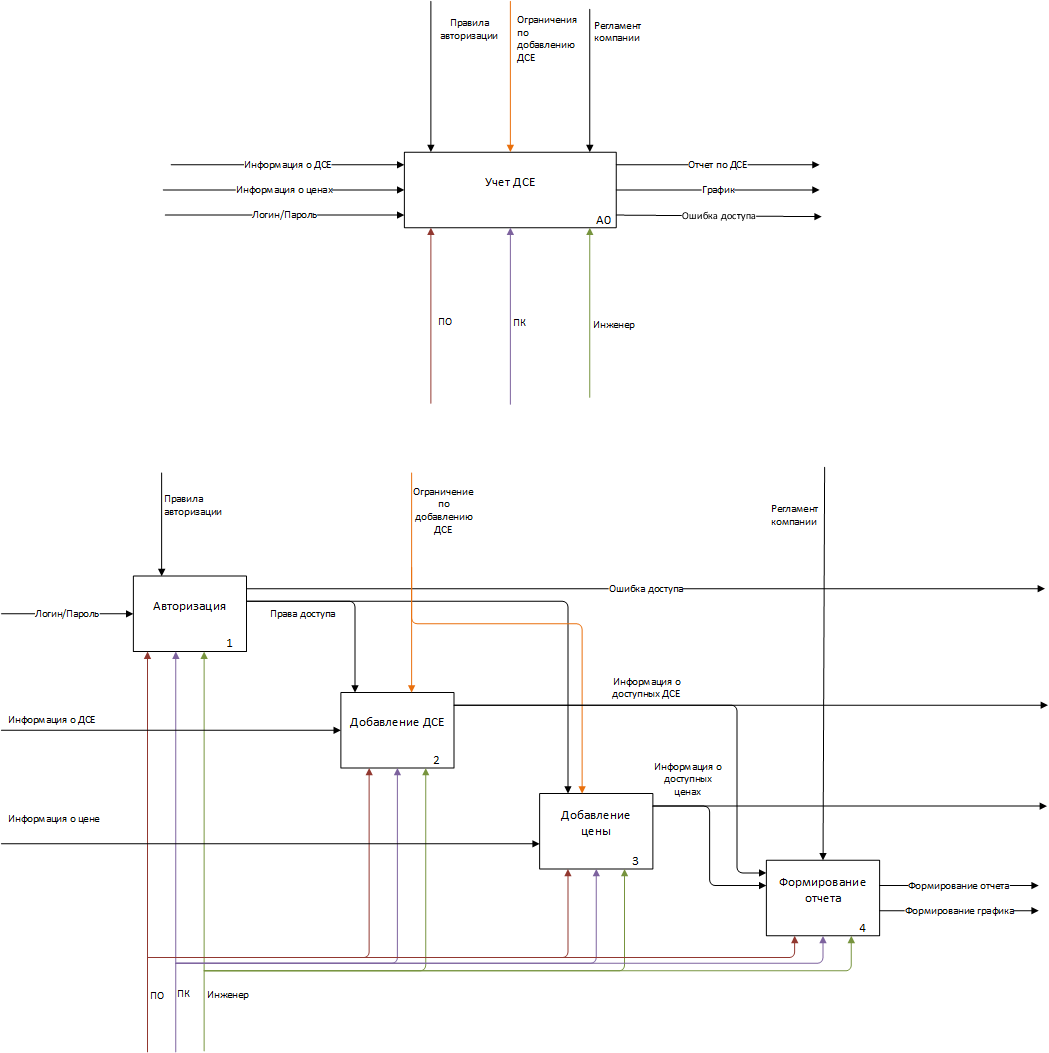


Рис. 53 – IDEF0

ОКОНЧАНИЕ ПРИЛОЖЕНИЯ 1

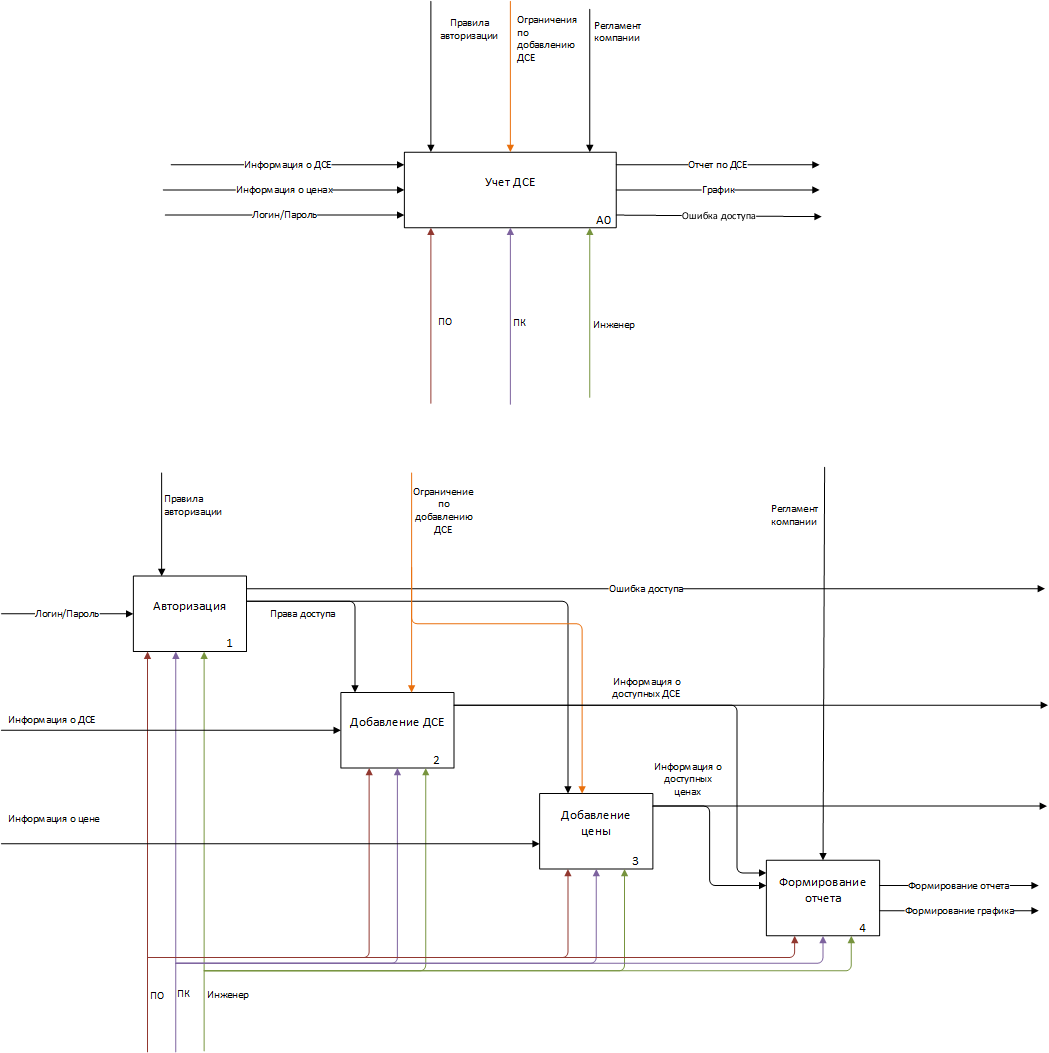


Рис. 54 – IDEF0

ПРИЛОЖЕНИЕ 2 – DFD

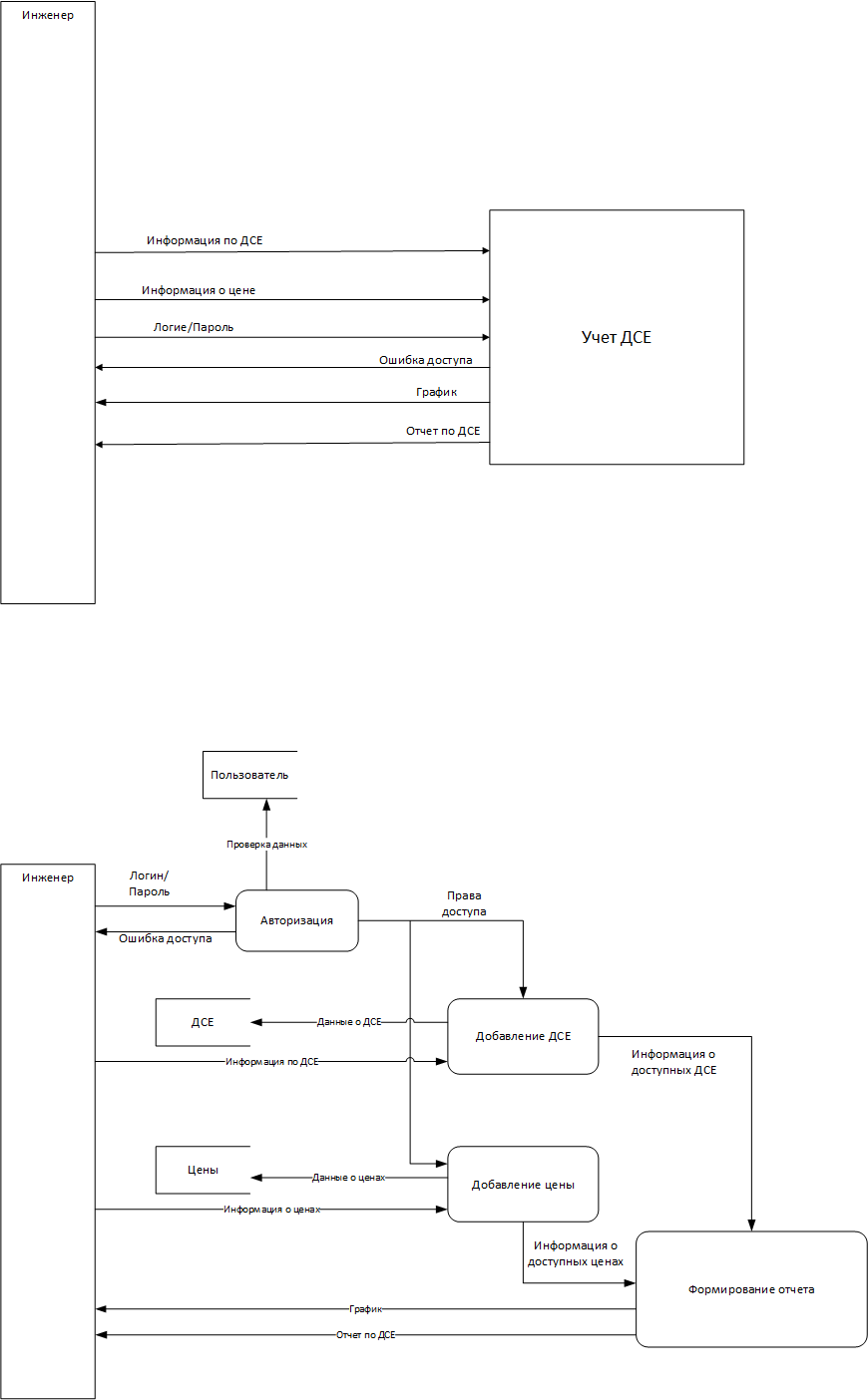


