Содержание

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Назначение и цели разработки | 4 |
| 2 Разработка технического проекта на основе анализа требований | 5 |
| 2.1 Определение спецификаций программного обеспечения | 5 |
| 2.2 Проектирование модели данных и диаграммы классов | 14 |
| 2.3 Конструирование прототипа | 21 |
| 3 Реализация | 30 |
| 3.1 Обоснование выбора средств разработки | 30 |
| 3.2 Разработка базы данных в среде СУБД | 31 |
| 3.3 Описание программных модулей | 35 |
| 4 Тестирование программных модулей | 54 |
| 4.1 Интеграционное тестирование | 54 |
| 4.2 Модульное тестирование | 54 |
| 5 Эксплуатационная документация | 56 |
| Заключение | 58 |
| Список использованных источников | 60 |
| Приложение А Техническое задание. Требования к программным модулям | 63 |
| Приложение Б Основные экранные формы и программный код | 68 |
| Приложение В Формы выходных документов | 82 |
| Приложение Г Тест-кейсы | 83 |
|  |  |

Введение

С развитием технологий создаются новые предприятия. С расширением бизнеса, технологии приходят на помощь и делают работу предприятия проще. В розничной торговле предприниматели должны быть готовы к резким изменениям потребительских предпочтений, чтобы обеспечить максимальное их удовлетворение. От скорости реакции на появление новых популярных продуктов во многом зависит прибыльность торгового бизнеса. Расчет показателей продаж нужно проводить регулярно, обеспечивая себя актуальными маркетинговыми данными.

Разработка уникальной подсистемы, которая учитывает все особенности торговли розничного магазина, будет выгодным вариантом для ведения бизнеса.

Основной задачей курсового проекта является разработка и внедрение подсистемы анализа продаж торговли розничного магазина одежды «Azur», которая позволит увеличить качество выбора решений при ведении продаж товаров.

1 Назначение и цели разработки

Заданием предусмотрена разработка подсистемы анализа продаж розничного магазина одежды «Azur». Программная подсистема имеет название «Azur», далее будет называться так.

«Azur» предназначена для автоматизации процессов анализа и помощи в выборе верных решений по ведению торговли, на основе учёта ведения товара, складского учёта, продаж товара клиентам.

Основными целями создания «Azur» являются:

* повышение эффективности исполнения процессов, путем сокращения непроизводительных и дублирующих операций, операций, выполняемых «вручную», оптимизации информационного взаимодействие участников процессов;
* повышение качества принятия управленческих решений за счет оперативности представления, полноты, достоверности и удобства форматов отображения информации.

2 Разработка технического проекта на основе анализа требований

2.1 Определение спецификаций программного обеспечения

Рассмотрим определение вариантов использования (прецедентов).

Подсистема анализа требуется прежде всего следующим заинтересованным лицам:

* товароведу;
* менеджеру по продажам;
* директору магазина;
* работник кадровой службы.

На начальном этапе создания системы можно ограничиться только тремя важными ролями действующих лиц: менеджер по продажам, товаровед и администратор (он же работник кадровой службы). Соответственно основные прецеденты (варианты использования) для разрабатываемой системы следующие.

Основные прецедент для менеджера по продажам:

* П1 – авторизация в системе;
* П2 – просмотр данных о товарах;
* П3 – просмотр данных анализа;
* П4 – работа с заявками на поставку;
* П5 – ведение данных о продажах.

Основные прецедент для товароведа:

* П6 – авторизация в системе;
* П7 – просмотр данных о товарах;
* П8 – ведение данных о товарах;
* П9 – ведение приходных накладных.

Основные прецедент для администратора:

* П10 – авторизация в системе;
* П11 – просмотр историй входа;
* П12 – формирование (удаление) аккаунтов;
* П13 – регистрация (удаление) пользователей и др.

Представим диаграмму вариантов использования, созданную средствам MS Visio 2019 (профессиональная версия), для проектируемого программного обеспечения на рисунке 1.

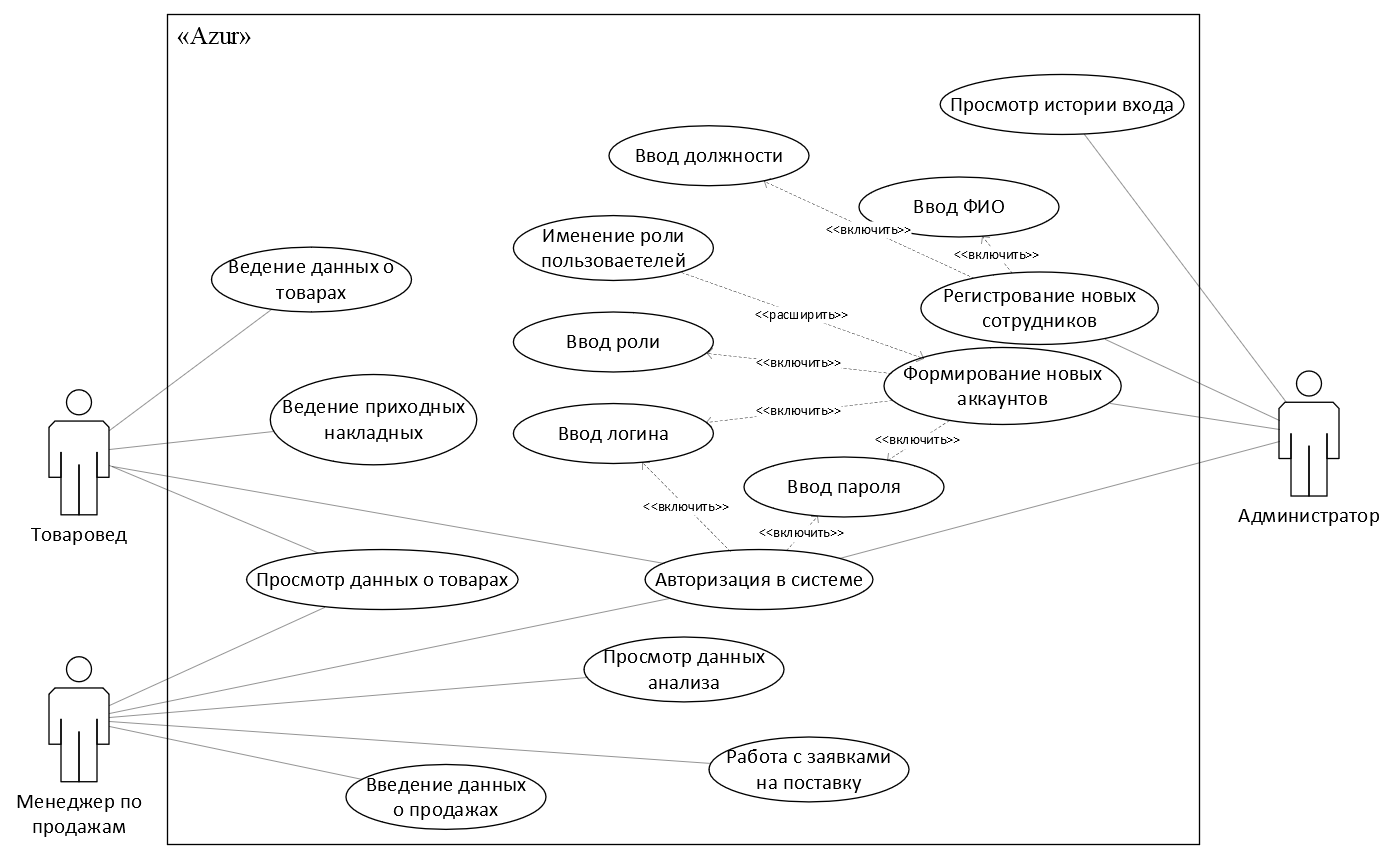


Рисунок 1 – Диаграмма вариантов использования программной подсистемы

«Azur»

Опишем подробно требования к реализации базовых вариантов использования, представленных на рисунке 1.

В таблице 1 представлено описание главного (основного) раздела сценария варианта использования (прецедента) «Работа с анализами данных».

Таблица 1- Главный раздел сценария варианта использования «Работа с данными

анализа»

|  |  |
| --- | --- |
| Вариант использования (прецедент) | Работа с анализами данных |
| Актеры | Менеджер по продажам, система |
| Краткое описание | После входа менеджера в систему, проверяется актуальность информации. Система на этой основе составляет графики и таблицы, проводит анализ и рекомендует менеджеру по продажам предпринять какие-либо действия. |
| Цель | Проведение достоверного анализа |
| Тип | Базовый |
| Ссылки на другие варианты использования | Включает в себя варианты использования:  авторизация;  составление рекомендации по закупкам;  изменение программы лояльности;  ввод данных о продажах;  просмотр входных данных. |

В таблице 2 описана последовательность действий, приводящая к успешному выполнению варианта использования «Работа с данными анализа».

Таблица 2 – Сценарий успешного выполнения варианта использования «Работа с данными анализа»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| 1.Менеджер по продажам входит в систему. | 2. Система определяет актуальность данных  Исключение 1. Данные не актуальны.  3. Система выводит сообщение, что данные актуальны. |
| 4. Менеджер по продажам просматривает данные об анализе. | 5. Система предоставляет данные, запрашиваемые менеджером |

Далее опишем сценарии исключений.

В таблице 3 представлены сценарии обработки исключительных ситуаций для варианта использования «Работа с данными анализа».

Таблица 3 – Обработка исключительных ситуаций для варианта использования «Работа с данными анализа»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| Исключение 1. Данные не актуальны. | |
| 1.Менеджер по продажам входит в систему. | 2.Система определила данные как не актуальные.  3. Система оповещает менеджера и просит обновить данные |
| 4. Менеджер обновляет данные. | 5. Система фиксирует изменение и предоставляет актуальные графики. |

Опишем алгоритм реализации описанных сценариев для варианта использования «Работа с данными анализа» с помощью диаграммы деятельности, представленной на рисунке 2.

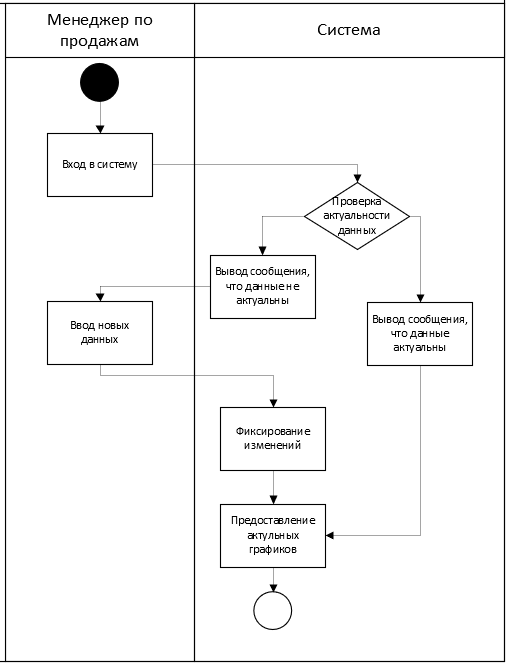


Рисунок 2 – Диаграмма деятельности для варианта использования «Работа с данными анализа»

В таблице 4 представлено описание раздела сценария варианта использования (прецедента) «Ввод данных о продажах».

Таблица 4 – Главный раздел сценария варианта использования «Ввод данных о продажах»

|  |  |
| --- | --- |
| Вариант использования (прецедент) | Ввод данных о продажах |
| Актеры | Менеджер по продажам, система |
| Краткое описание | Менеджер по продажам вносит продажи товаров по определённым периодам времени. Система фиксирует изменения. |
| Цель | Ввод корректных данных о продажах |
| Тип | Базовый |
| Ссылки на другие варианты использования | Включает в себя варианты использования:  авторизация;  ввод данных о продажах;  просмотр входных данных. |

В таблице 5 описана последовательность действий, приводящая к успешному выполнению варианта использования «Ввод данных о продажах».

Таблица 5 – Сценарий успешного выполнения варианта использования «Ввод данных о продажах»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
|  | 1.Система фиксирует конец рабочей смены |
| 2. Менеджер по продажам вносит данные о продажах товара.  Исключение 1. Менеджер не внес данные | 3. Система сохраняет данные и выводит изменённые графики за период  Исключение 2. Некоторые показатели товаров достигли критических отметок.  4. Система уведомляет пользователя, что торговля идёт хорошо |

Далее опишем сценарии исключений.

В таблице 6 представлены сценарии обработки исключительных ситуаций для варианта использования Ввод данных о продажах»

Таблица 6 – Обработка исключительных ситуаций для варианта использования «Ввод данных о продажах»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| Исключение 1. Менеджер не внес данные вовремя. | |
| 1. Менеджер не внёс данные вовремя. | 2. Система оповещает менеджера о том, что необходимо ввести данные для получения корректных данных анализа |
| Исключение 2. Некоторые показатели товаров достигли критических отметок. | |
| 2. Результаты анализа показали, что некоторые товары достигли критических отметок. | 3. Система предлагает менеджеру предпринять действия, которые улучшат ситуацию. |

Опишем алгоритм реализации описанных сценариев для варианта использования «Ввод данных о продажах» с помощью диаграммы деятельности, представленной на рисунке 3.

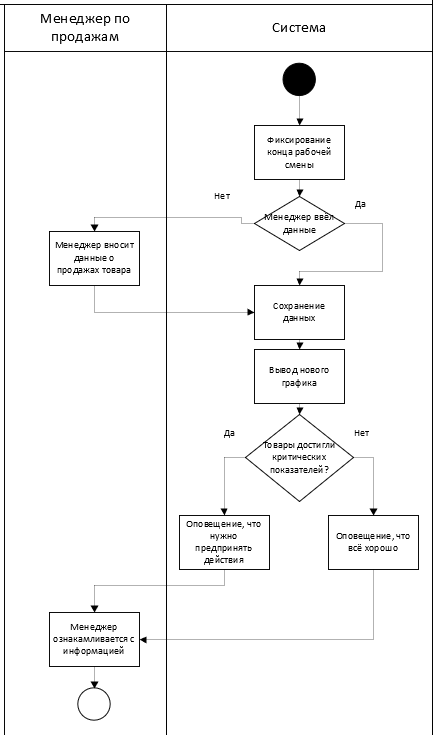


Рисунок 3 – Диаграмма деятельности для варианта использования «Ввод данных о продажах»

В таблице 7 представлено описание раздела сценария варианта использования (прецедента) «Ввод данных о товарах».

Таблица 7 – Главный раздел сценария варианта использования «Ввод данных о товарах»

|  |  |
| --- | --- |
| Вариант использования (прецедент) | Ввод данных о товарах |
| Актеры | Менеджер по продажам, товаровед, система |
| Краткое описание | Товаровед при поступлении нового товара, вносит данные об этом товаре в систему с его описанием и основными характеристиками. После сохранения данных системой и написания отчёта товароведом, менеджер по продажам просматривает отчёт |
| Цель | Фиксирование нового товара |
| Тип | Базовый |
| Ссылки на другие варианты использования | Включает в себя варианты использования:  авторизация;  ввод данных о продажах;  ввод данных о товарах;  составление отчётов по новым товаров. |

В таблице 8 описана последовательность действий, приводящая к успешному выполнению варианта использования «Ввод данных о товарах».

Таблица 8 – Сценарий успешного выполнения варианта использования «Ввод данных о товарах»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| 1.Товаровед получает новый товар.  Исключение 1. Новое поступление товара не осуществляется. | 2. Товаровед вносит новый товар в систему.  Исключение 2. Подобный товар уже существует.  3. Система сохраняет ввод новых товаров. |
| 4. Товаровед составляет отчёт по проделанной работе и отправляет менеджеру по продажам  Исключение 3. Товаровед не составил отчёт | 5. Менеджер по продажам получил отчёт от товароведа |
| 6. Менеджер проверил корректность введённых данных | 7. Система фиксирует отзыв менеджера |

Далее опишем сценарии исключений.

В таблице 9 представлены сценарии обработки исключительных ситуаций для варианта использования «Работа с анализами данных».

Таблица 9 – Обработка исключительных ситуаций для варианта использования «Ввод данных о товарах»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| Исключение 1. Новое поступление товара не осуществляется. | |
| 1.Товаровед не получил новый товар | 4. Товаровед подтверждает, что данные о товарах актуальны и изменения не требуют. |
| Исключение 2. Подобный товар уже существует. | |
| 2. Товаровед вносит новый товар в систему. | 3. Система уведомляет товароведа о уже существующим таком товаре  4. Система предлагает изменить уже существующий товар. |
| Исключение 3. Товаровед не составил отчёт | |
| 4. Товаровед не составил отчёт. | 5. Система оповещает товароведа о том, что необходимо составить отчёт. |

Опишем алгоритм реализации описанных сценариев для варианта использования «Ввод данных о товарах» с помощью диаграммы деятельности, представленной на рисунке 4.

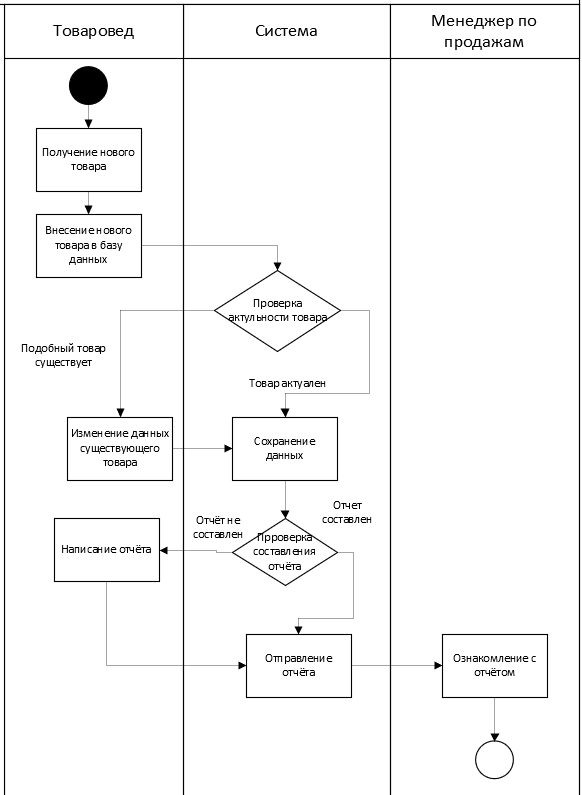


Рисунок 4 – Диаграмма деятельности для варианта использования «Ввод данных о товарах»

На следующем этапе опишем требования к реализации сценариев с учетом динамики во времени на уровне сообщений. Для это были разработаны диаграммы последовательности.

На рисунке 5 представлена диаграмма последовательности для варианта использования «Работа с данными анализа», которая отражает требования к синхронизации операций при реализации процесса продажи.

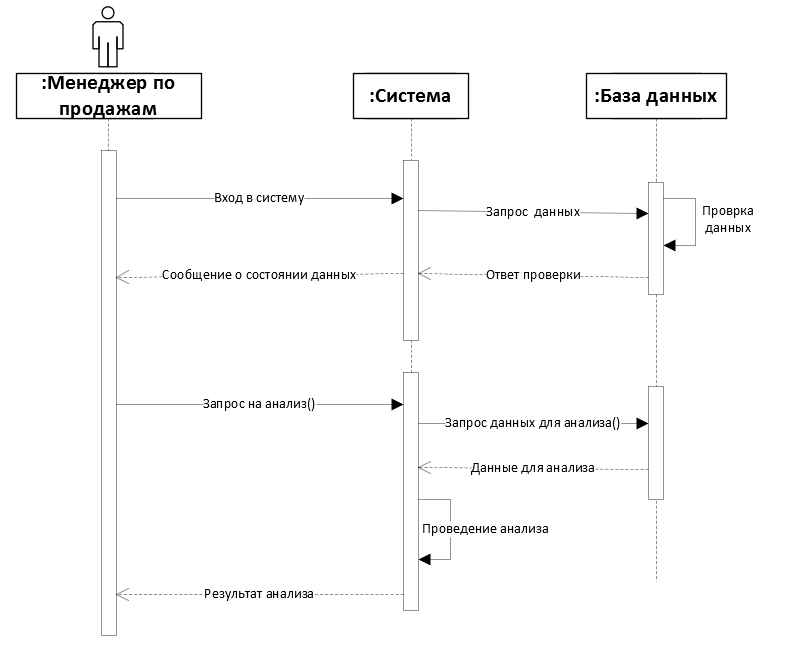


Рисунок 5 – Диаграмма последовательности для варианта использования «Продажа товара»

2.2 Проектирование модели данных и диаграммы классов

На основе анализа требований заказчика к функциям программной системы, требованиям к организации входных и выходных данных, с учетом спроектированных требований к реализации функций, описанных в диаграммах вариантов использования, деятельности и последовательности, была разработана модель данных системы и описана в виде ER-диаграммы, позволяющая четко описать требования к представлению логической структуры данных, на основе которой в последующем будет разработана физическая структура данных для хранения во внешней памяти и программной обработки.

Для построения ER-диаграммы применялись средства среды MS Visio.

На рисунке 6 представлена ER – диаграмма для программной подсистемы «Azur»

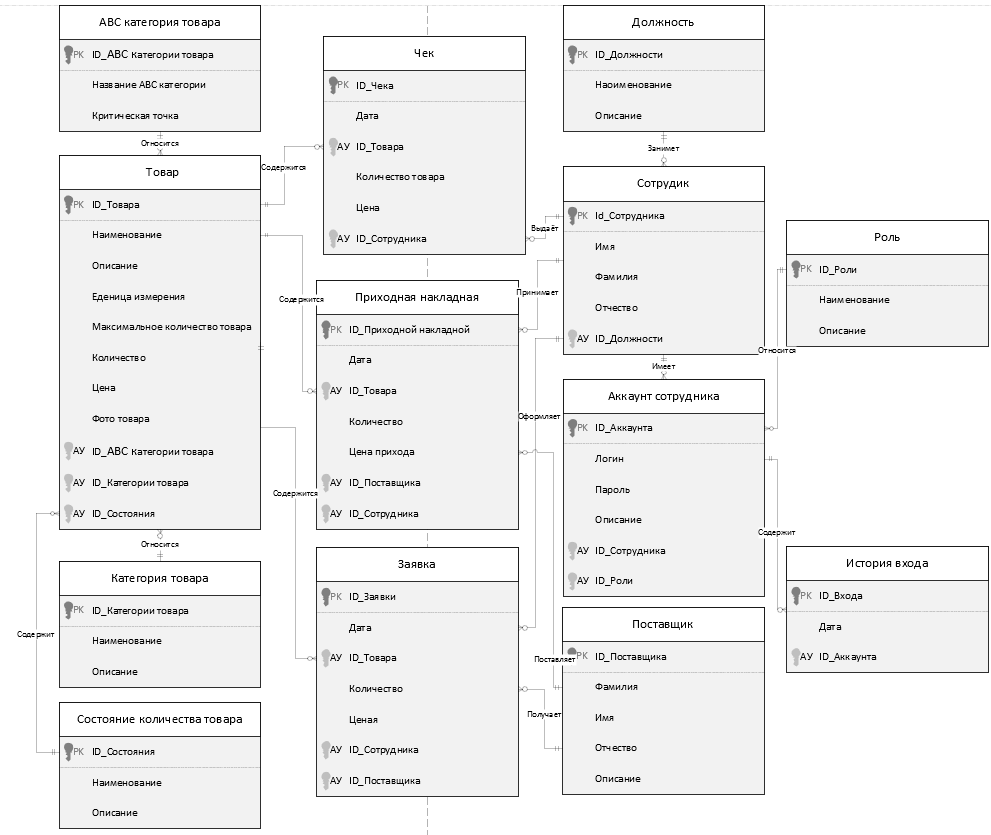


Рисунок 6 – ER-диаграмма для программной подсистемы

«Azur»

В таблице 10 представлены сведения о сущностях и связях между сущностями ER-диаграммы из рисунка 4.

Таблица 10 – Характеристики сущностей и связей между сущностями   
ER-диаграммы для программной подсистемы «Azur»

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Назначение сущности | Ключ | Характеристики связей |
| Товар | Содержит данные о товаре:   * наименование; * описание; * единица измерения; * максимально возможное количество товара на складе; * количество на данный момент; * цена; * фото товара. | ID\_Товара (PK) | Товар – Чек (1.. n)  Товар – Приходная накладная (1.. n)  Товар – Заявка (1.. n)  Товар – ABC категория товара (n..1)  Товар – Категория товара (n..1) |
| ABC категория товара | Содержит данные о категории ABC:   * наименование ABC категории; * критическая точка. | ID\_ABC категории товара (PK) | ABC категория товара – Товар (1.. n) |
| Категория товара | Содержит данные о категориях товара по предназначению:   * наименование категории; * описание категории. | ID\_Категории товара (PK) | Категория товара – Товар (1.. n) |
| Состояние количества товара | Содержит данные о состоянии количества товара:   * наименование состояния; * описание состояния. | ID\_Состояние товара (PK) | Состояние количества товара – Товар (1.. n) |
| Приходная накладная | Содержит данные о приходных накладных:   * дата оформления приходной накладной; * количество поступившего товара * цена поступившего товара. | ID\_Приходной накладной (PK) | Приходная накладная – Товар (n..1)  Приходная накладная – Сотрудник (n..1)  Приходная накладная – Поставщик (n..1) |

Продолжение таблицы 10

|  |  |  |  |
| --- | --- | --- | --- |
| Заявка | Содержит данные о заявках:   * дата оформления заявки; * количество заказанного товара; * цена заказанного товара. | ID\_Заявки (PK) | Заявка – Товар (n..1)  Заявка – Сотрудник (n..1)  Заявка – Поставщик (n..1) |
| Чек | Содержит данные о чеках:   * дата оформления чека; * количество проданного товара; * цена товара, за которую совершилась продажа. | ID\_Чека (PK) | Чек – Товар (n..1)  Чек – Сотрудник (n..1) |
| Поставщики | Содержит данные о поставщиках, которые могу поставлять товар на данный момент:   * имя поставщика; * фамилия поставщика; * отчество поставщика. | ID\_ Поставщика (PK) | Поставщик – Приходная накладная (1..n)  Поставщик – Заявка (1..n) |
| Сотрудник | Содержит данные о сотрудниках:   * имя сотрудника; * фамилия сотрудника; * отчество сотрудника. | ID\_Сотрудника (PK) | Сотрудник – Чек (1..n)  Сотрудник – Приходная накладная (1..n)  Сотрудник – Заявка (1..n)  Сотрудник – Должность (n..1)  Сотрудник – Аккаунт сотрудника (1..n) |
| Должность | Содержит данные о должностях, которые могут занимать сотрудники:   * наименование должности; * описание должности. | ID\_Должности (PK) | Должность – Сотрудник (1..n) |

Продолжение таблицы 10

|  |  |  |  |
| --- | --- | --- | --- |
| Аккаунт сотрудника | Содержит данные об действующих аккаунтов сотрудников:   * логин аккаунта; * пароль аккаунта; * описание аккаунта; | ID\_Аккаунта (PK) | Аккаунт сотрудника – Сотрудник (1..n)  Аккаунт сотрудника – Роль (n..1)  Аккаунт сотрудника – История входа (n..1) |
| Роль | Содержит данные о ролях аккаунтов в системе:   * наименование роли; * описание роли. | ID\_Роли (PK) | Роль – Сотрудник (1..n) |
| История входа | Содержит данные о истории входов сотрудников в систему:   * дата и время входа аккаунта в систему. | ID\_Входа (PK) | История входа – Аккаунт сотрудника (n..1) |

На следующем этапе была спроектирована диаграмма классов, которая описывает требования к внутренней организации проектируемого программного обеспечения на уровне интерфейсов и классов, их кооперации, атрибутов и методов классов, правил их взаимодействия.

На рисунке 7 представлена диаграмма классов для программной подсистемы

«Azur»

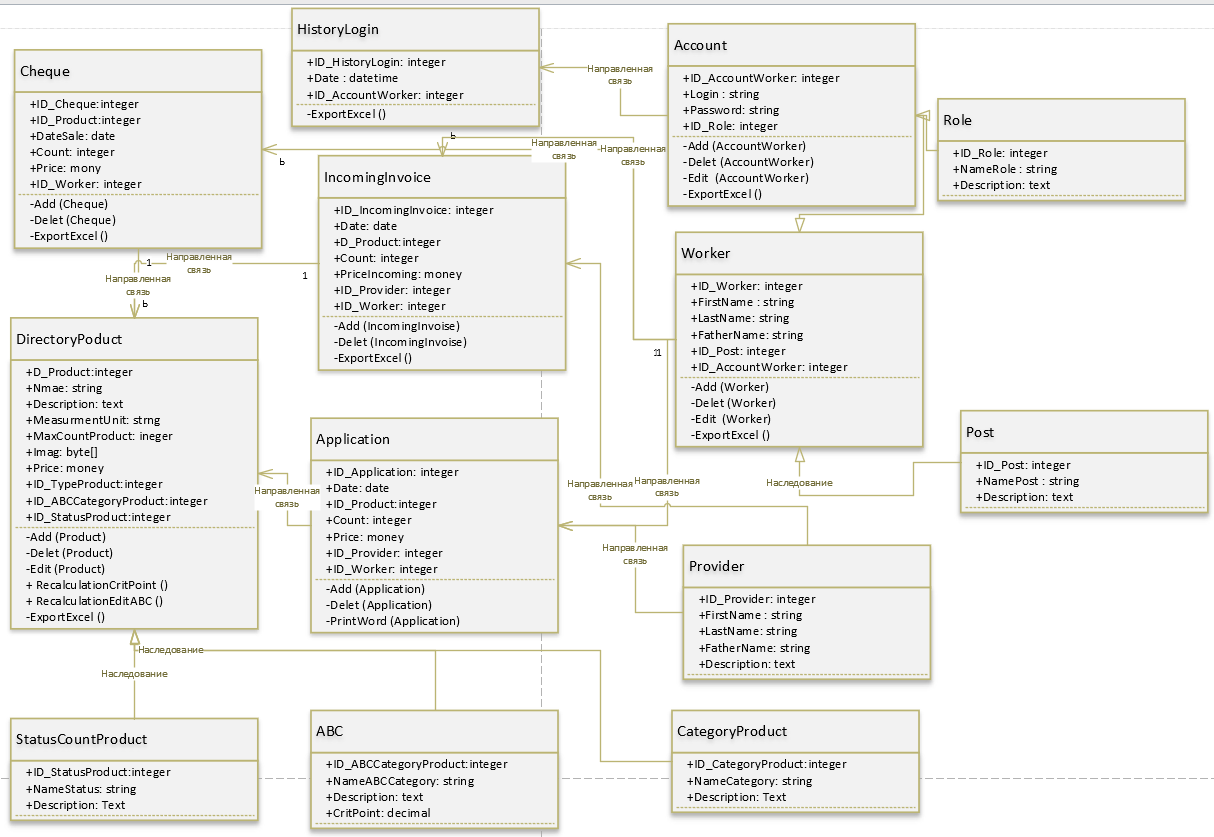


Рисунок 7 –Диаграмма классов для программной подсистемы

«Azur»

Диаграмма классов представляет собой таблицы, в которых хранится информация об их атрибутах и операций над ними.

Все атрибуты во всех классах публичные и к любым из них пользователи могут получить доступ, в соответствии своей роли.

Классы Post, Role, StatusCountProduct, ABC, CategoryProduct, Provider содержат в себе данные, которые пользователи системы самостоятельно изменить не могут, но при этом это данные предаются и используются в таблицах, к которым наследуются.

Классы Cheque, IncomingInvoice, Application содержат в себе данные, которые пользователь системы может добавлять, удалять, редактировать, экспортировать в Excel, они также связаны с другими таблицами.

Класс DirectoryPoduct содержит в себе атрибуты, описывающие товар. Пользователь системы может добавлять, удалять, редактировать уже существующие данные о продукте, экспортировать их в Excel. Операция RecalculationEditABC () является публичной и представляет из себя метод автоматического высчитывания и присваивания новой |ABC категории товару. Операция RecalculationCritPoint () выявляет соответствии текущего количества товара на складе с его критическая точка.

Класс Worker предоставляет данные о сотрудниках, которые пользователь системы может добавлять, удалять, редактировать и экспортировать данные в Excel, он также связан с другими таблицами.

Класс Account содержит атрибуты о зарегистрированных пользователях, которые имеют доступ к работе с системой, пользователь может добавлять, удалять, редактировать и экспортировать в Excel данные об аккаунтах. Классу Account присваивается роль из класса Role, которая предоставляет доступ к работе с определёнными бизнес-процессами системы.

Класс HistoryLogin содержит в себе данные о входах пользователей в систему с определённых аккаунтов, которые можно экспортировать Excel.

2.3 Конструирование прототипа

Для построения прототипа первоначально были составлены требования к макету приложения, для которого разрабатываются программные модули.

Все компоненты должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн (предпочтение отдается масштабируемой компоновке; должно присутствовать ограничение на минимальный размер окна; должна присутствовать возможность изменения размеров окна, где это необходимо; увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
* группировка элементов (в логические категории);
* использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);
* расположение и выравнивание элементов (метки, поля для ввода и т.д.);
* последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
* общая компоновка логична, понятна и проста в использовании;
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами или страницами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);
* соответствующий заголовок на каждом окне приложения.

Основные требования руководства по стилю:

* все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом, представленном на рисунке 6. Цвет, размер и пропорции логотипа не должны изменяться при использовании программы. Так же для приложения должна быть установлена иконка, соответствующая логотипу;
* тип шрифта – Comic Sans MS;
* цветовая схема предусматривает использование в качестве основного фона – белый цвет RGB (255, 255, 225), в качестве дополнительного – цвет RGB (237, 190, 187), для акцентирования внимания пользователя на целевое действие интерфейса – цвет RGB (161, 121, 119), для выделения наименований страниц цвет шрифта – RGB (111, 161, 158).



Рисунок 8 – Логотип приложения

С учетом требований к макету и руководству по стилю, для обеспечения требуемых функций, был разработан прототип программы.

На рисунке 9 представлен прототип окна программы для авторизации пользователей в системе.

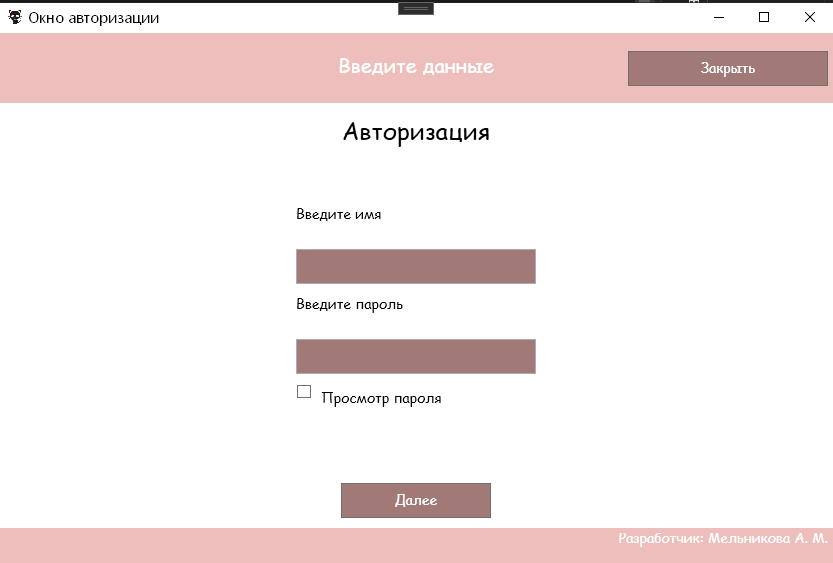


Рисунок 9 – Пример прототипа окна для реализации сценария варианта использования «Авторизация»

Пользователь должен обязательно иметь свой собственный логин и пароль, для входа, а их ввод обязателен и необходим. В зависимости от его роли в системе, предоставляется возможность работы с конкретными бизнес-процессами. Если пользователь не может зайти в систему, то ему необходимо требовать помощи у администратора.

Рассмотрим прототипы окон прецедентов для роли администратор.

После авторизации, администратор сразу может работать с вариантом использования «Формирование новых аккаунтов сотрудников». На рисунках 10 – 11 представлены прототипы страниц для этого прецедента.

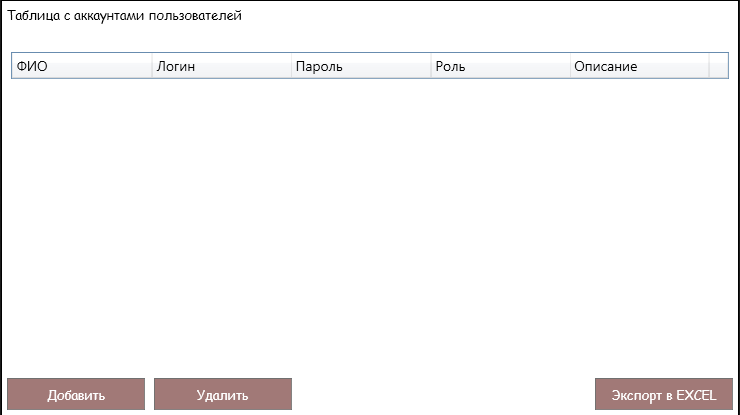


Рисунок 10 – Пример прототипа экранной формы с предоставлением данных о действующих аккаунтов сотрудников

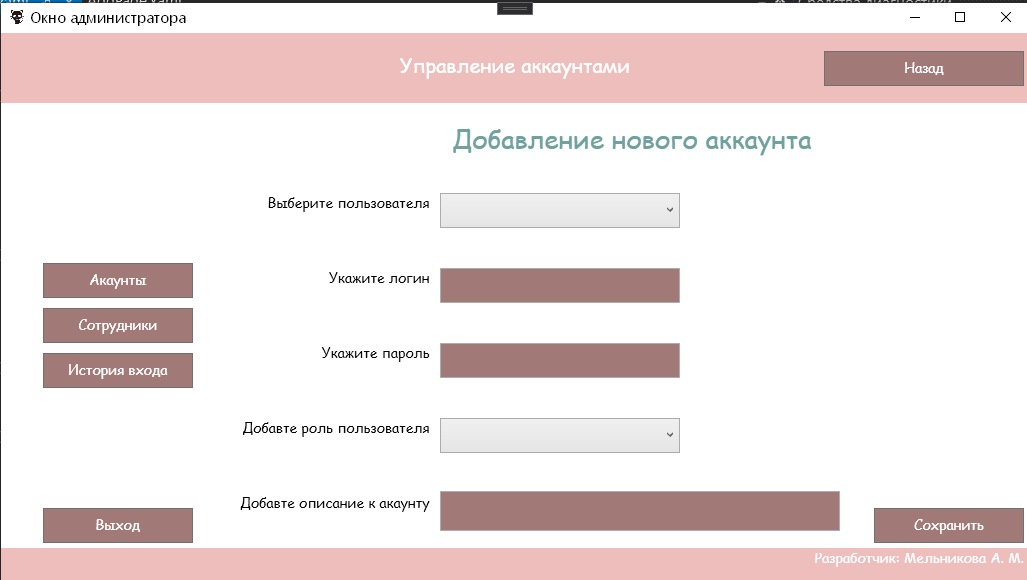


Рисунок 11 – Пример прототипа экранной формы для добавления нового аккаунта и редактирования уже существующего

Для выполнения прецедента «Регистрирование новых сотрудников» представлены прототипы страниц на рисунках 12 – 13.

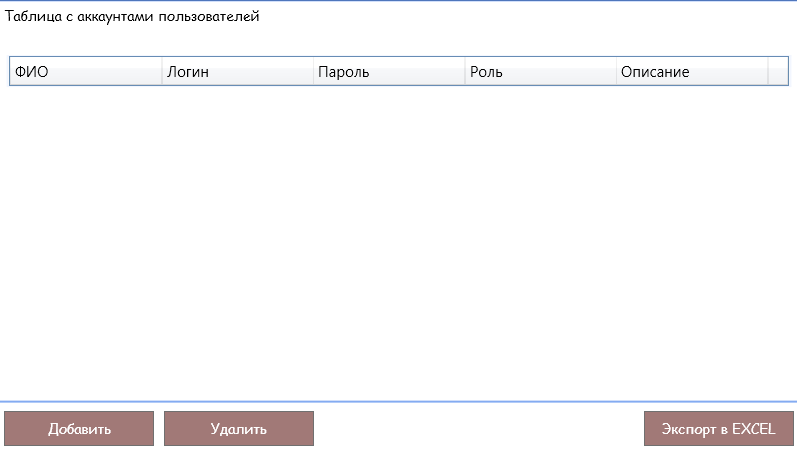


Рисунок 12 – Пример прототипа экранной формы со с писком сотрудников, работающих на предприятии

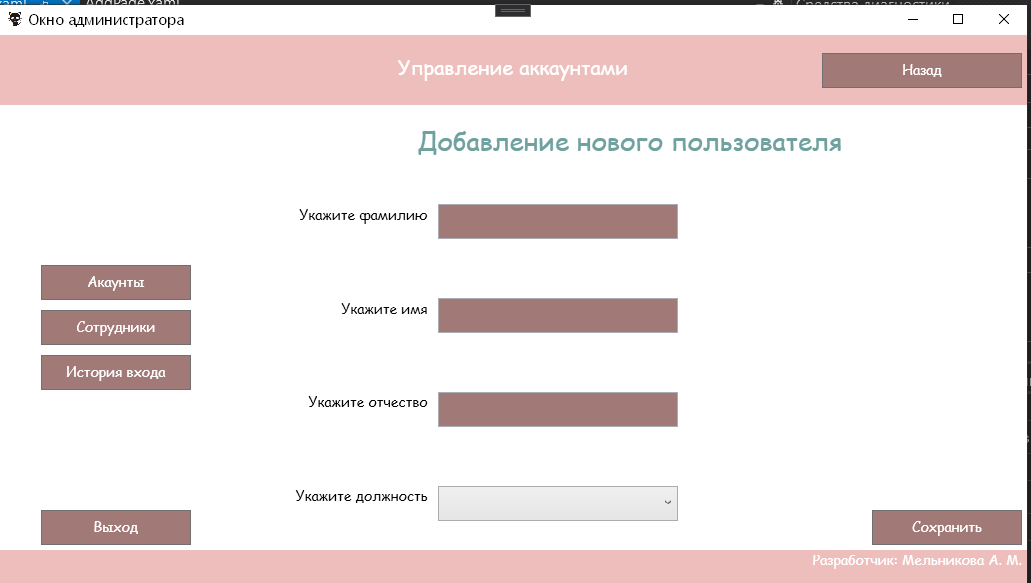


Рисунок 13 – Пример прототипа экранной формы для добавления нового сотрудника и редактирования уже существующего

Для просмотра историй входа сотрудников в систему, также реализован прототип прецедента «Просмотр историй входа», представлен на рисунке 14.