Рис. 55 – DFD

ОКОНЧАНИЕ ПРИЛОЖЕНИЯ 2

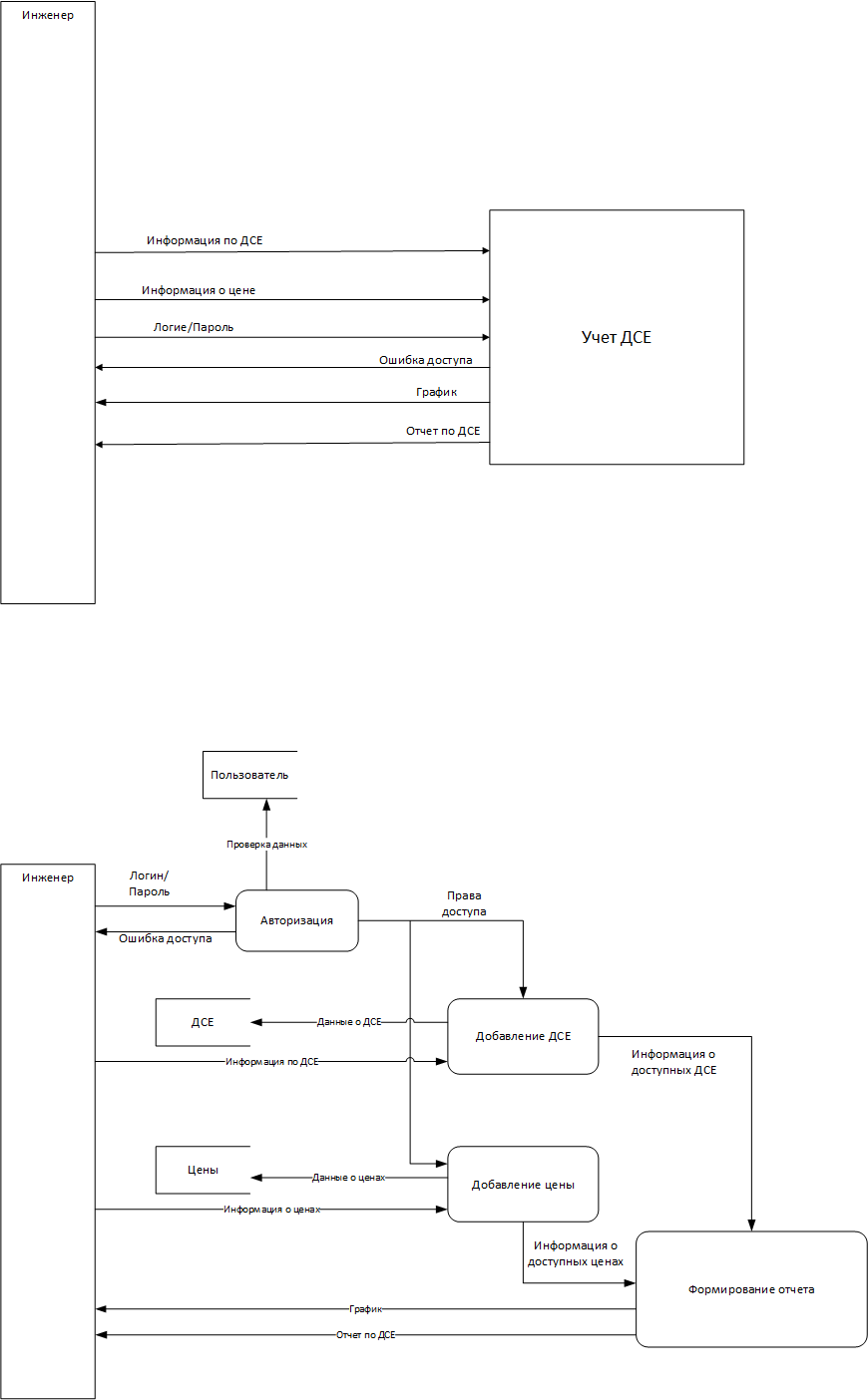


Рис. 56 – DFD

ПРИЛОЖЕНИЕ 3 – ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