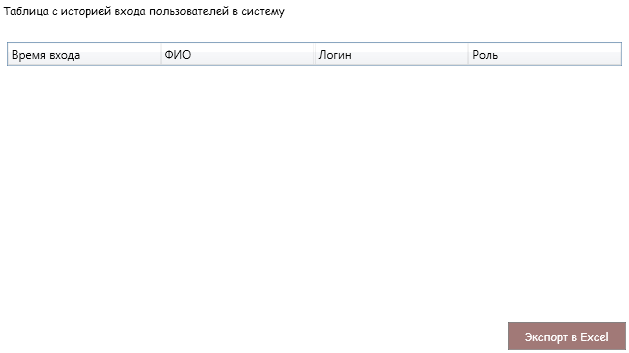


Рисунок 14 – Пример прототипа экранной формы для просмотра историй входа пользователей

Рассмотрим прототипы окон прецедентов для товароведа.

После авторизации, товаровед сразу может работать с вариантом использования «Ведение приходных накладных». На рисунках 15 – 16 представлены прототипы страниц для этого прецедента.

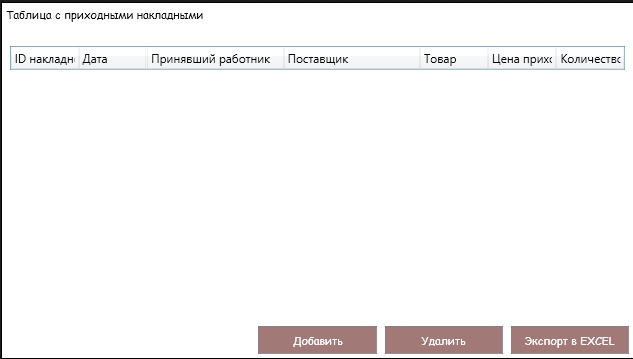


Рисунок 15 – Пример прототипа экранной формы для просмотра уже зарегистрированных в системе приходных накладных

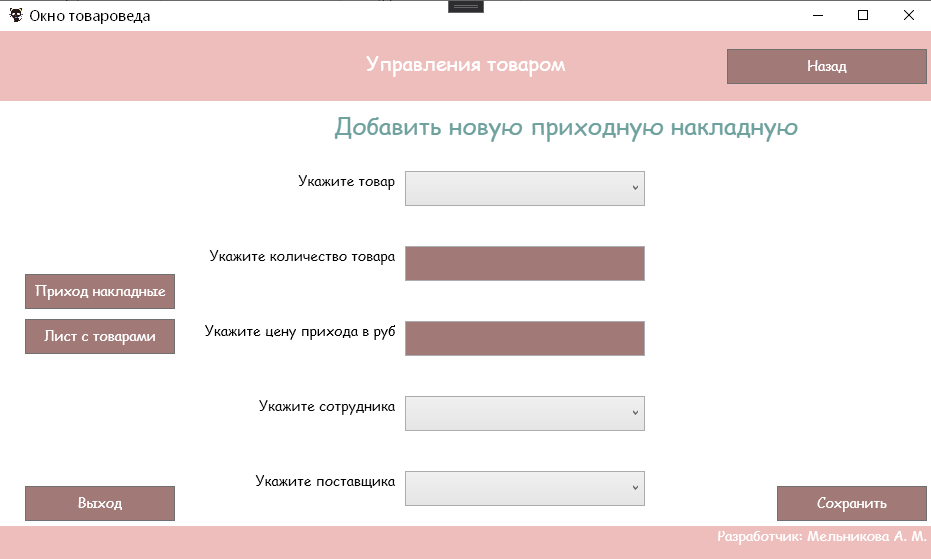


Рисунок 16 – Пример прототипа экранной формы для добавления новых приходных накладных

Для выполнения варианта использования «Ведение данных о товарах» были реализованы прототипы страниц, которые представлены на рисунках 17 – 18.

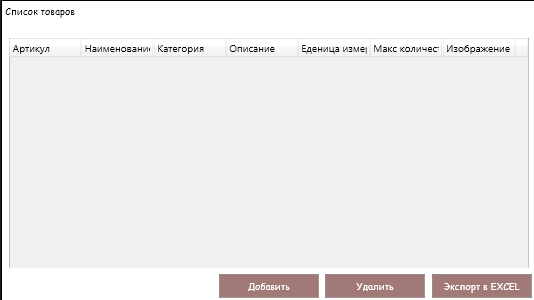


Рисунок 17 – Пример прототипа экранной формы для просмотра уже зарегистрированных товаров в системе

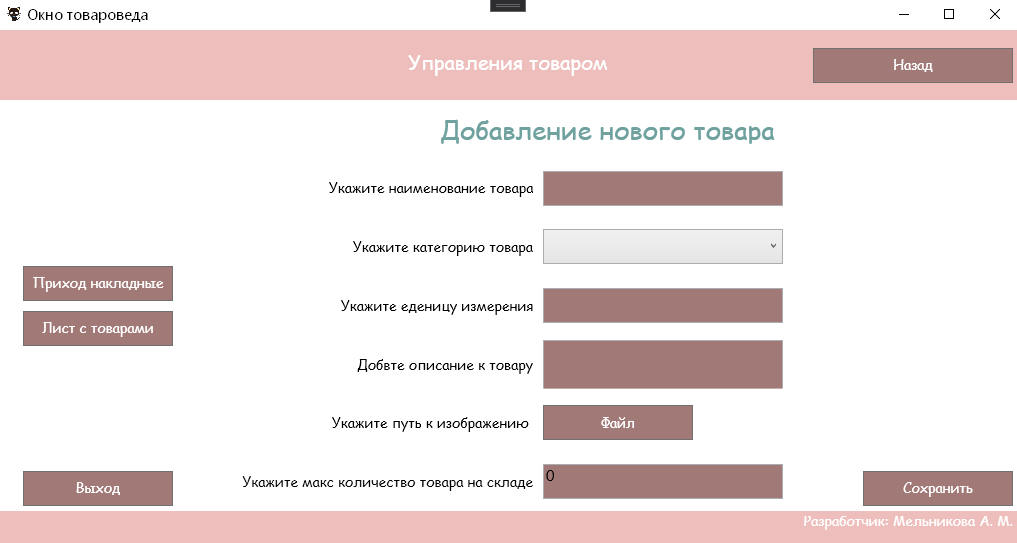


Рисунок 18 – Пример прототипа экранной формы для добавления новых товаров

Рассмотрим прототипы окон прецедентов для роли менеджер по продажам.

После авторизации, менеджер по продажам сразу может работать с вариантом использования «Работа с анализами продаж». На рисунках 19 представлены прототипы страницы для просмотра данных о товаре.

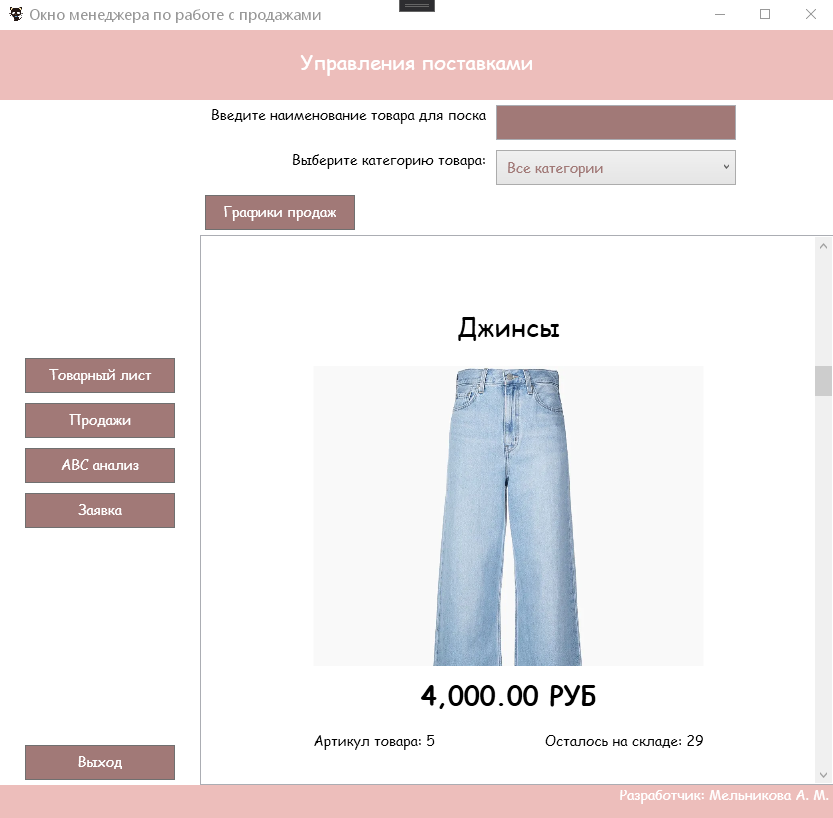


Рисунок 19 – Пример прототипа экранной формы для просмотра информации товара менеджеру по продажам

Для выполнения прецендента «Ведение данных о продажах», были реализованы прототипы, которые представлены на рисунках 20 – 21.

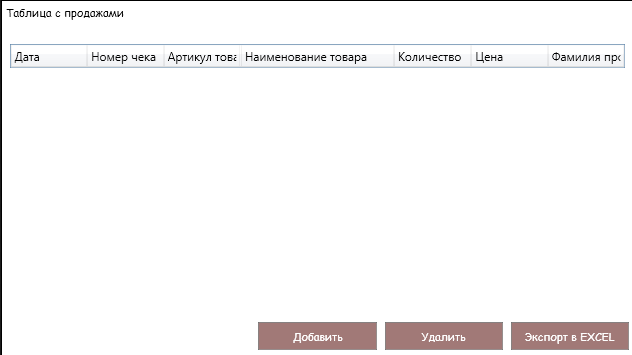


Рисунок 20 – Пример прототипа экранной формы для просмотра уже зарегистрированных в системе продаж

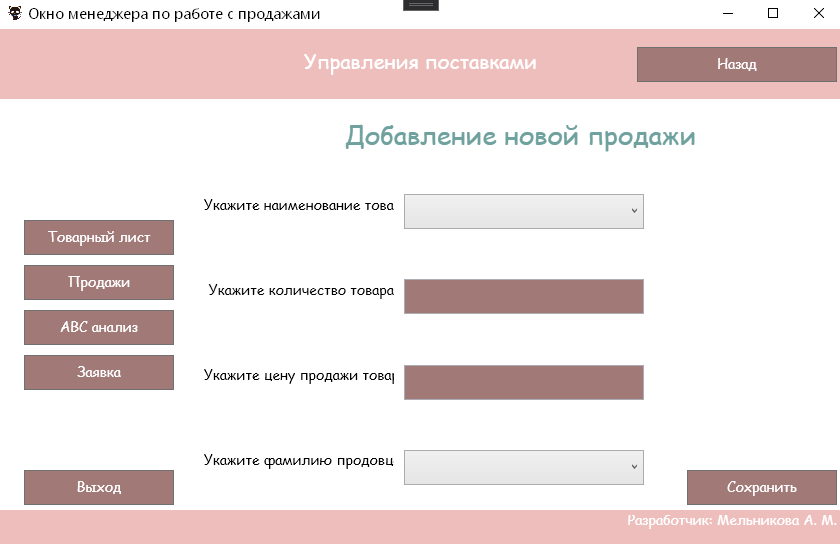


Рисунок 21 – Пример прототипа экранной формы для добавления новой продажи

На рисунках 22 – 23 представлены прототипы страниц для работы с этим прецедентом работы менеджера по продажам с вариантом использования «Работа с заявками на поставку».

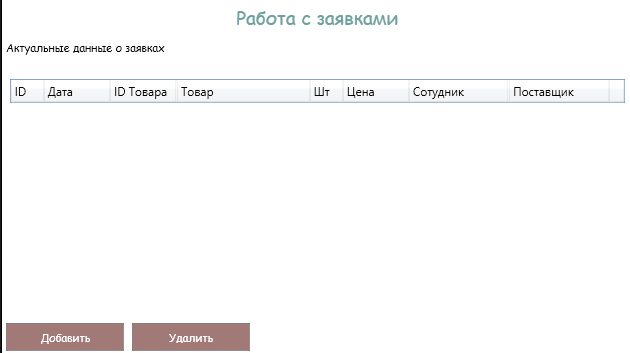


Рисунок 22 – Пример прототипа экранной формы для просмотра данных о уже зарегистрированных заявках в системе

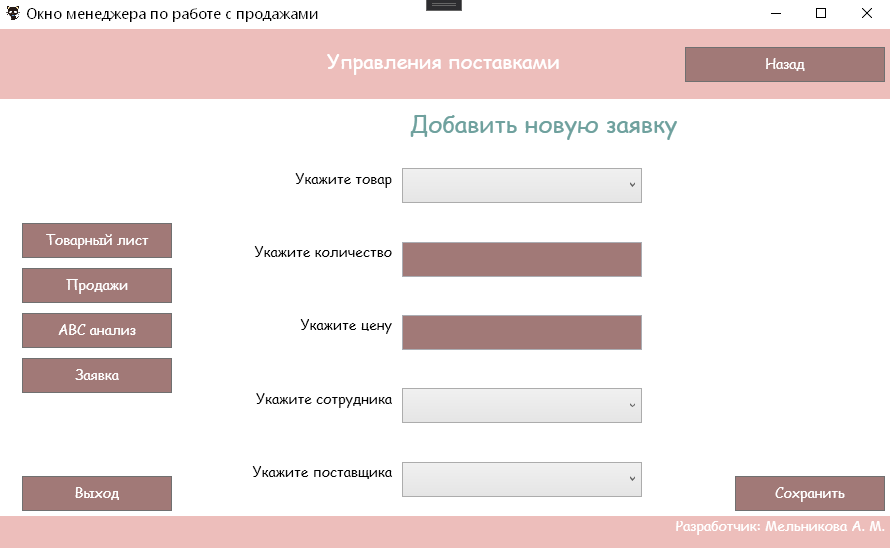


Рисунок 23 – Пример прототипа экранной формы для формирования новых заявок на поставку товара

Для того чтобы заявка сохранилась в виде документа Word, необходимо пользователю нажать на кнопку «Печать» рядом с нужной заковкой, далее появится страница с содержимым заявки, которое пользователь не может редактировать. Прототип этой страниц представлен на рисунке 24.



Рисунок 24 – Пример прототипа экранной формы для данных печатной заявки

Чтобы менеджер по продажам мог эффективно и быстро формировать новые заявки на поставку, был реализован ABC анализ продаж прототипы страниц которого представлен на рисунке 25.

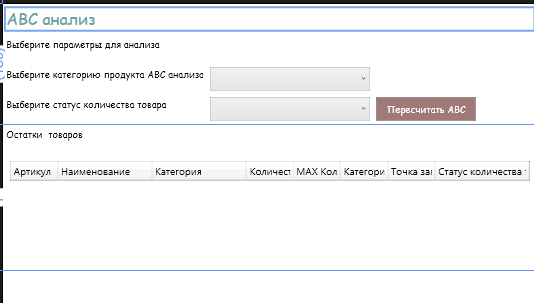


Рисунок 25 – Пример прототипа экранной формы для просмотра свойств товаров ABC анализа

Данные изображения подробно иллюстрируют примеры прототипов приложения, в разработке которого они будут использованы.

3 Реализация

3.1 Обоснование выбора средств разработки

Для разработки подсистемы анализа продаж розничного магазина одежды «Azur» была выбран интегрированная среда разработки программного обеспечения MS Visual Studio 2019 Enterprise 2019 Версия 16.9.0, в таблице 11 представлено обоснование выбора IDE.

Таблица 11 – Обоснование выбора среды программирования

|  |  |  |  |
| --- | --- | --- | --- |
| Среда разработки | Geany | Microsoft Visual Studio | MonoDevelop |
| Разработчик | Team | Microsoft | Novell и Mono community |
| Платформа | UNIX / Windows | Windows | Linux, macOS |
| Лицензия | Открытая | Закрытая | Открытая |
| Разработка GUI | Нет | Да | Да |
| Отладчик | Да | Да | Да |
| Покрытие кода | Нет | Да | Нет |

Microsoft Visual Studio из всех представленных интегрированных средств разработки подходит для написания программного обеспечения с пользовательским интерфейсом на операционную систему Windows. Также большую роль при выборе IDE сыграло наличие покрытия кода, которое показывает процент исходного кода программы, который был выполнен в процессе тестирования, что очень важно при тестировании программного обеспечения.

Программный проект «Azur» написан на платформе WPF (.NET Framework) версии 4.7.2.

Логический код приложения «Azur» полностью написан на языке программирования C#.

Для работы с текстовыми документами Word подключена дополнительная ссылка на библиотеку Microsoft Word 16.0 Object Library, версии 8.7.

Для работы с документами Excel подключена также дополнительная ссылка на библиотеку Microsoft Excel 16.0 Object Library, версии 1.9.

Выбранные средства разработки удовлетворяют требования технического задания [4].

3.2 Разработка базы данных в среде СУБД

На основе ранее спроектированной ER-диаграммы в среде СУБД Microsoft SQL Server была разработана БД MelProduct, которая содержит все необходимые таблицы для работы системы.

Был сформирован скрипт, который содержит в себе данные о структуре и таблицах БД. Скрипт имеет название script.sql.

Структура БД представлена на рисунке 26 в виде диаграммы.

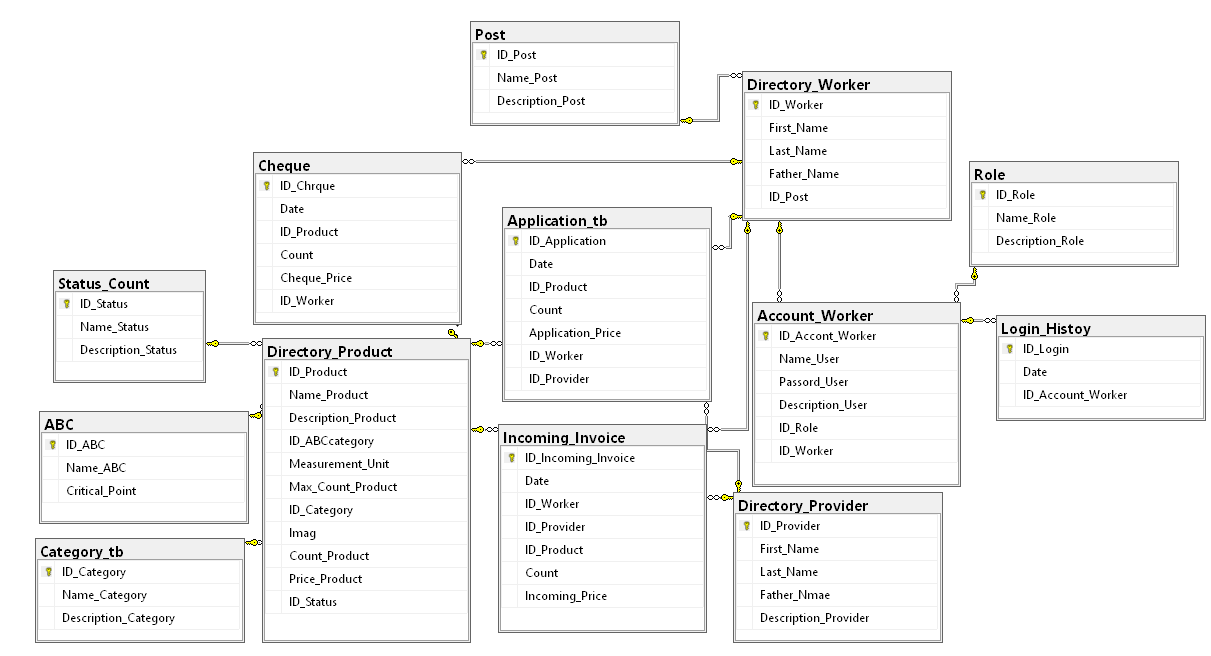


Рисунок 26 – Диаграмма, представляющая структуру БД

В таблице 11 представлено краткое описание таблиц БД.