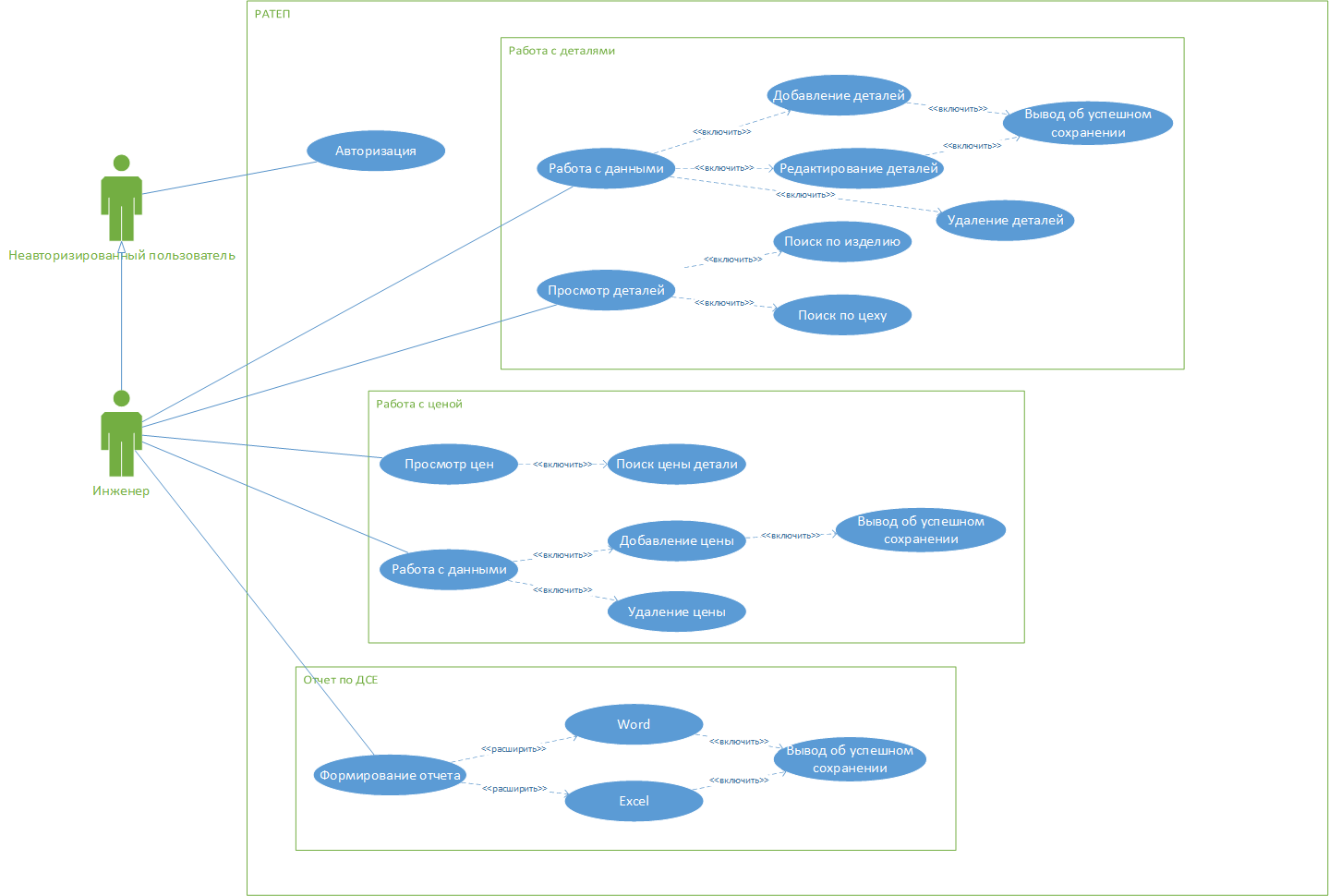


Рис. 57 – Диаграмма вариантов использования

ПРИЛОЖЕНИЕ 4 – ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ

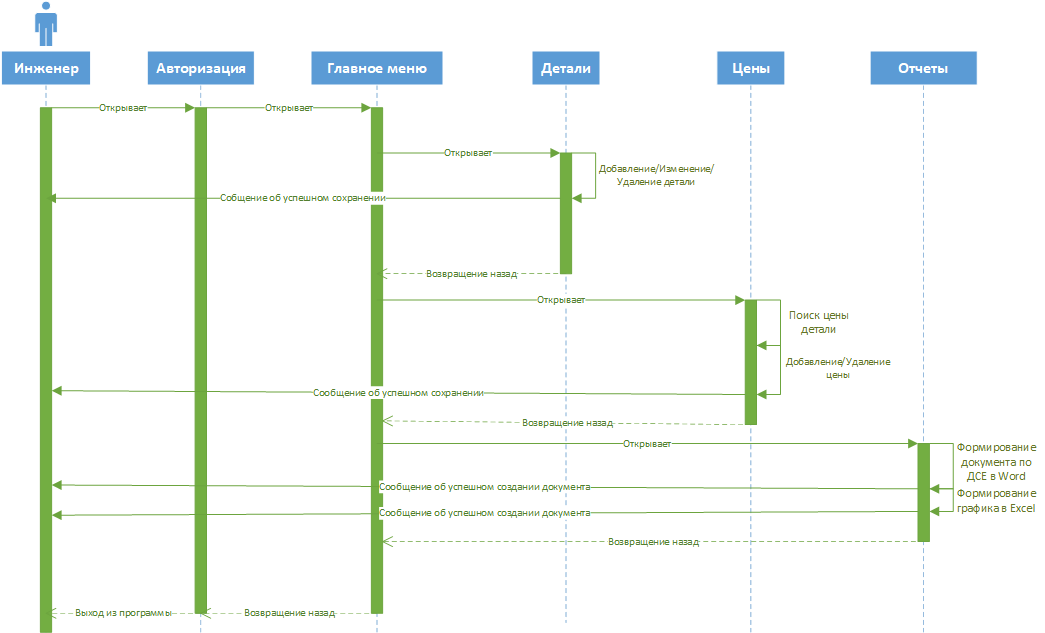


Рис. 58 – диаграмма последовательности

ПРИЛОЖЕНИЕ 5 – ДИАГРАММА КЛАССОВ

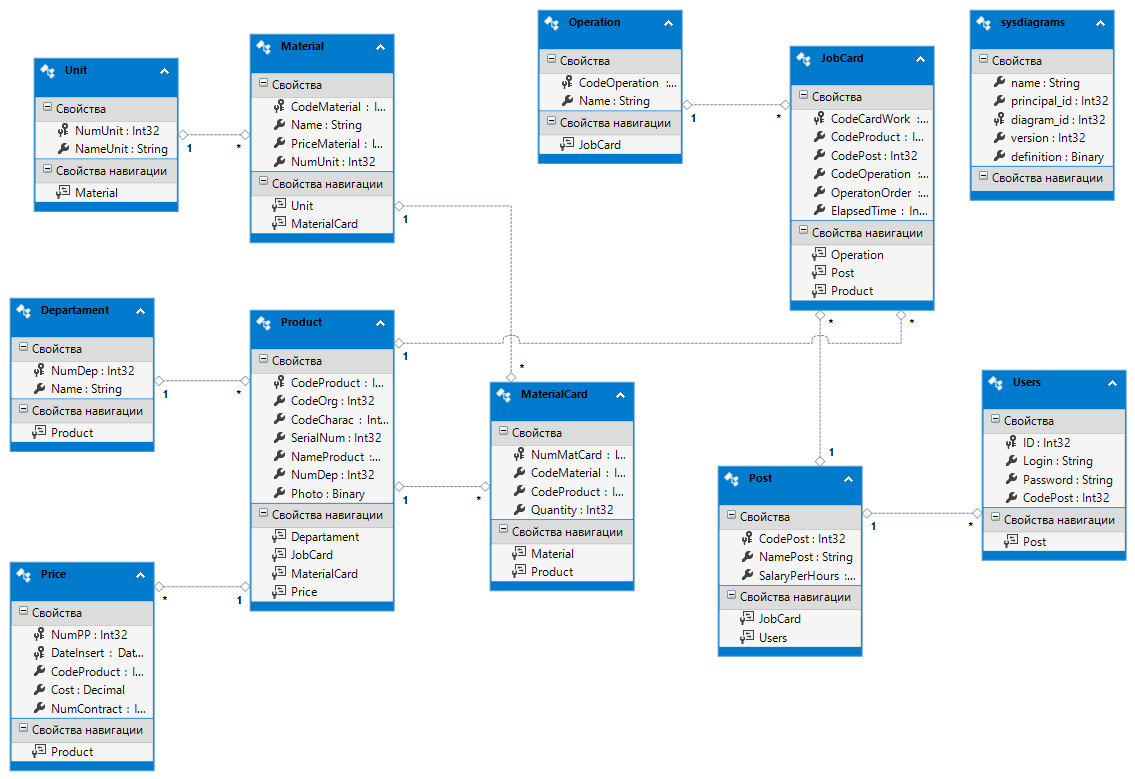


Рис. 59 – диаграмма классов

ПРИЛОЖЕНИЕ 6 – ДИГРАММА ДЕЯТЕЛЬНОСТИ