Таблица 12 – Описание таблиц, составляющих структуру БД

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Назначение таблицы | Ключ | Характеристики связей |
| Directory\_Product | Содержит данные о товаре:   * Name\_Product – наименование; * Description\_Product – описание; * Measurement\_Unit – единица измерения; * Max\_Count\_Product – максимально возможное количество товара на складе; * Count\_Product – количество на данный момент; * Price\_Product – цена; * Imag – фото товара. | ID\_Product (PK) | Directory\_Product – Чек (1.. n)  Directory\_Product – Incoming\_Invoice (1.. n)  Directory\_Product – Application\_tb (1.. n)  Directory\_Product – ABC (n..1)  Directory\_Product – Category\_tb (n..1)  Directory\_Product – Status\_Count (n..1) |
| ABC | Содержит данные о категории ABC:   * Name\_ABC – наименование ABC категории; * Critical\_Point – критическая точка. | ID\_ABC (PK) | ABC – Directory\_Product (1.. n) | |
| Category\_tb | Содержит данные о категориях товара по предназначению:   * Name\_Category – наименование категории; * Description\_Category – описание категории. | ID\_Category (PK) | Category\_tb – Directory\_Product (1.. n) | |
| Status\_Count | Содержит данные о состоянии количества товара:   * Name\_Status – наименование состояния; * Description\_Status – описание состояния. | ID\_Status (PK) | ID\_Status – Directory\_Product (1.. n) | |

Продолжение таблицы 12

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Назначение таблицы | Ключ | Характеристики связей |
| Incoming\_Invoice | Содержит данные о приходных накладных:   * Date – дата оформления приходной накладной; * Count – количество поступившего товара * Incoming\_Price – цена поступившего товара. | ID\_Incoming\_Invoice (PK) | Incoming\_Invoice – Directory\_Product (n..1)  Incoming\_Invoice – Directory\_Worker (n..1)  Incoming\_Invoice – Directory\_Provider (n..1) |
| Application\_tb | Содержит данные о заявках:   * Date – дата оформления заявки; * Count – количество заказанного товара; * Application\_Price – цена заказанного товара. | ID\_Application (PK) | Application\_tb – Directory\_Product (n..1)  Application\_tb – Directory\_Worker (n..1)  Application\_tb – Directory\_Provider (n..1) |
| Cheque | Содержит данные о чеках:   * Date – дата оформления чека; * Count – количество проданного товара; * Cheque\_Price – цена товара, за которую совершилась продажа. | ID\_Chrque (PK) | Cheque – Directory\_Product (n..1)  Cheque – Directory\_Worker (n..1) |
| Directory\_Provider | Содержит данные о поставщиках, которые могу поставлять товар на данный момент:   * First\_Name – имя поставщика; * Last\_Name – фамилия поставщика; * Father\_Nmae – отчество поставщика; * Description\_Provider – описание поставщика. | ID\_Provider (PK) | Directory\_Provider – Incoming\_Invoice (1..n)  Directory\_Provider – Application\_tb (1..n) |

Продолжение таблицы 12

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Назначение таблицы | Ключ | Характеристики связей |
| Directory\_Worker | Содержит данные о сотрудниках:   * First\_Name – имя сотрудника; * Last\_Name – фамилия сотрудника; * Father\_Nmae – отчество сотрудника. | ID\_Worker (PK) | Directory\_Worker– Чек (1..n)  Directory\_Worker– Incoming\_Invoice (1..n)  Directory\_Worker– Application\_tb (1..n)  Directory\_Worker– Post (n..1)  Directory\_Worker– Account\_Worker (1..n) |
| Post | Содержит данные о должностях, которые могут занимать сотрудники:   * Name\_Post – наименование должности; * Description\_Post – описание должности. | ID\_Post (PK) | Post– Directory\_Worker (1..n) |
| Account\_Worker | Содержит данные об действующих аккаунтов сотрудников:   * Name\_User – логин аккаунта; * Passord\_User – пароль аккаунта; * Description\_User – описание аккаунта; | ID\_Accont\_Worker (PK) | Account\_Worker – Directory\_Worker (1..n)  Account\_Worker – Role (n..1)  Account\_Worker – Login\_Histoy (n..1) |
| Role | Содержит данные о ролях аккаунтов в системе:   * Name\_Role – наименование роли; * Description\_Role – описание роли. | ID\_Role (PK) | Role – Account\_Worker (1..n) |
| Login\_Histoy | Содержит данные о истории входов сотрудников в систему:   * Date – дата и время входа аккаунта в систему. | ID\_Login (PK) | Login\_Histoy – Account\_Worker (n..1) |

Таблицы в базе данных приведены к третьей нормальной форме.

3.3 Описание программных модулей

Структура программной подсистемы розничного магазина одежды «Azure» представлена на рисунке 27.

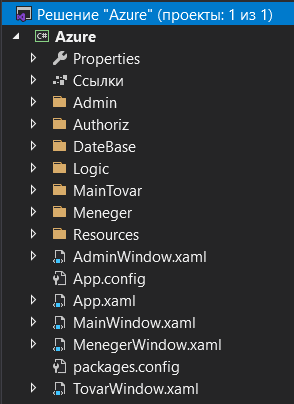


Рисунок 27 – Структура проекта приложения анализа продаж

Далее будут представлены структуры папок проекта с описанием их компонентов.

В папке Admin хранятся страницы, которые предназначены для пользователя с ролью администратора.

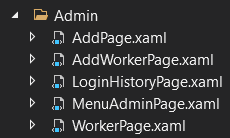


Рисунок 28 – Структура папки Admin проекта приложения анализа продаж

Таблица 12 – Описание назначений всех модулей из папки Admin

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Назначение | Описание |
| AddPage.xaml | Добавление нового аккаунта для уже зарегистрированного в системе пользователю. | Администратор заполняет форму корректными данными, после их сохранения добавляется в систему новый аккаунт. |
| AddWorkerPage.xaml | Добавление нового сотрудника в систему, для дальнейшей с ним работы. | Администратор заполняет форму корректными данными, после их сохранения в систему добавляется новый сотрудник. |
| MenuAdminPage.xaml | Отображение аккаунтов пользователей из бд | Можно экспортировать таблицу с аккаунтами пользователей в excel. |
| WorkerPage.xaml | Отображение сотрудников из бд | Можно экспортировать таблицу с сотрудниками в excel. |

В папке Authoriz хранятся страница и папка с вспомогательным классом, которые позволяют зарегистрированному пользователю авторизоваться в системе, модули представлены на рисунке 29.

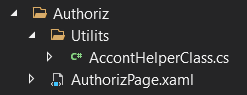


Рисунок 29 – Структура папки Authoriz проекта приложения анализа продаж

Таблица 13 – Описание назначений всех модулей из папки Admin

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Назначение | Описание |
| AccontHelperClass.cs | Содержит метод, который, вызывается при авторизации. | Содержит метод, который принимает значения о текущем активном аккаунте |

Продолжение таблицы 13

|  |  |  |
| --- | --- | --- |
| AuthorizPage.xaml | Авторизация пользователей в системе | Страница, которая в дальнейшем перенаправляет пользователей, согласно роли. |

В папке DateBase хранятся классы и модель базы данных, представленные на рисунке 30.

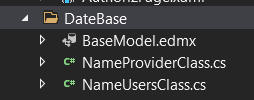


Рисунок 30 – Структура папки DateBase проекта приложения анализа продаж

Таблица 14 – Описание назначений всех модулей из папки DateBase

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Назначение | Описание |
| BaseModel.edmx | Модель бд для хранения и обработки информации. | Хранит в себе все данные о таблицах и их структурах. |
| NameProviderClass.cs | Группирует данные о фамилии, имени и отчества поставщика и полей таблиц в переменную | Один метод, выполняющий группировку данных. |
| NameUsersClass.cs | Группирует данные о фамилии, имени и отчества сотрудника и полей таблиц в переменную | Один метод, выполняющий группировку данных. |

В папке Logic хранятся две папки с классами, которые выполняют основную логику ABC анализа и управляют точками запаса, компоненты представлены на рисунке 31.

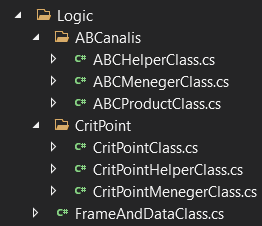


Рисунок 31 – Структура папки Logic проекта приложения анализа продаж

Таблица 15 – Описание назначений всех модулей из папки Logic

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Назначение | Описание |
| ABCHelperClass.cs | Высчитывает определённые значения для дальнейшей быстрой их обработки | Высчитывает общую стоимости, общее количества и выполняет проверку на наличие приходных накладных. |
| ABCMenegerClass.cs | Выполняется основная логика ABC анализа. | Присваивание товарам новых данных о их ABC категории. |
| ABCProductClass.cs | Присваивание методам и переменным значения из таблиц и других методов. | Данные в дальнейшем используются при расчёте ABC анализа. |
| CritPointClass.cs | Присваивание методам и переменным значения из таблиц и других методов. | Данные в дальнейшем используются при расчёте критических точек запаса. |
| CritPointHelperClass.cs | Отдельный класс, который помогает высчитывать дополнительные данные. | Содержит только проверку наличия ABC категории у товара. |
| CritPointMenegerClass.cs | Проверка количество товара на складе с их точкой запасов. | Точка запасов высчитывается по определённой категории товара. |
| FrameAndDataClass.cs | Хранит все значения окон и фреймов | Фреймы и окна представлены в виде публичных статичных методов. |

В папке MainTovar хранятся страницы, которые позволяют пользователю с ролью товароведа выполнять прецеденты: «Ведение данных о товарах», «Ведение приходных накладных», «Просмотр данных о товарах», которые представлены на рисунке 32.

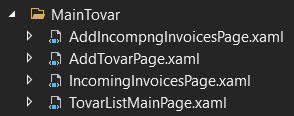


Рисунок 32 – Структура папки MainTovar проекта приложения анализа продаж

Таблица 16 – Описание назначений всех модулей из папки MainTovar

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Назначение | Описание |
| AddIncompngInvoicesPage.xaml | Добавление новой приходной накладной | Пользователь может добавлять новую приходную накладную |
| AddTovarPage.xaml | Добавление новых данных о товарах. | Пользователь может добавлять и редактировать данные о товарах. |
| IncomingInvoicesPage.xaml | Вывод данных о уже существующих приходных накладных | Пользователь может просматривать, удалять и экспортировать данные в Excel. |
| TovarListMainPage.xaml | Вывод данных о уже существующих товаров | Пользователь может просматривать, удалять и экспортировать данные в Excel. |

В папке Meneger хранятся страницы, которые позволяют пользователю с ролью менеджера продаж выполнять прецеденты: «Просмотр данных о товарах», «Просмотр данных анализа», «Работа с заявками на поставку», «Введение данных о продажах» которые представлены на рисунке 32.

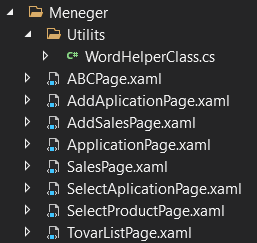


Рисунок 33 – Структура папки Meneger проекта приложения анализа продаж

Таблица 17 – Описание назначений всех модулей из папки Meneger

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Назначение | Описание |
| WordHelperClass.cs | Класс, помогающий составить заявку в виде документа Word. | Заявка сохраняется на компьютере пользователя. |
| ABCPage.xaml | Отображение таблицы с результатами ABC анализа. | Содержит в себе фильтры для поиска информации. |
| AddAplicationPage.xaml | Добавление новой заявки. | Заявка сохраняется в бд. |
| AddSalesPage.xaml | Добавление новой продажи. | Продажа сохраняется в бд. |
| ApplicationPage.xaml | Отображение данных уже существующих заявках. | Данные выводятся в виде таблицы. |
| SalesPage.xaml | Отображение данных уже существующих продажах. | Данные выводятся в виде таблицы. |
| SelectAplicationPage.xaml | Выводит данные, которые в дальнейшем будут использованы в Word документе заявки. | Данные выводятся в виде уже заполненной формы, которую пользователь редактировать не может. |
| SelectProductPage.xaml | Предоставляет информацию о товаре и его продажах. | Графики о продажах товара пользователь может выводить на печать. |
| TovarListPage.xaml | Просмотр дынных о существующих товаров. | Содержит в себе фильтры для поиска информации. |

В папке Resources хранятся необходимые для работы дополнительные данные, которые представлены на рисунке 34.

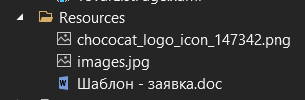


Рисунок 34 – Структура папки Resources проекта приложения

Таблица 18 – Описание назначений всех модулей из папки Resources

|  |  |
| --- | --- |
| Наименование модуля | Назначение |
| chococat\_logo\_icon\_147342.png | Изображение логотипа предприятия. |
| images.jpg | Загрузка картинки товара, при её отсутствии в бд. |
| Шаблон – заявка.doc | Шаблон для дальнейшего программного заполнения заявки. |

Далее будут представлены скриншоты экранных форм страниц тексты основных обработчиков представлены в Приложении Б.

На рисунке 35 представлено окно авторизации пользователей в системе. Код разметки страницы «Авторизация» представлен на рисунке Б.1. Код обработчика для кнопки «Далее» представлен на рисунке Б.2. После выполнения обработчика осуществляется переход в следующие окно.

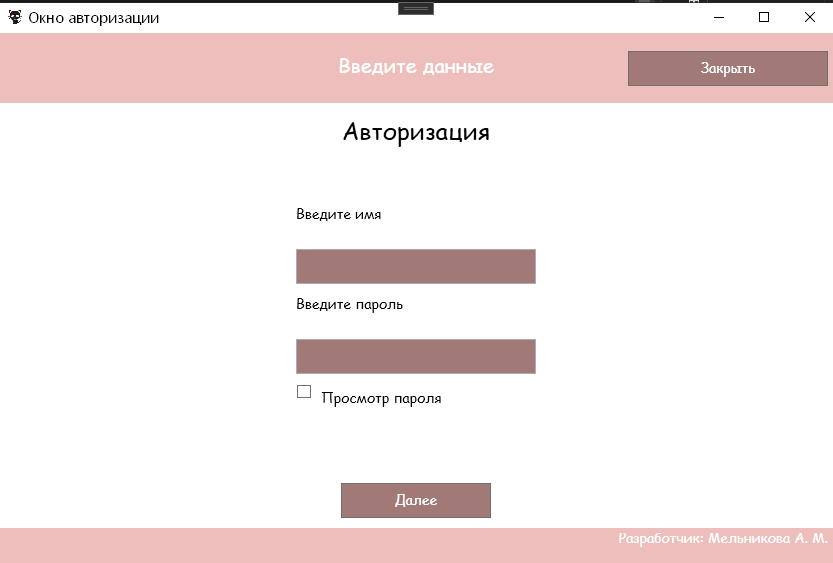


Рисунок 35 – Экранная форма окна для реализации сценария варианта использования «Авторизация»

Рассмотрим экранные формы окон для роли администратора.

После авторизации, администратор сразу может работать с аккаунтами пользователей. Код разметки страницы «Действующие аккаунты сотрудников» представлен на рисунке Б.3, экранная форма страницы представлена на рисунке 36, с уже заполненной таблицей предварительных данных.



Рисунок 36 – Экранная форма страницы с предоставлением данных о действующих аккаунтов сотрудников

Кнопка «Добавить» имеет обработчик, при выполнении которого осуществляется переход на страницу «Добавление нового аккаунта». Код обработчика кнопки представлен на рисунке Б.4. Код экранной формы представлен на рисунке Б.5, сама экранная форма представлена на рисунке 37.

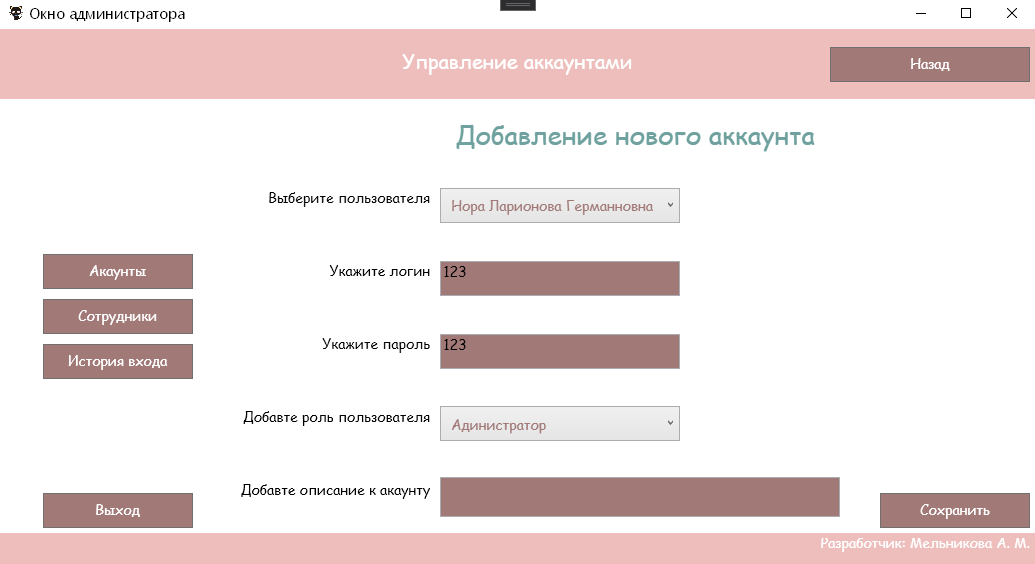


Рисунок 37 – Экранная форма страницы для добавления нового аккаунта и редактирования уже существующего

Экранная форма заполнена данными о новом аккаунте. Нажав на кнопку «Сохранить» Новый аккаунт добавится в систему. Обработчик представлен на рисунке Б.6, результат выполнения обработчика представлен на рисунке 38.



Рисунок 38 – Результат выполнения добавления нового аккаунта в систему

Нажав на кнопку «Редактирование» произойдут те же действий, что при добавлении нового аккаунта, только данные в форму «Добавление нового аккаунта» будут заполнены выбранным нами аккаунтом. Обработчик кнопки «Редактировать» представлен на рисунке Б.7.

Чтобы воспользоваться кнопкой «Удалить» сначала необходимо выбрать элемент таблицы, затем нажать на копку. Код её обработчика представлен на рисунке Б.8. На рисунках 39 – 40 представлен процесс удаления записей.

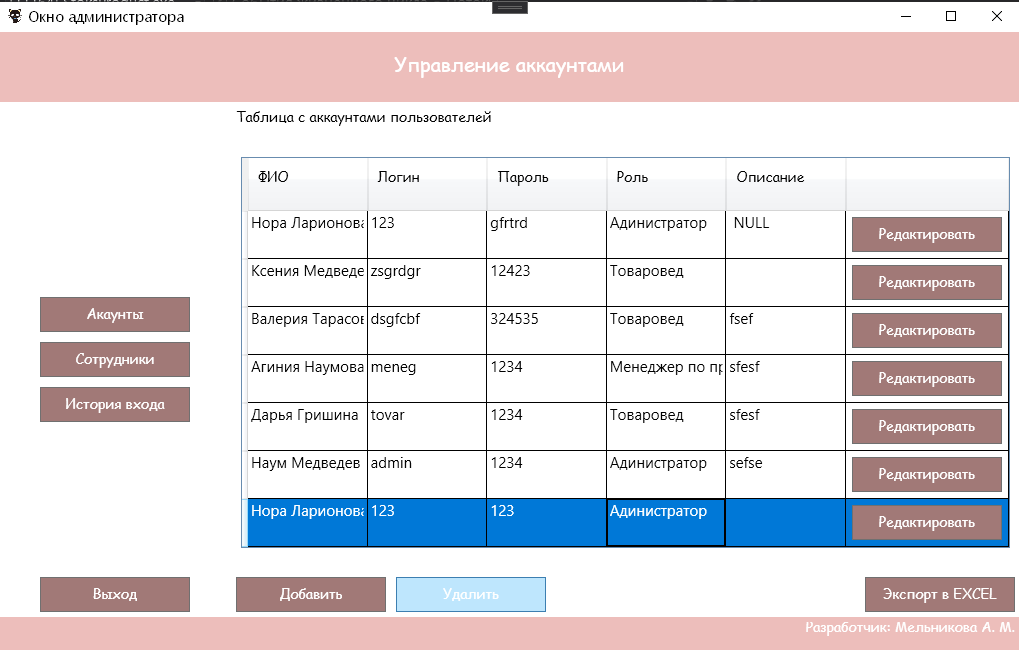


Рисунок 39 – Выбор записи, которую необходимо удалить



Рисунок 40 – Результат выполнения удаления записи.

Кнопка «Экспорт в Excel» выводит текущие данные, указанные в таблице формы в таблицу Excel. На рисунке Б.9 представлен обработчик кнопки, а на рисунке 41 результат работы этого обработчика.

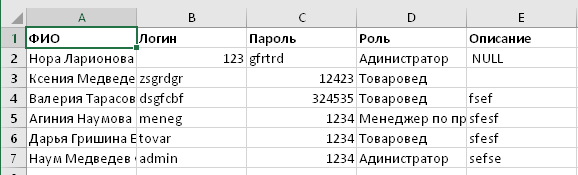


Рисунок 41 – Результат выполнения экспорта таблицы в Excel.

Для выполнения регистрации новых сотрудников пользователь работает со страницей, код которой представлен на рисунке Б.10, экранная форма на рисунке 42.



Рисунок 42 – Экранная форма страницы со списком сотрудников

Кнопка «Добавить» имеет обработчик, при выполнении которого осуществляется переход на страницу «Добавление нового сотрудника». Код экранной формы добавления сотрудника представлен на рисунке Б.11, сама экранная форма представлена на рисунке 43.

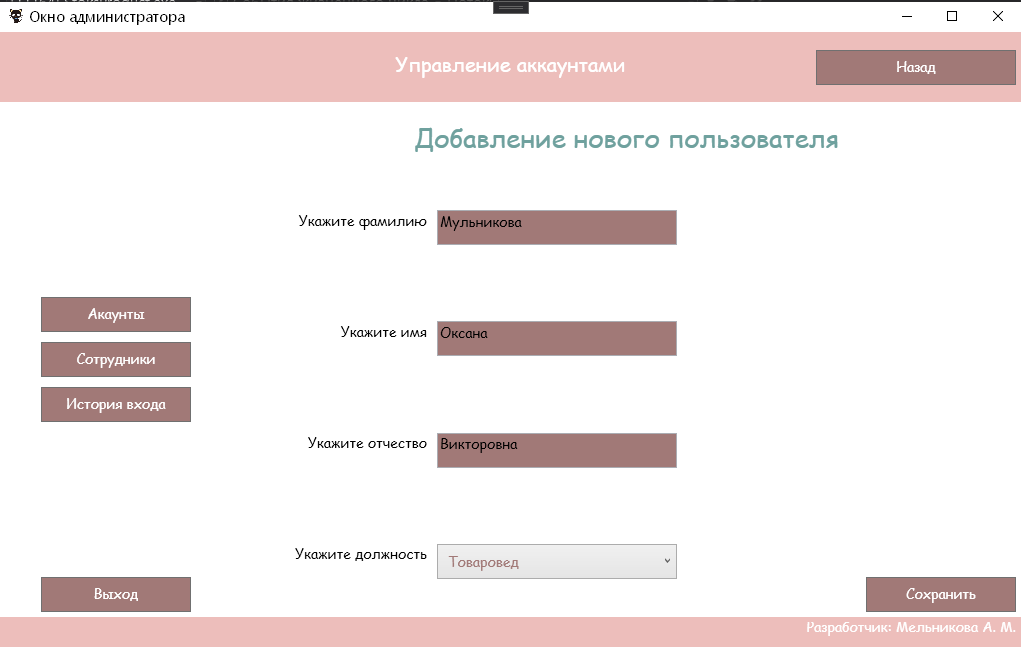


Рисунок 43 – Экранная форма страницы для добавления нового сотрудника и редактирования уже существующего

Экранная форма заполнена данными о новом пользователе системы. Нажав на кнопку «Сохранить», новый пользователь добавится в систему. Обработчик представлен на рисунке Б.12, результат выполнения обработчика представлен на рисунке 44.

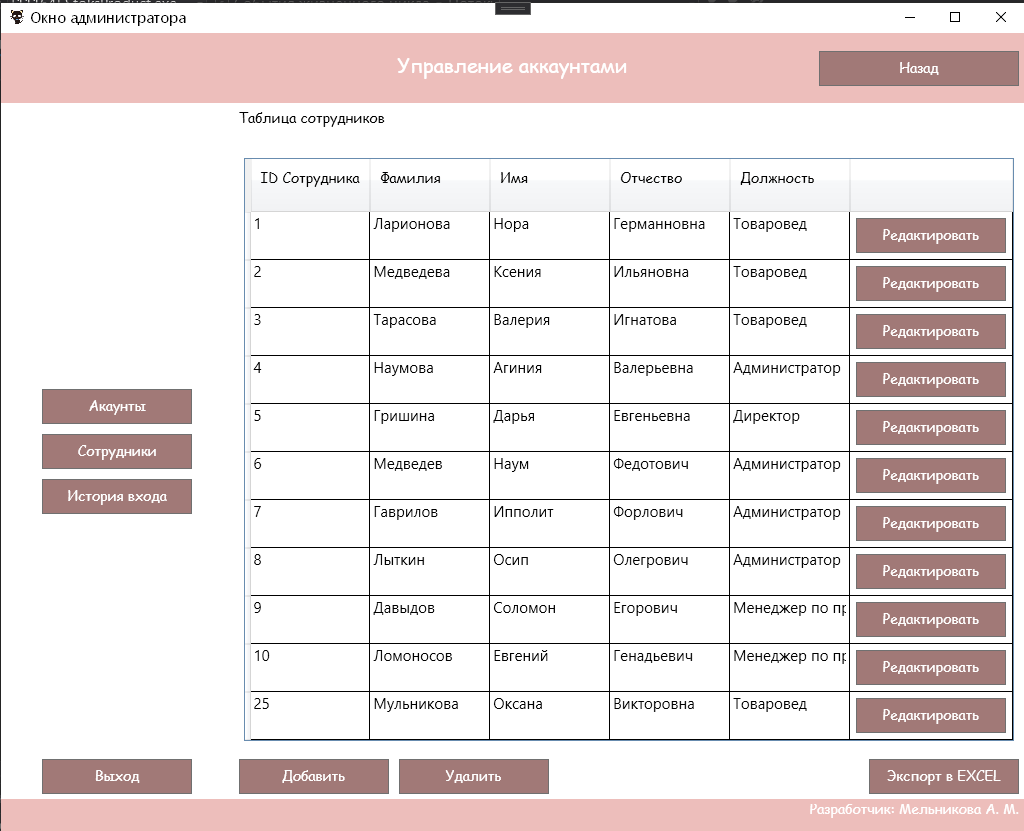


Рисунок 44 – Результат выполнения добавления нового аккаунта в систему

Нажав на кнопку «Редактировать», произойдут те же действий, которые представлены на рисунках 42 и 43, только данные в форму «Добавление нового сотрудника» будут заполнены выбранным нами аккаунтом. Обработчик кнопки «Редактировать» представлен на рисунке Б.13.

Чтобы воспользоваться кнопкой «Удалить» сначала необходимо выбрать элемент таблицы, затем нажать на копку. Код её обработчика представлен на рисунке Б.14. На рисунках 45 - 46 представлен процесс удаления записей.

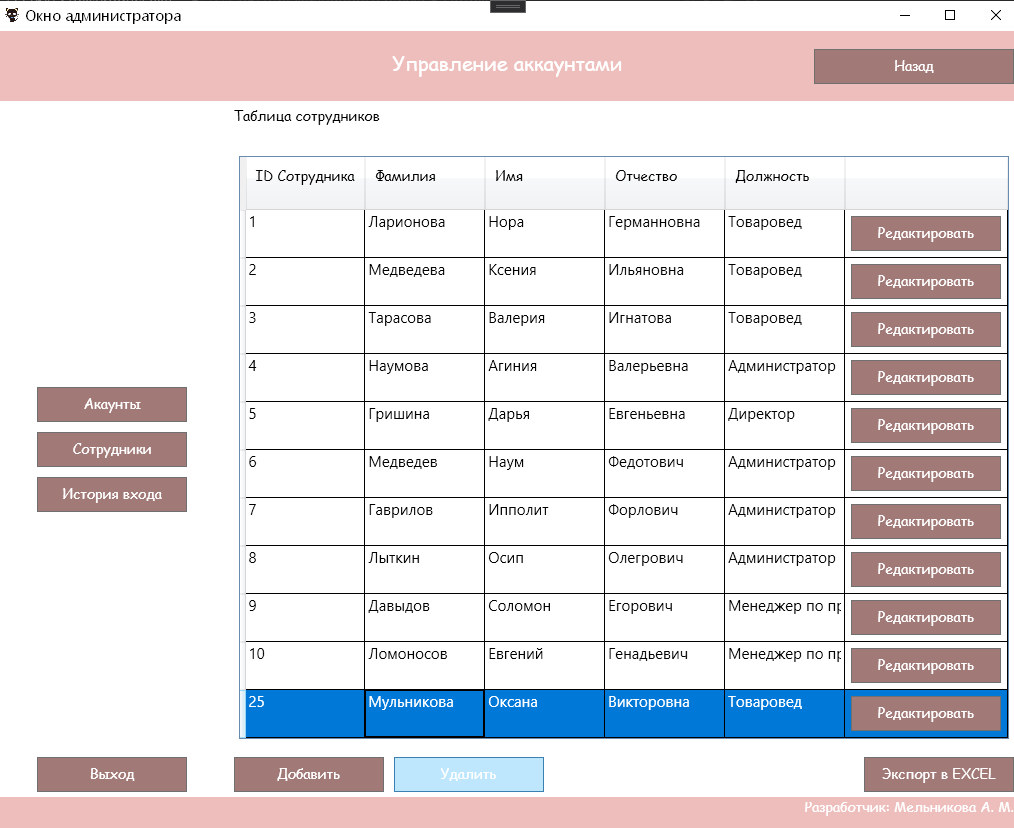


Рисунок 45 – Выбор записи, которую необходимо удалить

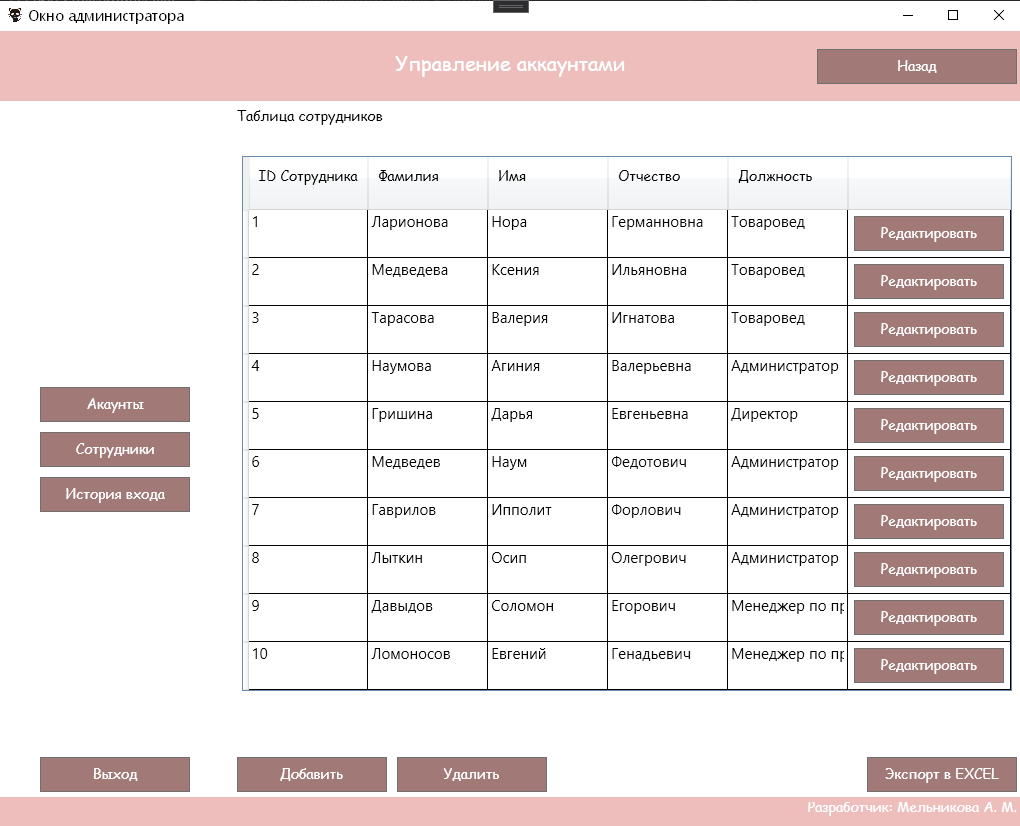


Рисунок 46 – Результат выполнения удаления записи

Кнопка «Экспорт в Excel» выводит текущие данные, указанные в таблице формы в таблицу Excel. На рисунке Б.15 представлен обработчик кнопки, а на рисунке 47 результат работы этого обработчика.

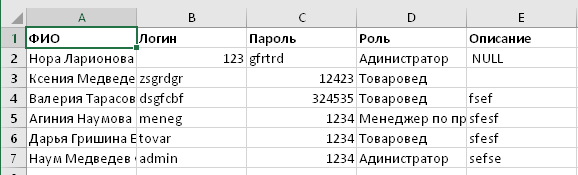


Рисунок 47 – Результат выполнения экспорта таблицы в Excel.

Для просмотра историй входа сотрудников в систему используется страница «Просмотр историй входа», представленная на рисунке 48. Программ Код экранной формы представлен на рисунке Б.16.

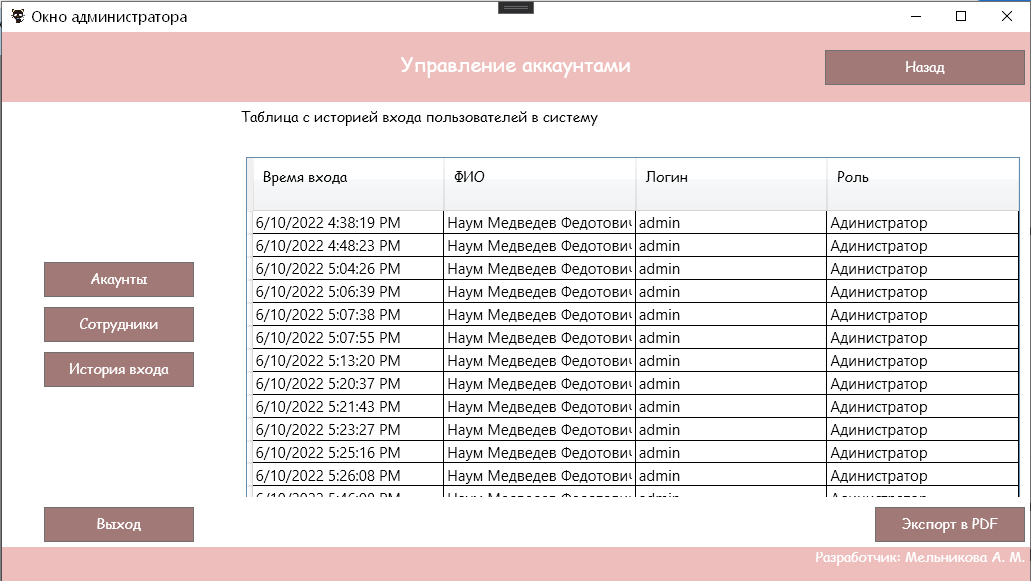


Рисунок 48 – Экранная форма страницы для просмотра историй входа сотрудников

Кнопка «Экспорт в Excel» выводит текущие данные, указанные в таблице формы в таблицу Excel. На рисунке Б.17 представлен обработчик кнопки, а на рисунке 49 результат работы этого обработчика.

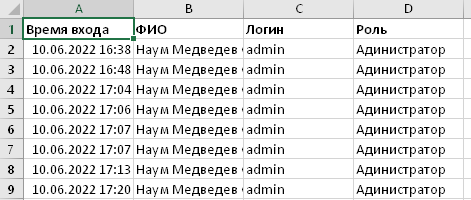


Рисунок 49 – Результат выполнения экспорта таблицы в Excel.

Рассмотрим экранные окна для выполнения прецедентов товароведа.

После авторизации, товаровед сразу может работать с «Ведение приходных накладных». На рисунках 50 – 51 представлены страниц для этого прецедента.

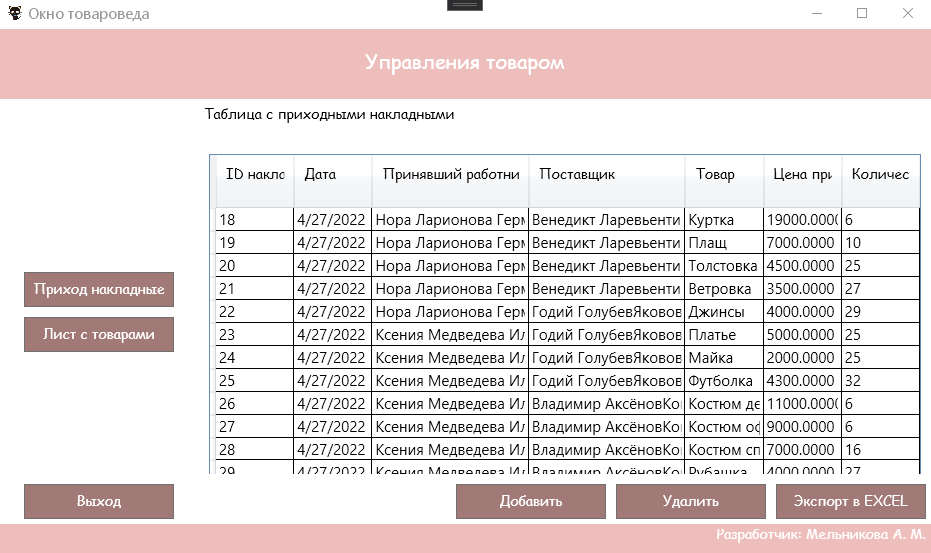


Рисунок 50 – Экранная форма страницы для просмотра уже зарегистрированных в системе приходных накладных

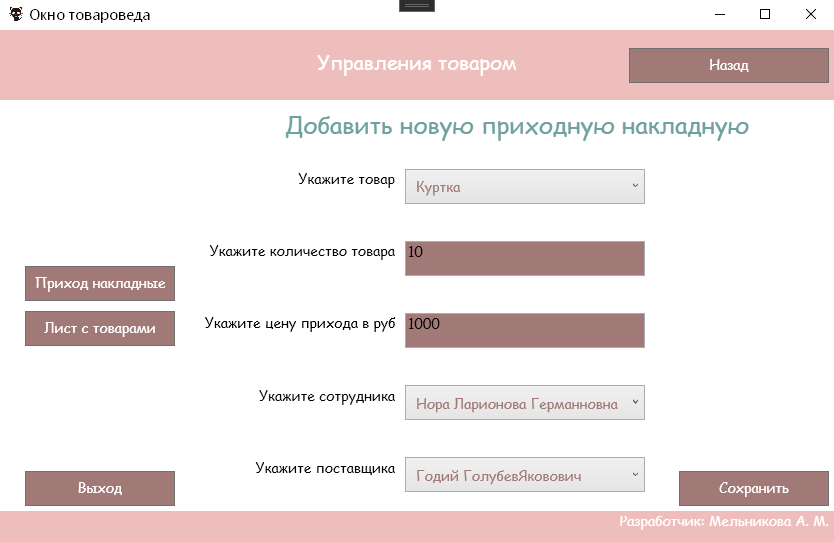


Рисунок 51 – Экранная форма страницы для добавления новых приходных накладных

После нажатия на кнопку «Сохранить», данные, введённые в поля, сохраняются в базу данных и текущее количество товара на складе увеличивается. Обработчик кнопки представлен на рисунке Б.18, результат выполнения обработчика представлен на рисунке 51.