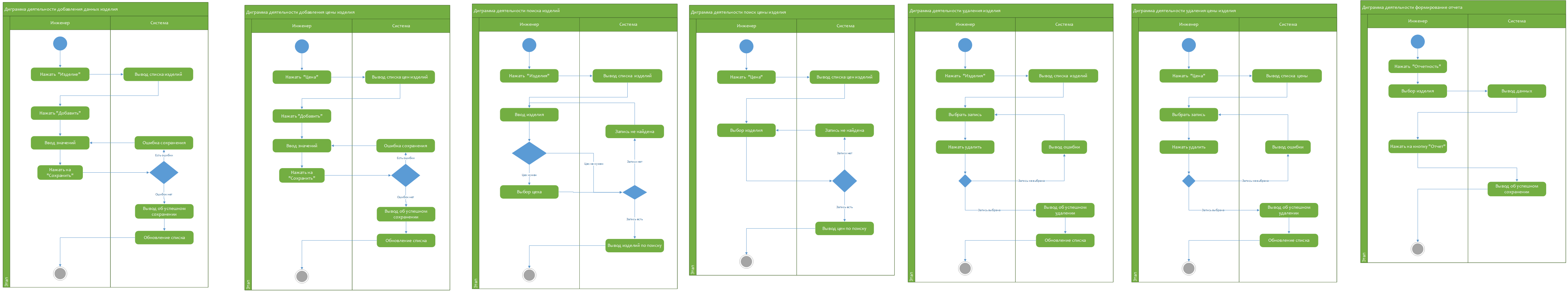


Рис. 60 – дигамма деятельности

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 6

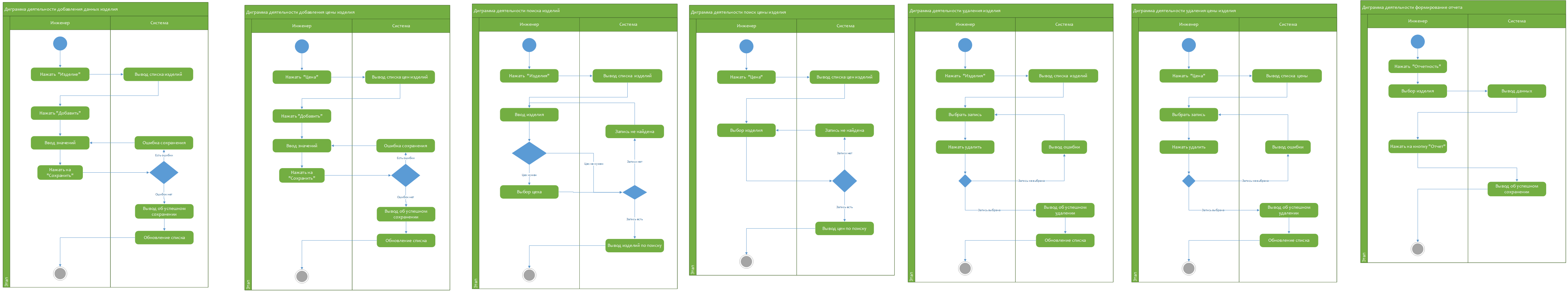