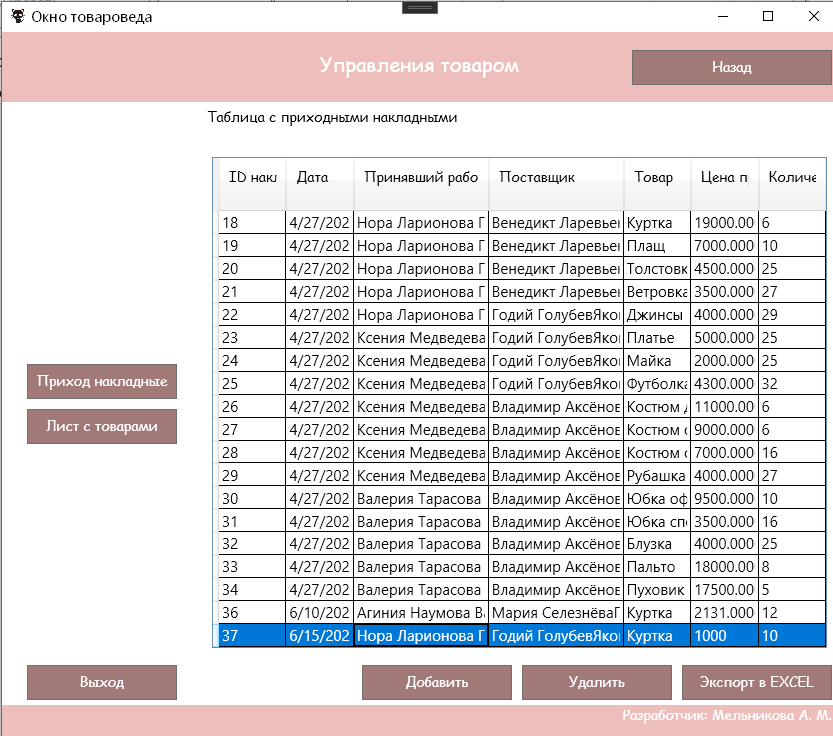


Рисунок 51 – Результат добавления новой приходной накладной

Для выполнения варианта использования «Ведение данных о товарах» были реализованы страниц, которые представлены на рисунках 52 – 53.



Рисунок 52 – Экранная форма страницы для просмотра уже зарегистрированных товаров в системе

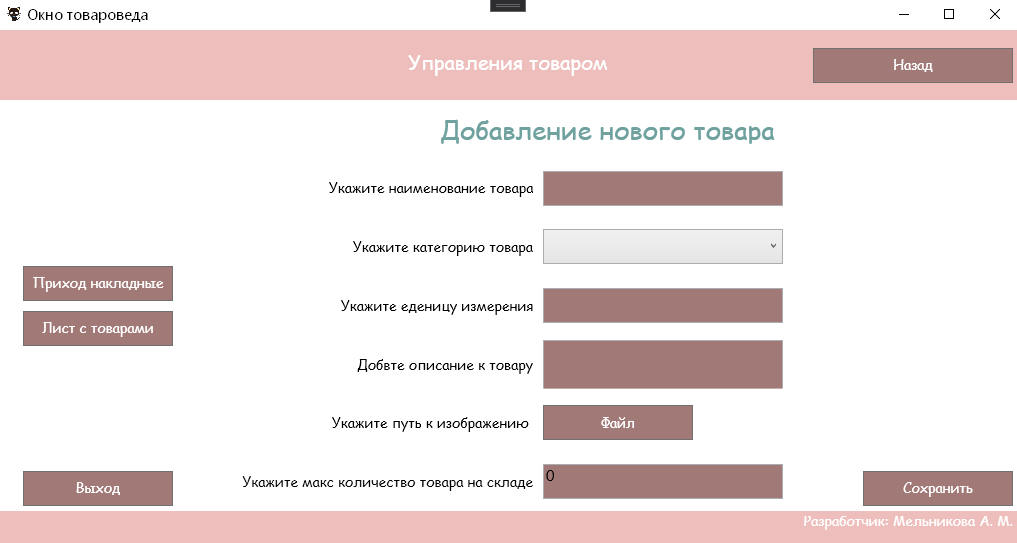


Рисунок 53 – Экранная форма страницы для добавления новых товаров

После нажатия на кнопку «Файл», перед пользователем предоставится окно, в котором он может выбрать картинку для товара. Код обработчика кнопки, предоставлен не рисунке Б.19.

Рассмотрим прототипы окон прецедентов для роли менеджер по продажам.

После авторизации, менеджер по продажам сразу может работать с вариантом использования «Работа с анализами продаж». Для этого пользователю сначала необходимо выбрать товар, далее нажать на кнопку «График продаж». На рисунках 54 – 55 представлены страниц для работы с этим прецедентом.

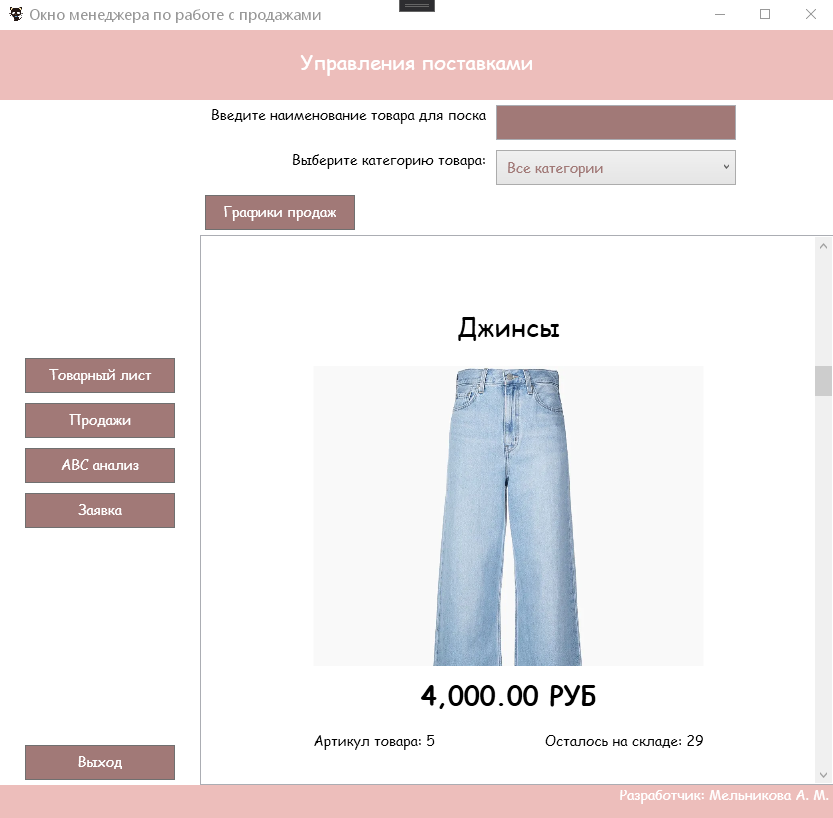


Рисунок 54 – Экранная форма просмотра информации товара

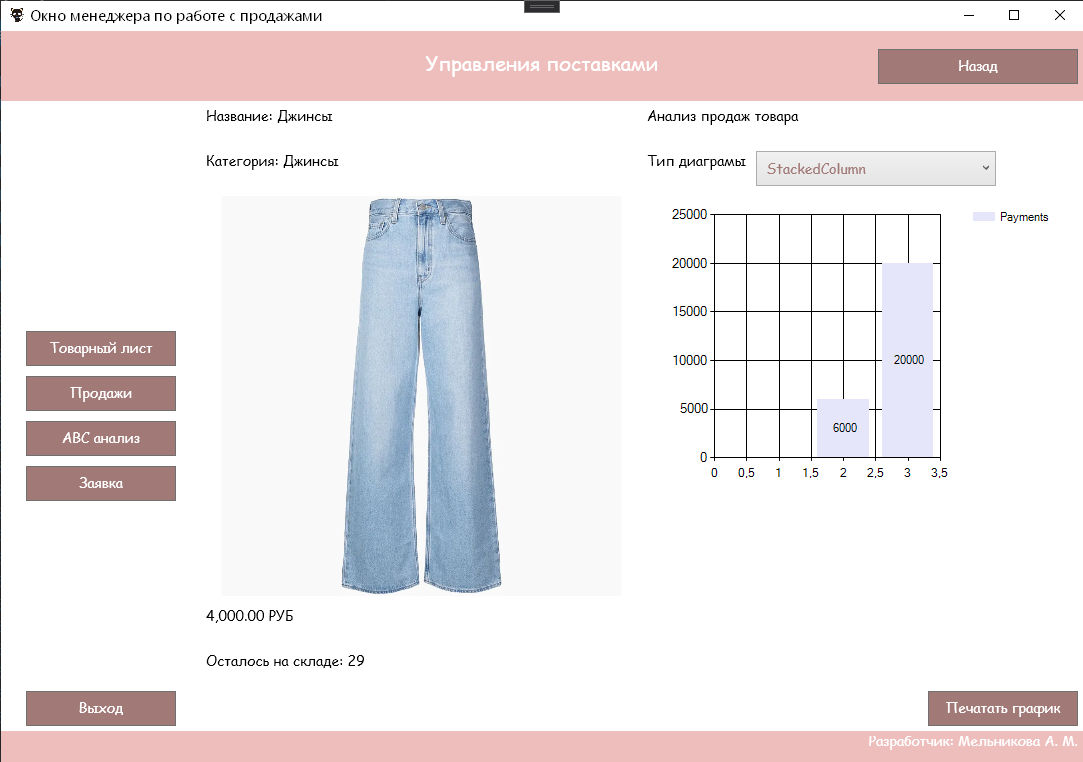


Рисунок 55 – Экранная форма страницы для просмотра графиков продаж товара и вывода их на печать

Для выполнения прецендента «Ведение данных о продажах», были реализованы страницы, которые представлены на рисунках 56 – 57.

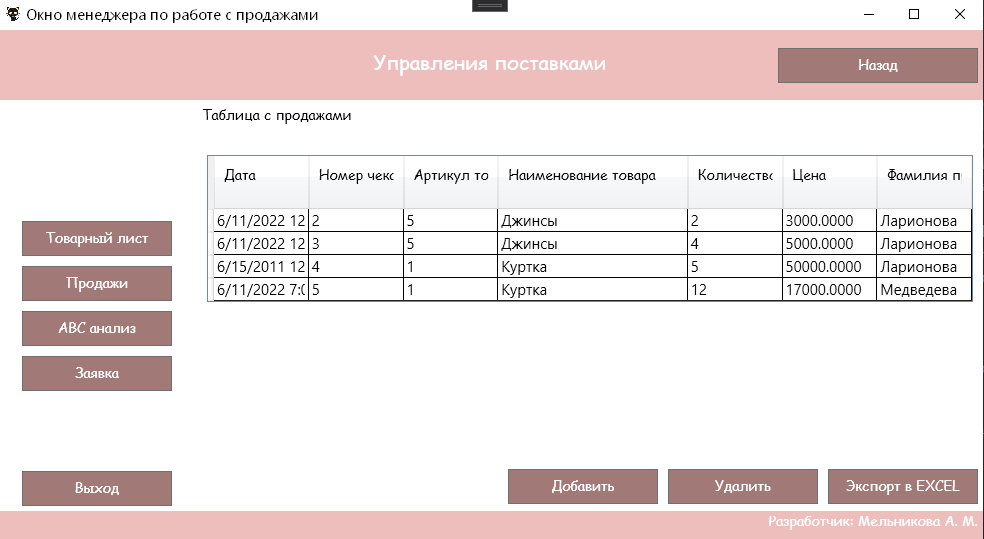


Рисунок 56 – Экранная форма для просмотра продаж

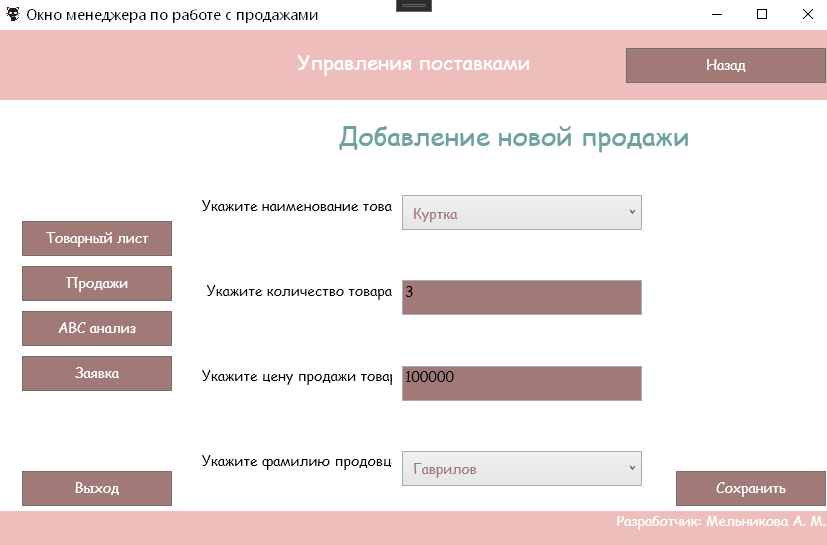


Рисунок 57 – Экранная форма страницы для добавления новой продажи

Для работы менеджера по продажам с вариантом использования «Работа с заявками на поставку». На рисунках 58 – 59 представлены страницы для работы с этим прецедентом.

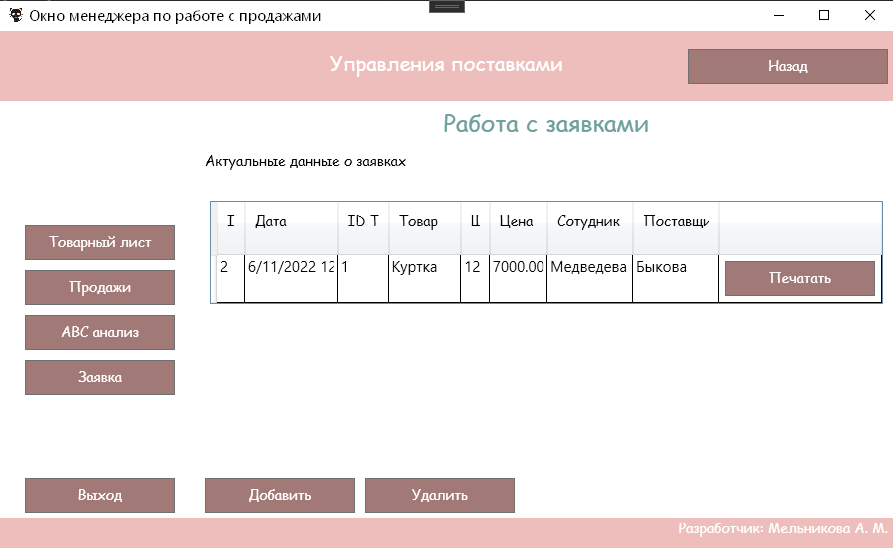


Рисунок 58 – Экранная форма страницы для просмотра данных о уже зарегистрированных заявках в системе

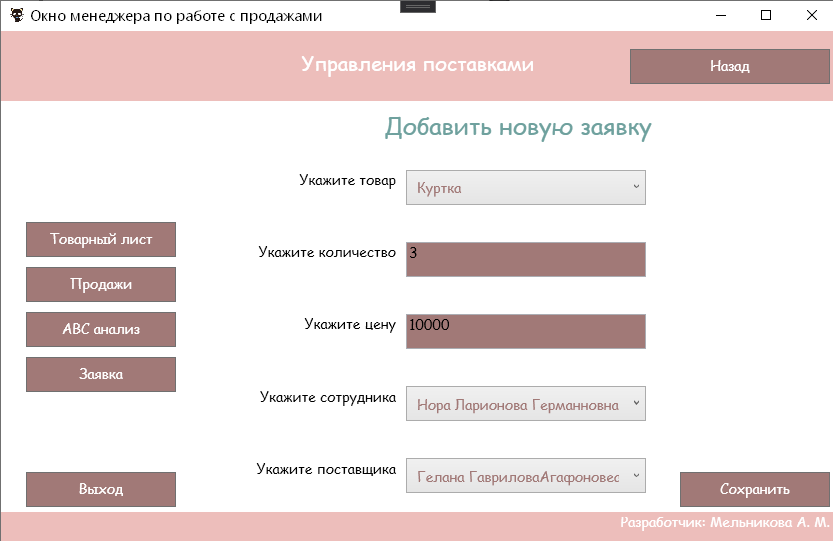


Рисунок 59 – Экранная форма страницы для формирования новых заявок на поставку товара

Для того чтобы заявка сохранилась в виде документа Word, необходимо пользователю нажать на кнопку «Печать» рядом с нужной заковкой, далее появится страница с содержимым заявки, которое пользователь не может редактировать. Страница представлена на рисунке 60.



Рисунок 60 – Экранная форма страницы для данных печатной заявки

В результате на компьютер пользователя сохраняется заявка, в виде документа Word, конечный её формат представлен на рисунке 61. Код обработчика кнопки «Печать» представлена на рисунке Б.20.

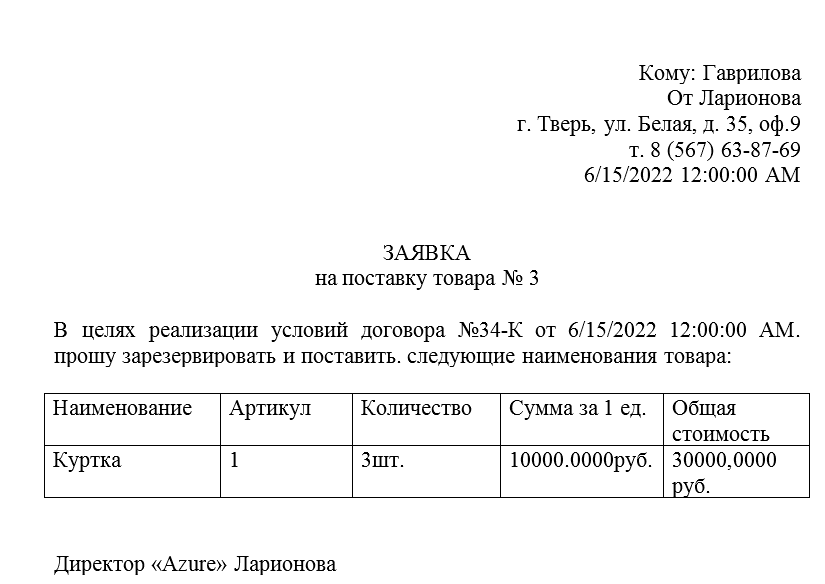


Рисунок 61 – Заявка на поставку нового товара

Чтобы менеджер по продажам мог эффективно и быстро формировать новые заявки на поставку, был реализован ABC анализ продаж страница которого представлен на рисунке 62. Код экранной формы страницы «ABC анализ» представлен на рисунке Б.21.

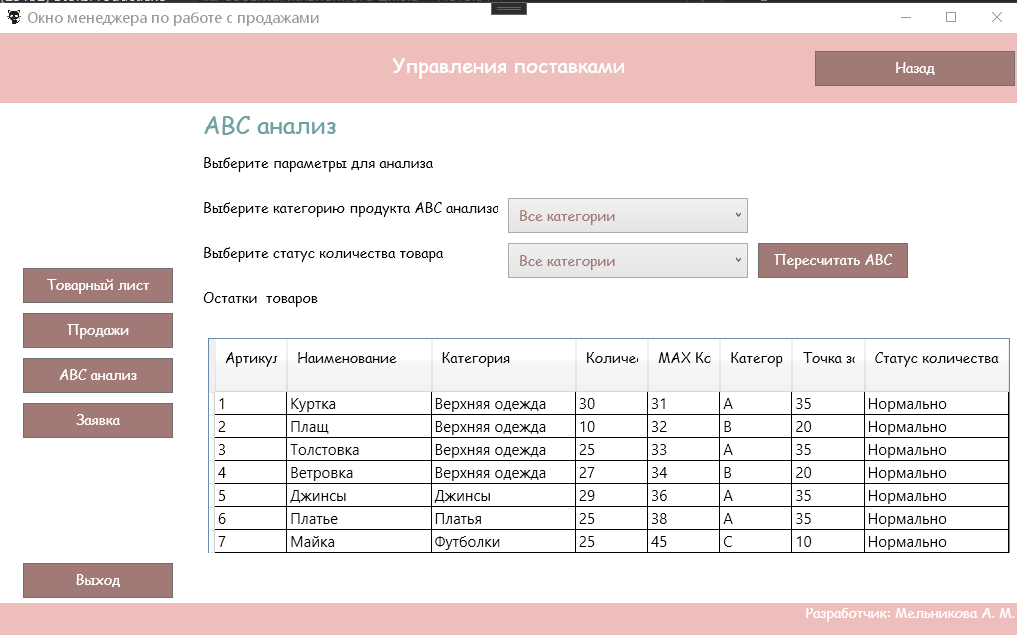


Рисунок 62 – Экранная форма страницы для просмотра свойств товаров ABC анализа

При нажатии на кнопку «Пересчитать ABC» таблица с данными ABC анализа обновляется и перезаполняется при изменениях записей. Обработчик кнопки представлен на рисунке Б.22.

4 Тестирование программных модулей

4.1 Интеграционное тестирование

Основной задачей интеграционного то тестирования для программной подсистемы «Azure» является проверка выполнение авторизации пользователей при различных условиях. Система не должна предоставлять доступ незарегистрированными пользователем и корректно переводить на следующие окно уже зарегистрированного пользователя. Тест- кейсы представлены в Приложении Д.

4.2 Модульное тестирование

Основной задачей модульного тестирования для программной подсистемы «Azure» проверка на ввод и обработку различных типов данных пользователем. На рисунке 63 представлены unit - тесты, а на рисунке 65 результаты их выполнения.

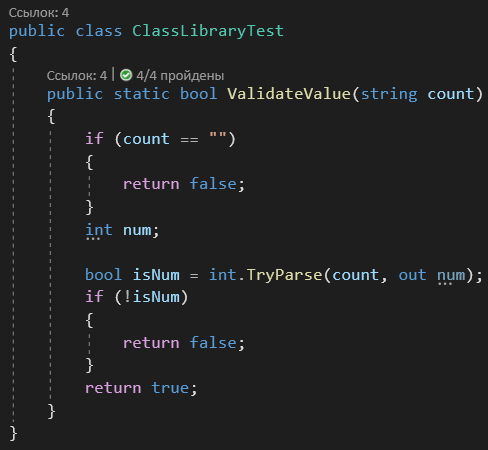


Рисунок 63 – Обработчик модульных тестов

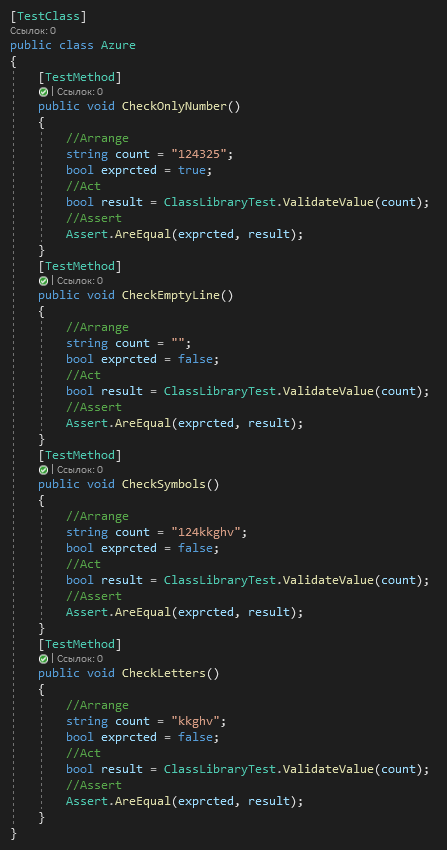


Рисунок 64 – Unit – тесты

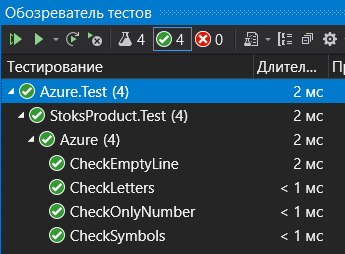


Рисунок 65 – Результат выполнения тестов

5 Эксплуатационная документация

5.1 Общие указания

Инструкция составлена для компьютерного оборудования, характеристики которого удовлетворяют требования технического задания [4.3] с установленной операционной системой Windows 10.

В информационной подсистеме реализованы функции ведение учёта товара, проведение анализа продаж, ведение продаж, ведение приходных накладных, ведение товара, производится учёт поставщиков и заказ нового товара.

5.2 Меры безопасности

За аппаратным обеспечением строго запрещается:

* трогать разъемы соединительных кабелей;
* прикасаться к питающим проводам и устройствам заземления;
* прикасаться к экрану и к тыльной стороне монитора, клавиатуры;
* включать и отключать аппаратуру без указания преподавателя;
* класть диск, книги, тетради на монитор и клавиатуру;
* работать во влажной одежде и влажными руками.

5.3 Порядок работы

К эксплуатации оборудования допускается рабочий персонал, который имеет базовый навык в работе с компьютерным оборудованием.

Порядок проведения проверки знаний для работы с программным приложением не требуется.

5.4 Проверка правильности функционирования

Проверкой работоспособности оборудования занимается технический администратор, который должен выполнять полный его разбор раз в год

Проверка актуальности работы алгоритмов анализа и графиков осуществляется старшим менеджером по продажам. Он выполняет те анализы, которые может выполнять система, но вручную на бумажном носителе, далее сравниваются результаты анализов.

5.5 Указания о действиях в разных режимах

При нормальном режиме программное оборудование будет выполнять свои функции корректно и без сбоев.

При аварийном отключении оборудования вся информация, которая уже была сохранена в системе будет предоставлять свои данные для дальнейшего использования. Если данные не сохранились в системе, то доступ к ней не недоступен, информация считается потерянной.

Заключение

В результате проделанной работы в рамках курсового проектирования была выполнена разработка подсистемы анализа продаж розничного магазина одежды «Azur». Разработка программных модулей системы велась согласно требованиям ГОСТ 19.102-77 «Единая система программной документации (ЕСПД). Стадии разработки».

Задачи, которые стояли передо мной как разработчика в начале выполнения проекта были:

* разработать удобный и интуитивно понятный интерфейс приложения;
* создать базу данных, которая хранит в себе все необходимые сведения для ведения учета товаров и их продаж;
* реализовать наглядное представление результатов анализов для пользователя.

Технический проект разработан в соответствии с техническим заданием. При его разработке выполнялись работы, необходимые для обеспечения предъявляемых к изделию требований и позволяющие получит полное представление о структуре разработанного программного обеспечения.

Результатами реализации информационно-программного компонента является полностью готовое к работе приложение анализа продаж розничного магазина одежды «Azur». Программное приложение полностью соответствует требованиям, указанных в техническом задании и спроектировано на основе технического проекта.

Итоги тестирования показывают, что поведение системы полностью удовлетворяет цели тестирования и дефектов в работе с ней не было найдено.

Содержание эксплуатационной документации даёт полностью понять пользователю как эксплуатировать систему в дальнейшей её корректной и эффективной работе.

В дальнейшем развитии разработки система может дополняться внедрением дополнительных модулей по проведению ещё эффективному проведению анализа продаж товара и его реализацией, разработкой подсистемы лояльности клиентов, ведением клиентской базы.

Результаты проведённой работы по созданию программного приложения «Azur» можно использовать в работе учёта товарооборота в магазине одежды, ведение и учёт сотрудников, необходимой документации для работы с товарами с их поставками и продажами.

**Список использованных источников**

1. ГОСТ 2.105 – 2019. Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам (Издание с Изменением N 1) = Unified system for design documentation. General requirements for textual documents: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 29 апреля 2019 г. № 175-ст: введен впервые: дата введения 2021-02-01 / Разработан Федеральным государственным унитарным предприятием «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ»). – Москва.: Стандартинформ, 2021. – 35 с. –Текст непосредственный.
2. ГОСТ 7.0.100 – 2018. Система стандартов по информации, библиотечному и издательскому делу (СИБИД). Библиографическая запись. Библиографическое описание. Общие требования и правила составления (с Поправкой) = System of standards on information, librarianship and publishing. Bibliographic record. Bibliographic description. General requirements and rules: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 года № 1050-ст: введен впервые: дата введения 2019-07-01 / Разработан Федеральным государственным унитарным предприятием "Информационное телеграфное агентство России (ИТАР-ТАСС)", филиал "Российская книжная палата", Федеральным государственным бюджетным учреждением "Российская государственная библиотека", Федеральным государственным бюджетным учреждением "Российская национальная библиотека". – Москва: Стандартинформ, 2018. – 128 с. – Текст непосредственный.
3. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1) = Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3351: введен впервые: дата введения 1980-01-01. – Москва: Стандартинформ, 2010. – 4 с. – Текст непосредственный.
4. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: руководящий документ по стандартизации: издание официальное: утверждены и введены в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 27 декабря 1990 г. № 3380: дата введения 1992-01-01 / Разработан Министерством электротехнической промышленности и приборостроения СССР. – Москва.: ИПК Издательство стандартов, 2002 г. – 27 с. –Текст непосредственный.
5. Рудаков, А.В. Технология разработки программных продуктов: учебник для студ. учреждений среднего профессионального образования / А.В. Рудаков. – 11-е изд., стер. – М.: Издательский центр «Академия», 2017. – 208 с. – (Профессиональное образование). – ISBN 978-5-4468-4734-1. – Текст: непосредственный.
6. Семакин, И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-6228-3. – Текст: непосредственный.
7. Федорова, Г.Н. Основы проектирования баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 224 с. – (Профессиональное образование). – ISBN 978-5-4468-6170-5. – Текст: непосредственный.
8. Федорова, Г.Н. Разработка, администрирование и защита баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 288 с. – (Профессиональное образование). – ISBN 978-5-4468-6994-7. – Текст: непосредственный.
9. Российская Федерация. Министерство образования и науки. Об утверждении федерального государственного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование: Приказ Министерства образования науки Российской Федерации от 09.12.2016 № 1547: редакция от 16.01.2017: зарегистрировано в Минюсте России 26.12.2016 № 44936. – Текст: электронный // СПС «Гарант» [сайт] – URL: [https://www.garant.ru/ products/ipo/prime/](https://www.garant.ru/%20products/ipo/prime/) doc/71477324/#review (дата обращения: 11.10.2020).
10. [METANIT.COM](https://metanit.com/). Сайт о программировании. Полное руководство по языку программирования С# 9.0 и платформе .NET 5: [сайт] – Текст: электронный. – URL: <https://metanit.com/sharp/tutorial> (дата обращения 20.10.2020).

Приложение А

(обязательное)

Техническое задание. Требования к программным модулям

1 Введение

1.1 Наименование программы

Наименование программной подсистемы «Azur»

1.2 Краткая характеристика области применения

Программная подсистема «Azur» предназначена для автоматизации процесса анализа данных о торговле товарами, на основе регулярного мониторинга продаж единственного филиала магазина одежды.

2 Основания для разработки

Основанием для разработки является Договор 36 от 10.04.2022. Договор утвержден Директором магазина одежды «Azur» Ивановым Иваном Ивановичем, именуемым в дальнейшем Заказчиком, и Мельниковой Александрой Михайловной, именуемая в дальнейшем Исполнителем, 10.07.2022.

Согласно Договору, Исполнитель обязан разработать и установить программную систему «Azur» на оборудовании Заказчика не позднее 10.08.2022, предоставить исходные коды и документацию к разработанной системе не позднее 25.06.2022.

Наименование темы разработки – «Разработка подсистемы анализа продаж розничного магазина одежды «Azur».

Условное обозначение темы разработки (шифр темы) – «Azur -01».

3 Назначение разработки

3.1 Функциональное назначение

Программная подсистема будет использоваться тремя категориями пользователей: менеджер по продажам, товаровед, администратор.

Программная подсистема должна обеспечить возможности: выявление групп товаров, которые генерируют наибольшую прибыль в магазине, а какие наименьшую; проведение сравнения эффективности работы персонала на разных сменах; выявление сезонность продаж. Также пользователь системы может наблюдать количество проданного товара и его сколько человек его вернуло, после покупки, с указанием причины.

Менеджер по продажам имеет возможность просматривать, входные данные, просматривать и настраивать анализы товаров, работать с графиками и составлять по ним отчёты. Также на основе анализов Менеджер по продажам может предоставлять рекомендации по закупке товаров или изменять программы лояльности.

Товаровед имеет возможность просматривать, вносить и редактировать данные в систему, составлять отчёты по входной информации.

Администратор может регистрировать новых пользователей системы и удалять старых, отслеживать историю входа, изменять роли пользователей.

3.1 Эксплуатационное назначение

Программная система будет использоваться на рабочих местах пользователей, представляющих собой персональные компьютеры, имеющие доступ к сети Интернет.

4 Требования к программе

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

В разрабатываемой системе должны храниться результаты анализа за разный период времени и графики, в которых отображаются данные о продажах товаров.

Анализировать продажи необходимо по нескольким группам:

1. Анализ структуры товарооборота (ABC-анализ). Позволяет выявить наиболее прибыльные товарные группы и товары, приносящие минимум дохода.

Графики должны выстраиваться по продаже конкретного товара за каждый чек в разный период времени.

Система должна из базы данных о товарах брать данные, необходимые ей для работы, проводить анализ, результаты которого представлены в виде диаграмм или таблиц, удобные для чтения. Также подсистема должна рекомендовать выполнить действия по ведению торговли.

4.1.2 Требования к организация входных и выходных данных

Требования к входным и выходным данным описаны на основе данных собеседования с заказчиком.

Входные данные:

* товар – артикул, название, описание товара, характеристики (цвет, размер, вид), количество единиц товара находящиеся на складе и в торговом зале;
* чек – фиксирование продажи, дата, товар, который клиент купил, количество, проданного товара, цена товара, общая стоимость чека, кассир, обслуживающий клиента;
* чек о возврате товара – причина возврата товара, товар, который клиент вернул, количество, товара, цена товара, общая стоимость чека, кассир, обслуживающий клиента;
* сведения о пользователях системы (ФИО, должность), данные о продавцах магазина, количество проданного ими товара за определённый период времени.

Выходные данные:

* динамика товара как единицы, так и целой категории, представленный в виде наглядных графиков или составленных в отчёты;
* рекомендации по ведению продажи товарами (что лучше закупить);
* таблицы с данными товаров и продаж.

4.2 Требования к надежности и безопасности

Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему. Необходимо разграничить пользовательские права, т.е. не дать менеджеру по продажам возможности просмотр статистик по эффективности работы персонала. Одновременно в системе может находиться несколько пользователей. Система должна обеспечить одновременный доступ к одним и тем же данным в базе данных, для чего необходимо применение архитектуры клиент–сервер. Кроме того, в системе должно быть предусмотрено резервное копирование и восстановление данных, а также самовосстановление системы после сбоев в операционной системе или отключения электропитания.

4.3 Требования к составу и параметрам технических средств

Состав технических средств предусматривает наличие компьютеров – рабочих станций и компьютеров -серверов.

Компьютер-рабочая станция включает:

* процессор x86 с тактовой частотой, не менее 1 ГГц;
* оперативную память объемом, не менее 4 Гб;
* видеокарту, монитор, мышь, клавиатура.

Два компьютера – сервера (для функционирования СУБД) (основной и резервный), включают в себя:

* процессор x86 с тактовой частотой, не менее 1 ГГц;
* оперативную память объемом, не менее 1 Гб;
* видеокарту, монитор, мышь.

4.4 Требования к информационной и программной совместимости

На серверах должны быть установлены ОС Windows Server 2019, сервер БД Microsoft SQL Server Express 2019, SQL Server Management Studio 18.8.

На компьютерах-рабочих станциях должна быть установлена операционная система ОС Microsoft Windows 10 Pro (Edu). Все формируемые отчеты должны иметь возможность экспортирования в редактор электронных таблиц Microsoft Office 2019 или 365 (Excel).

Приложение Б

(обязательное)

Программный код

Программный код представлен в контексте текстового редактора MS Visual Studio 2019 Enterprise 2019 Версия 16.9.0.

<Page x:Class="StoksProduct.Authoriz.AuthorizPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Authoriz"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AuthorizPage">

<Grid>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<TextBlock HorizontalAlignment="Center" FontSize="24" Margin="0 0 0 50">Авторизация</TextBlock>

<TextBlock>Введите имя</TextBlock>

<TextBox MaxLength="50" Name="tbNameUser"></TextBox>

<TextBlock>Введите пароль</TextBlock>

<TextBox MaxLength="50" Name="tbUserPassword" ></TextBox>

<PasswordBox MaxLength="50" Name="psbUser" Margin="0 -45 0 0"></PasswordBox>

<CheckBox Name="chkPassword" Click="chkPassword\_Click" >Просмотр пароля</CheckBox>

<Button Margin="0 50 0 0" Name="btnNext" Click="btnNext\_Click">Далее</Button>

</StackPanel>

</Grid>

</Page>

Рисунок Б.1 – Программный код разметки страницы «Авторизация»

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

{

if (tbNameUser.Text == "")

{

MessageBox.Show("Введите имя пользователя", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

if (psbUser.Password == "")

{

MessageBox.Show("Введите пароль", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

if (tbNameUser.Text != "" && psbUser.Password != "")

{

string login = tbNameUser.Text.Trim();

string password = Convert.ToString(psbUser.Password.Trim());

Authorization(login, password);

}}

public void Authorization(string login, string password)

{

var account = App.GetContext.Account\_Worker.ToList();

var curentAccount = account.Where(p => p.Name\_User.Equals(login)).ToList();

if (curentAccount.Count == 1)

{

if (curentAccount[0].Passord\_User.Equals(password))

{

switch (curentAccount[0].Role.Name\_Role)

{

case "Адинистратор":

AdminWindow adminWindow = new AdminWindow();

adminWindow.Show();

Login\_Histoy count = new Login\_Histoy

{

Date = DateTime.Now,

Account\_Worker = curentAccount[0]

};

App.GetContext.Login\_Histoy.Add(count);

App.GetContext.SaveChanges();

AccontHelperClass.AccountWorker = curentAccount[0];

FrameAndDataClass.AuthWin.Close();

break;

case "Товаровед":

TovarWindow tovarWindow = new TovarWindow();

tovarWindow.Show();

Login\_Histoy ccount = new Login\_Histoy

{

Date = DateTime.Now,

Account\_Worker = curentAccount[0]

};

App.GetContext.Login\_Histoy.Add(ccount);

App.GetContext.SaveChanges();

AccontHelperClass.AccountWorker = curentAccount[0];

FrameAndDataClass.AuthWin.Close();

break;

case "Менеджер по продажам":

MenegerWindow menegerWindow = new MenegerWindow();

menegerWindow.Show();

Login\_Histoy cccount = new Login\_Histoy

{

Date = DateTime.Now,

Account\_Worker = curentAccount[0]

};

App.GetContext.Login\_Histoy.Add(cccount);

App.GetContext.SaveChanges();

AccontHelperClass.AccountWorker = curentAccount[0];

FrameAndDataClass.AuthWin.Close();

break;

default:

MessageBox.Show("Ошибка авторизации");

break;

}

}

else

MessageBox.Show("Неверный пароль");

}

else

MessageBox.Show("Неверный логин");

}

Рисунок Б.2 – обработчика для кнопки «Далее» на странице «Авторизация»

<Page x:Class="StoksProduct.Admin.MenuAdminPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Admin"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800" IsVisibleChanged="Page\_IsVisibleChanged"

Title="MenuAdminPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<StackPanel>

<TextBlock>Таблица с аккаунтами пользователей</TextBlock>

<DataGrid IsReadOnly="True" AutoGenerateColumns="False" Name="dgrdUser" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Header="ФИО" Binding="{Binding Directory\_Worker.FullName}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Логин" Binding="{Binding Name\_User}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Пароль" Binding="{Binding Passord\_User}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Роль" Binding="{Binding Role.Name\_Role}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Описание" Binding="{Binding Description\_User}" Width ="\*"></DataGridTextColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Name="btnEdit" Click="btnEdit\_Click"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid></StackPanel>

<StackPanel Grid.Row="2" VerticalAlignment="Bottom" Orientation="Horizontal">

<Button Name="btnAdd" Content="Добавить" Click="btnAdd\_Click" />

<Button Name="btnDelet" Content="Удалить" Click="btnDelet\_Click"></Button>

</StackPanel>

<StackPanel Grid.Row="2" VerticalAlignment="Bottom" HorizontalAlignment="Right">

<Button Name="btnWrite" Content="Экспорт в EXCEL" Click="btnWrite\_Click" ></Button></StackPanel> </Grid></Page>

Рисунок Б.3 – Программный код страницы «Действующие аккаунты сотрудников»

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

{ FrameAndDataClass.MainAdminFrame.Navigate(new AddPage(null));}

Рисунок Б.4 – Программный код обработчика для кнопки «Добавить» на странице «Действующие аккаунты сотрудников»

<Page x:Class="StoksProduct.Admin.AddPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Admin"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<TextBlock Grid.ColumnSpan="2" Text="Добавление нового аккаунта" Foreground="#6FA19E" FontSize="26" VerticalAlignment="Center" HorizontalAlignment="Center"></TextBlock>

<Grid Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Выберите пользователя"/></Grid>

<Grid Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Укажите логин"/></Grid>

<Grid Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Укажите пароль"/></Grid>

<Grid Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Добавте роль пользователя"/></Grid>

<Grid Grid.Row="5" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Добавте описание к акаунту"/></Grid>

<Grid Grid.Row="1" Grid.Column="1" HorizontalAlignment="Left">

<ComboBox SelectedItem="{Binding Directory\_Worker}" IsReadOnly="True" Name="cbLastName" DisplayMemberPath="FullName"></ComboBox></Grid>