Рис. 61 – диаграмма деятельности

ОКОНЧАНИЕ ПРИЛОЖЕНИЯ 6

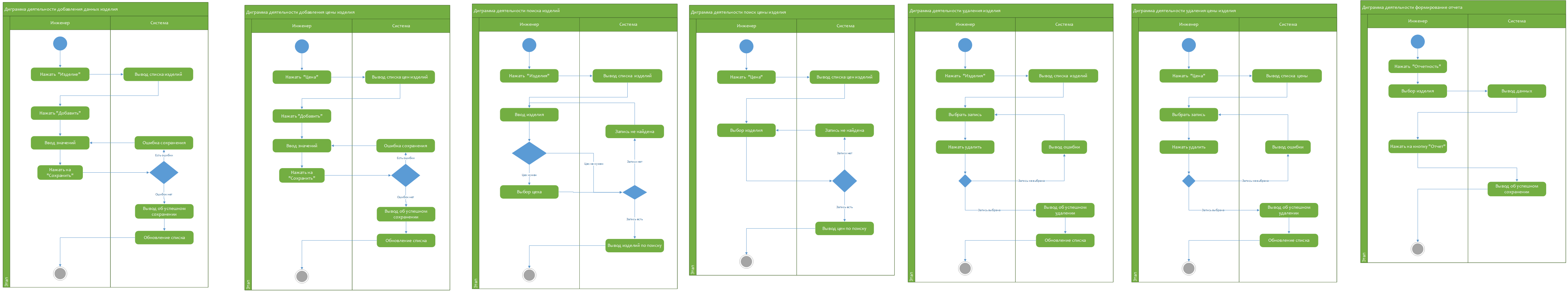


Рис. 62 – диаграмма деятельности

ПРИЛОЖЕНИЕ 7 – ДИАГРАММА РАЗВЕРТЫВАНИЯ

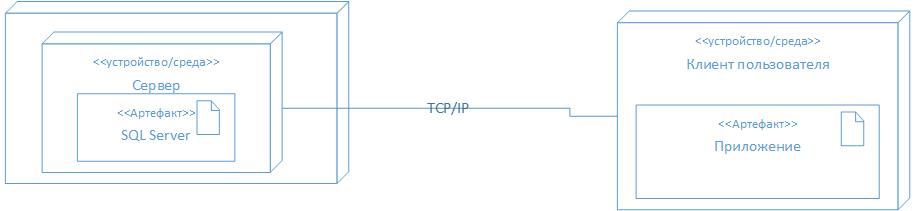


Рис. 63 – диаграмма развертывания

ПРИЛОЖЕНИЕ 8 – АРХИТЕКТУРА ПРИЛОЖЕНИЯ

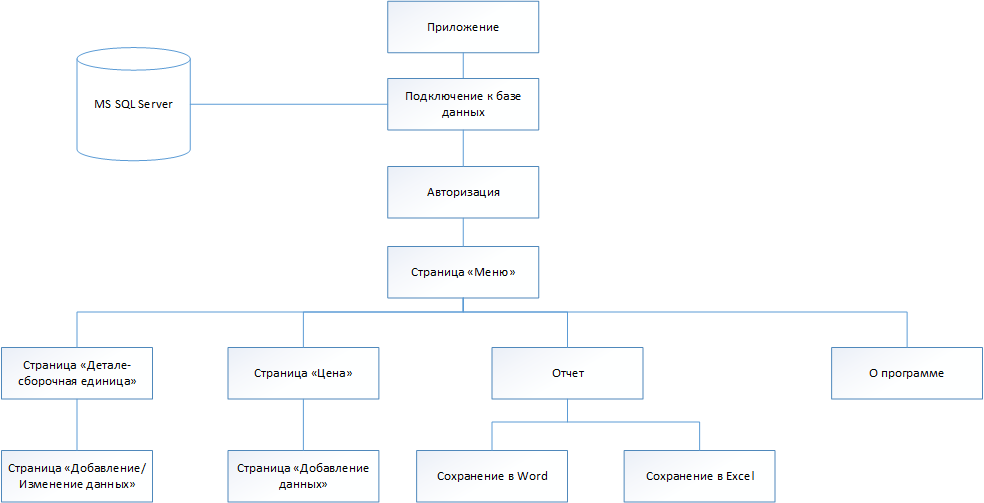


Рис. 64 – архитектура приложения

ПРИЛОЖЕНИЕ 9 – ЛОГИЧЕСКАЯ МОДЕЛЬ

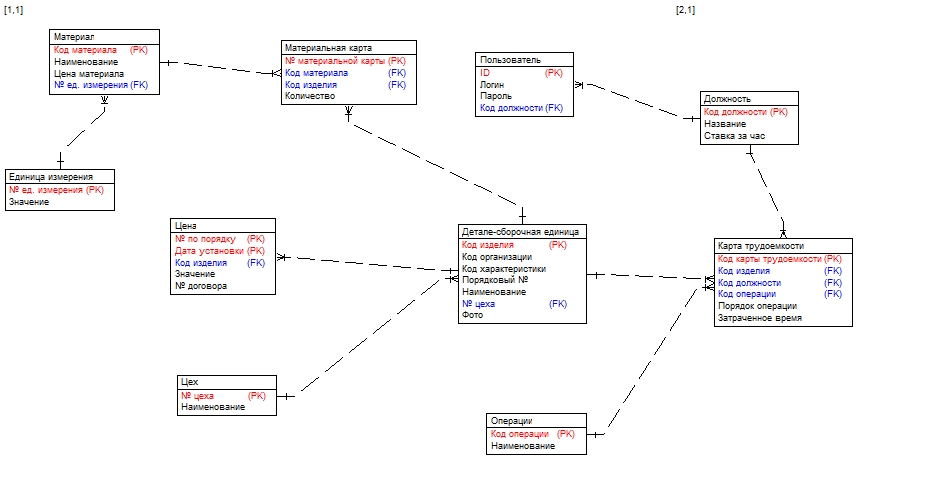


Рис.65 – логическая модель

ПРИЛОЖЕНИЕ 10 – ФИЗИЧЕСКАЯ МОДЕЛЬ

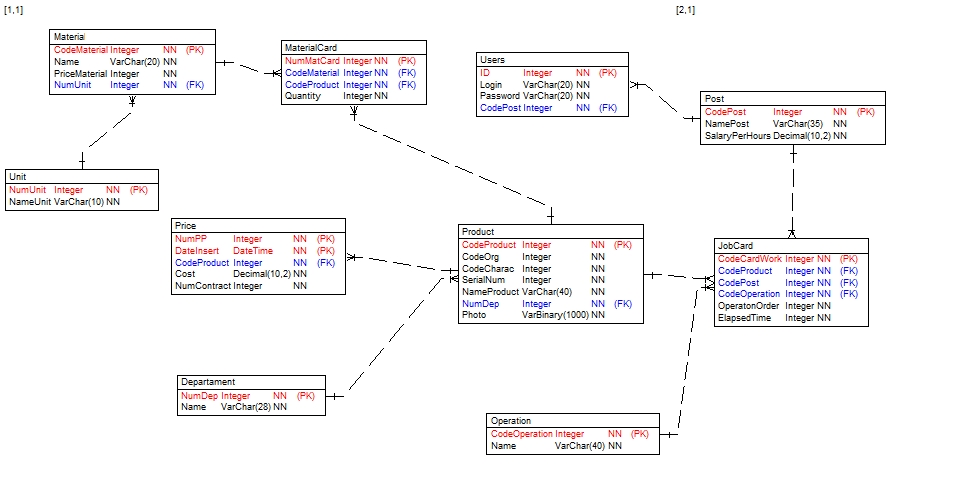


Рис. 66 – физическая модель