<Grid Grid.Row="2" Grid.Column="1" HorizontalAlignment="Left">

<TextBox Text="{Binding Name\_User}" MaxLength="50" Name="tbLogin"></TextBox></Grid>

<Grid Grid.Row="3" Grid.Column="1" >

<TextBox Text="{Binding Passord\_User}" MaxLength="50" Name="tbPassword" HorizontalAlignment="Left"></TextBox></Grid>

<Grid Grid.Row="4" Grid.Column="1" >

<ComboBox SelectedItem="{Binding Role}" Name="cbRole" DisplayMemberPath="Name\_Role" HorizontalAlignment="Left"></ComboBox></Grid>

<Grid Grid.Row="5" Grid.Column="1" >

<TextBox Text="{Binding Description\_User}" MaxLength="100" Name="tbDescription" Height="40" Width="400" HorizontalAlignment="Left"></TextBox>

<Button Name="btnSave" Content="Сохранить" VerticalAlignment="Bottom" HorizontalAlignment="Right" Click="btnSave\_Click"></Button>

</Grid>

</Grid>

</Page>

Рисунок Б.5 – Программный код страницы «Добавление нового аккаунта»

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

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(tbLogin.Text))

errors.AppendLine("Укажите Логин сотрудника");

if (string.IsNullOrWhiteSpace(tbPassword.Text))

errors.AppendLine("Укажите Пароль сотрудника");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;}

//Создается нновый экземпляр аккаунта, к которому присваиваются значния

Account\_Worker currentAccount = new Account\_Worker

{ ID\_Accont\_Worker = \_currentAccount.ID\_Accont\_Worker,

ID\_Worker = (cbLastName.SelectedItem as DateBase.Directory\_Worker).ID\_Worker,

ID\_Role = (cbRole.SelectedItem as Role).ID\_Role,

Name\_User = tbLogin.Text,

Passord\_User = tbPassword.Text,

Description\_User = tbDescription.Text};

//Если аккаунта ещё нет в бд, то создаётся новая запись

if (currentAccount.ID\_Accont\_Worker == 0)

App.GetContext.Account\_Worker.Add(currentAccount);

try

{ App.GetContext.SaveChanges();

MessageBox.Show("Информация сохранена!");

FrameAndDataClass.MainAdminFrame.GoBack();}

catch (Exception ex)

{ MessageBox.Show(ex.Message.ToString());}}

Рисунок Б.6 – Программный код обработчика для кнопки «Сохранить» на странице «Добавление нового аккаунта»

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

{

FrameAndDataClass.MainAdminFrame.Navigate(new AddPage((sender as System.Windows.Controls.Button).DataContext as Account\_Worker));

}

Рисунок Б.7 – Программный код обработчика для кнопки «Редактировать» на странице «Действующие аккаунты сотрудников»

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

{

var usersForRemoving = dgrdUser.SelectedItems.Cast<Account\_Worker>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {usersForRemoving.Count()} элементов?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

App.GetContext.Account\_Worker.RemoveRange(usersForRemoving);

App.GetContext.SaveChanges();

MessageBox.Show("Данные удалены!");

dgrdUser.ItemsSource = App.GetContext.Account\_Worker.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}}}

Рисунок Б.8 – Программный код обработчика для кнопки «Удалить» на странице «Действующие аккаунты сотрудников»

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

{ //Создаётся открытие приложения

Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1]

for (int j = 0; j < dgrdUser.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dgrdUser.Columns[j].Header;

}

for (int i = 0; i < dgrdUser.Columns.Count; i++)

{

for (int j = 0; j < dgrdUser.Items.Count; j++)

{

TextBlock b = dgrdUser.Columns[i].GetCellContent(dgrdUser.Items[j]) as TextBlock;

if (b == null)

break;

Range myRange = (Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}}}

Рисунок Б.9 – Программный код обработчика для кнопки «Экспорт в Excel» на странице «Действующие аккаунты сотрудников»

<Page x:Class="StoksProduct.Admin.WorkerPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Admin"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800" IsVisibleChanged="Page\_IsVisibleChanged"

Title="WorkerPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="31\*"/>

<RowDefinition Height="369\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<StackPanel Grid.RowSpan="2">

<TextBlock>Таблица сотрудников</TextBlock>

<DataGrid IsReadOnly="True" AutoGenerateColumns="False" Name="dgrdWorker" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Header="ID Сотрудника" Binding="{Binding ID\_Worker}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Фамилия" Binding="{Binding Last\_Name}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Имя" Binding="{Binding First\_Name}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Отчество" Binding="{Binding Father\_Name}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Должность" Binding="{Binding Post.Name\_Post}" Width ="\*"></DataGridTextColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Name="btnEdit" Click="btnEdit\_Click"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</StackPanel>

<StackPanel Grid.Row="2" VerticalAlignment="Bottom" Orientation="Horizontal" >

<Button Name="btnAdd" Content="Добавить" Click="btnAdd\_Click" />

<Button Name="btnDelet" Content="Удалить" Click="btnDelet\_Click"></Button>

</StackPanel>

<StackPanel Grid.Row="2" VerticalAlignment="Bottom" HorizontalAlignment="Right">

<Button Name="btnWrite" Content="Экспорт в EXCEL" Click="btnWrite\_Click" ></Button>

</StackPanel>

</Grid>

</Page>

Рисунок Б.10 – Программный код страницы «Сотрудники»

<Page x:Class="StoksProduct.Admin.AddWorkerPage1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Admin"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddWorkerPage1">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<TextBlock Grid.ColumnSpan="2" Text="Добавление нового пользователя" Foreground="#6FA19E" FontSize="26" VerticalAlignment="Center" HorizontalAlignment="Center"></TextBlock>

<Grid Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Укажите фамилию"/>

</Grid>

<Grid Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Укажите имя"/>

</Grid>

<Grid Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Укажите отчество"/>

</Grid>

<Grid Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="Укажите должность"/>

</Grid>

<Grid Grid.Row="1" Grid.Column="1" HorizontalAlignment="Left">

<TextBox MaxLength="50" Text="{Binding Last\_Name}" Name="tbLastName"></TextBox>

</Grid>

<Grid Grid.Row="2" Grid.Column="1" HorizontalAlignment="Left">

<TextBox MaxLength="50" Text="{Binding First\_Name}" Name="tbFirstName"></TextBox>

</Grid>

<Grid Grid.Row="3" Grid.Column="1" HorizontalAlignment="Left">

<TextBox MaxLength="50" Text="{Binding Father\_Name}" Name="tbFatherName"></TextBox>

</Grid>

<Grid Grid.Row="4" Grid.Column="1" >

<ComboBox SelectedItem="{Binding Post}" Name="cbPost" DisplayMemberPath="Name\_Post" HorizontalAlignment="Left"></ComboBox>

<Button Name="btnSave" Content="Сохранить" VerticalAlignment="Bottom" HorizontalAlignment="Right" Click="btnSave\_Click"></Button>

</Grid>

</Grid>

</Page>

Рисунок Б.11 – Программный код страницы «Добавление нового сотрудника»

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

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(tbFirstName.Text))

errors.AppendLine("Укажите Имя сотрудника");

if (string.IsNullOrWhiteSpace(tbLastName.Text))

errors.AppendLine("Укажите Фамилию сотрудника");

if (cbPost.SelectedIndex == -1)

errors.AppendLine("Укажите Должность сотрудника");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

//Создаётся новый экземпляр сотрудника, к которому присваиваются данные

DateBase.Directory\_Worker currentWorker = new DateBase.Directory\_Worker

{

ID\_Worker = \_currentWorker.ID\_Worker,

First\_Name = tbFirstName.Text,

Last\_Name = tbLastName.Text,

Father\_Name = tbFatherName.Text,

ID\_Post = (cbPost.SelectedItem as DateBase.Post).ID\_Post,

};

//Если сторудника нет в бд, то добавляется новый

if (currentWorker.ID\_Worker == 0)

App.GetContext.Directory\_Worker.Add(currentWorker);

try

{

App.GetContext.SaveChanges();

MessageBox.Show("Информация сохранена!");

FrameAndDataClass.MainAdminFrame.GoBack();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

Рисунок Б.12 – Программный код обработчика для кнопки «Сохранить» на странице «Добавление нового сотрудника»

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

{

FrameAndDataClass.MainAdminFrame.Navigate(new AddWorkerPage1((sender as System.Windows.Controls.Button).DataContext as DateBase.Directory\_Worker));

}

Рисунок Б.13 – Программный код обработчика для кнопки «Редактировать» на странице «Сотрудники»

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

{

var usersForRemoving = dgrdWorker.SelectedItems.Cast<DateBase.Directory\_Worker>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {usersForRemoving.Count()} элементов?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{ App.GetContext.Directory\_Worker.RemoveRange(usersForRemoving);

App.GetContext.SaveChanges();

MessageBox.Show("Данные удалены!");

dgrdWorker.ItemsSource = App.GetContext.Directory\_Worker.ToList();}

catch (Exception ex)

{MessageBox.Show(ex.Message.ToString());}}}

Рисунок Б.14 – Программный код обработчика для кнопки «Удалить» на странице «Сотрудники»

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

{

Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

for (int j = 0; j < dgrdWorker.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dgrdWorker.Columns[j].Header;

}

for (int i = 0; i < dgrdWorker.Columns.Count; i++)

{

for (int j = 0; j < dgrdWorker.Items.Count; j++)

{

TextBlock b = dgrdWorker.Columns[i].GetCellContent(dgrdWorker.Items[j]) as TextBlock;

if (b == null)

break;

Range myRange = (Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

Рисунок Б.15 – Программный код обработчика для кнопки «Экспорт в Excel» на странице «Сотрудники»

<Page x:Class="StoksProduct.Admin.LoginHistoryPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Admin"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="LoginHistoryPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<StackPanel>

<TextBlock>Таблица с историей входа пользователей в систему </TextBlock>

<DataGrid IsReadOnly="True" AutoGenerateColumns="False" Name="dgrdStory" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Header="Время входа" Binding="{Binding Date}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="ФИО" Binding="{Binding Account\_Worker.Directory\_Worker.FullName}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Логин" Binding="{Binding Account\_Worker.Name\_User}" Width ="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Роль" Binding="{Binding Account\_Worker.Role.Name\_Role}" Width ="\*"></DataGridTextColumn></DataGrid.Columns>

</DataGrid>

</StackPanel>

<StackPanel Grid.Row="2" VerticalAlignment="Bottom" HorizontalAlignment="Right">

<Button Name="btnWrite" Content="Экспорт в Excel" Click="btnWrite\_Click" ></Button></StackPanel></Grid></Page>

Рисунок Б.16 – Программный код страницы «Просмотр историй входа»

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

{

Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

for (int j = 0; j < dgrdStory.Columns.Count; j++){

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dgrdStory.Columns[j].Header;

}

for (int i = 0; i < dgrdStory.Columns.Count; i++){

for (int j = 0; j < dgrdStory.Items.Count; j++){

TextBlock b = dgrdStory.Columns[i].GetCellContent(dgrdStory.Items[j]) as TextBlock;

if (b == null)break;

Range myRange = (Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;}}}

Рисунок Б.17 – Программный код обработчика для кнопки «Экспорт в Excel» на странице «Просмотр историй входа»

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

{

StringBuilder errors = new StringBuilder();

if (cbSupplier.SelectedIndex == -1)

errors.AppendLine("Укажите поставщика");

if (cbSotrud.SelectedIndex == -1)

errors.AppendLine("Укажите сотрудника");

if (cbTovar.SelectedIndex == -1)

errors.AppendLine("Укажите товар");

if (string.IsNullOrWhiteSpace(tbCount.Text))

errors.AppendLine("Укажите количество товара");

if (tbCount.Text == "")

errors.AppendLine("Укажите верные данные о количестве товара");

if (tbPrise.Text == "")

errors.AppendLine("Укажите укажите цену товара в рублях");

if (errors.Length > 0)

{ MessageBox.Show(errors.ToString());

return;}

//Создание нового экзепляра приходатовара

Incoming\_Invoice \_currentApp = new Incoming\_Invoice{

ID\_Provider = (cbSupplier.SelectedItem as DateBase.Directory\_Provider).ID\_Provider,

ID\_Product = (cbTovar.SelectedItem as DateBase.Directory\_Product).ID\_Product,

ID\_Worker = (cbSotrud.SelectedItem as DateBase.Directory\_Worker).ID\_Worker,

Date = DateTime.Now,

Count = Convert.ToInt32(tbCount.Text),

Incoming\_Price = Convert.ToDecimal(tbPrise.Text)};

if(\_currentApp.ID\_Incoming\_Invoice == 0)

App.GetContext.Incoming\_Invoice.Add(\_currentApp);

try{

Directory\_Product product = App.GetContext.Directory\_Product.Where(c => c.ID\_Product == \_currentApp.ID\_Product).FirstOrDefault();

product.Count\_Product += Convert.ToInt32(tbCount.Text);

product.Price\_Product = Convert.ToInt32(tbCount.Text);

App.GetContext.SaveChanges();

MessageBox.Show("Информация сохранена!");

FrameAndDataClass.TovarFrame.GoBack();}

catch (Exception ex){

MessageBox.Show(ex.Message.ToString());}}

Рисунок Б.18 – Программный код обработчика для кнопки «Сохранить» на странице «Добавление новых приходных накладных»

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

{OpenFileDialog dialog = new OpenFileDialog();

if (dialog.ShowDialog() == true){

\_currentProduct.Imag = File.ReadAllBytes(dialog.FileName);

MessageBox.Show("Картинка добавлена");}}

Рисунок Б.19 – Программный код обработчика для кнопки «Файл» на странице «Добавление нового товара»

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

var helper = new WordHelperClass("Resources/Шаблон - заявка.doc");

var stoim = decimal.Parse(tbCount.Text) \* decimal.Parse(tbPrise.Text.Replace('.', ','));

//Поиск в шаблоне значений и замена их на необходимые

var items = new Dictionary<string, string>{

{"<ORG>", tbSuppliers.Text },

{"<FIO> ", tbSotrud.Text },

{"<DATE>", tbDateAplicate.Text },

{"<ID> ", tbIDAplicate.Text },

{"<PRODUCT>", tbNameProduct.Text },

{"<ID\_PROD>", tbIDProduct.Text },

{"<COUNT> ", tbCount.Text },

{"<STOIM> ", tbPrise.Text },

{"<SUM>", Convert.ToString(stoim)},};

helper.Process(items);

MessageBox.Show("Заявка сформирована успешно. Файл сохранился в папке Resourse");

FrameAndDataClass.MenegerFrame.Navigate(new ApplicationPage());}

Рисунок Б.20 – Программный код обработчика для кнопки «Печать» на странице «Информация о заявке»

private void btnABC\_Click(object sender, RoutedEventArgs e){

ABCMenegerClass.RecalculationABC();

App.GetContext.ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

dtgTovarList.ItemsSource = App.GetContext.Directory\_Product.ToList();

MenegerCritPointClass.RecalculationCritPoint();

App.GetContext.ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

dtgTovarList.ItemsSource = App.GetContext.Directory\_Product.ToList();}

Рисунок Б.22 – Программный код обработчика для кнопки «Пересчитать ABC» на странице «ABC анализ»

<Page x:Class="StoksProduct.Meneger.ABCPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StoksProduct.Meneger"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ABCPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0">

<TextBlock Foreground="#6FA19E" FontSize="24">ABC анализ</TextBlock>

<TextBlock>Выберите параметры для анализа</TextBlock>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Выберите категорию продукта ABC анализа" Width="295"></TextBlock>

<ComboBox Name="cbParametr" DisplayMemberPath="Name\_ABC" SelectionChanged="cbParametr\_SelectionChanged"></ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Выберите статус количества товара" Width="295"></TextBlock>

<ComboBox Name="cbStatus" DisplayMemberPath="Name\_Status" SelectionChanged="cbStatus\_SelectionChanged"></ComboBox>

<Button Name="btnABC" Content="Пересчитать ABC" Click="btnABC\_Click"></Button>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1">

<TextBlock Text="Остатки товаров"></TextBlock>

<DataGrid Name="dtgTovarList" IsReadOnly="True" AutoGenerateColumns="False" VerticalAlignment="Top" Height="auto">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Binding="{Binding ID\_Product}" Header="Артикул"></DataGridTextColumn>

<DataGridTextColumn Width="2\*" Binding="{Binding Name\_Product}" Header="Наименование"></DataGridTextColumn>

<DataGridTextColumn Width="2\*" Binding="{Binding Category\_tb.Name\_Category}" Header="Категория"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Binding="{Binding Count\_Product}" Header="Количество"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Binding="{Binding Max\_Count\_Product}" Header="MAX Количество"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Binding="{Binding ABC.Name\_ABC}" Header="Категория ABC"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Binding="{Binding ABC.Critical\_Point}" Header="Точка запасов в процентах"></DataGridTextColumn>

<DataGridTextColumn Width="2\*" Binding="{Binding Status\_Count.Name\_Status}" Header="Статус количества товара"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid></StackPanel></Grid></Page>

Рисунок Б.21 – Программный код страницы «ABC анализ»

Приложение В

(справочное)

Формы выходных документов

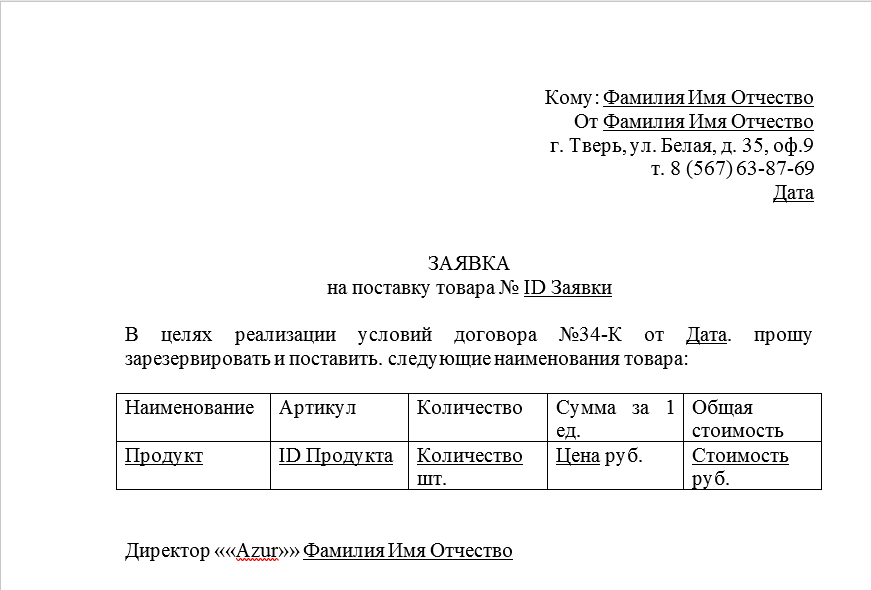


Рисунок В.1 – Шаблон заявки на поставку товара

Приложение Г

(обязательное)

Тест-кейсы

Таблица Д.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | Azur |
| Номер версии | Azur -01 |
| Имя тестера | Мельникова Александра Михайловна |
| Даты тестирования | 12.06.2022 |

Таблица Д.2 – Test case #1

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_1 |
| Приоритет теста | Высокий. |
| Название тестирования/Имя | Авторизация не зарегистрированного пользователя в системе. |
| Резюме испытания | Безуспешная авторизация. |
| Шаги тестирования | Ввод логина, ввод пароля, нажать кнопку «Авторизация» |
| Данные тестирования | Логин: любые данные  Пароль: любые данные  Логин и пароль не должны совпадать с уже имеющимися в бд |
| Ожидаемый результат | Система не даёт доступ к работе с бизнес-процессами системы. |
| Фактический результат | Система не авторизовала пользователя. |
| Предпосылки | У пользователя нет аккаунта в системе, и он не зарегистрирован. |
| Постусловия | Вывод окна с ошибкой авторизации. |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

Таблица Д.3 – Test case #2

|  |  |
| --- | --- |
| Test Case # | TC\_UI\_2 |
| Приоритет теста | Высокий. |
| Название тестирования/Имя | Авторизация зарегистрированного пользователя в системе. |
| Резюме испытания | Успешная авторизация. |
| Шаги тестирования | Ввод логина, ввод пароля, нажать кнопку «Авторизация» |
| Данные тестирования | Логин: meneg  Пароль: 1234 |
| Ожидаемый результат | Система успешно авторизует пользователя и даёт доступ к работе. |
| Фактический результат | Система авторизует пользователя и даёт доступ к работе в роли менеджера по продажам. |
| Предпосылки | Пользователь зарегистрирован в системе и имеет активный аккаунт, данные о котором хранятся в бд. |
| Постусловия | Предоставляет доступ к работе с системой. |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |