**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

направление специальности 1-40 05 01-01 Информационные системы и технологии (в проектировании и производстве)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по дисциплине «Разработка приложений баз данных для информационных систем»

на тему: «**РАЗРАБОТКА *WEB*-ПРИЛОЖЕНИЯ БАЗ ДАННЫХ «ФАБРИКА МЕБЕЛИ**»

Исполнитель: студент гр. ИТП-21

Шеляхин М. С.

Руководитель: преподаватель

Точко В. Н.

Дата проверки: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписи членов комиссии

по защите курсового проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель 2020

Установа адукацыі «Гомельскі дзяржаўны

тэхнічны ўніверсітэт імя П.В. Сухога»

Факультэт аўтаматызаваных і інфармацыйных сістэм

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РЭЦЭНЗІЯ**

**на курсавы праект (работу)**

па дысцыпліне \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ,

выканана студэнтам \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

групы \_\_\_\_\_\_\_\_\_\_

I. Пералік заўваг па тэксту курсавога праекта (работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

II. Агульная характарыстыка работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc53516789)

[1 ЛОГИЧЕСКАЯ И ФИЗИЧЕСКАЯ СТРУКТУРА БАЗЫ ДАННЫХ 6](#_Toc53516790)

[1.1 Логическая структура базы данных 6](#_Toc53516791)

[2.1 Физическая структура спроектированной базы данных 7](#_Toc53516792)

[ЗАКЛЮЧЕНИЕ 9](#_Toc53516793)

[Список используемых источников 10](#_Toc53516794)

[ПРИЛОЖЕНИЕ А 11](#_Toc53516795)

[ПРИЛОЖЕНИЕ Б 11](#_Toc53516796)

[ПРИЛОЖЕНИЕ В 11](#_Toc53516797)

[ПРИЛОЖЕНИЕ Г 11](#_Toc53516798)

[ПРИЛОЖЕНИЕ Д 11](#_Toc53516799)

ВВЕДЕНИЕ

В XXI веке для существования и успешного развития любого предприятия в условиях конкуренции необходимо выполнение следующих условий:

– активное взаимодействие с покупателем в режиме реального времени;

– эффективный контроль потока данных, поступающих от покупателя предприятию и обратно;

– постоянное продвижение предприятия с помощью рекламы.

В качестве современного подхода к реализации всех этих условий применяется создание сайта или клиент-серверного приложения для работы в сети Интернет. Сайты представляют из себя набора страниц, объединенных в единый ресурс, и имеют простую архитектуру и небольшой размер. Приложение представляет из себя компьютерное приложение, разработанное для сети Интернет, и одной из его особенностей является работа с контентом и личными данными пользователя.

Для курсового проекта будет создано *web*-приложение, так как оно дает пользователям возможность вводить, получать и манипулировать данными с помощью взаимодействия. Данное взаимодействие будет характеризовываться возможностью создания заказа предприятию, получения актуальной информации о предприятии и его ресурсах и т.д.

Для фабрики мебели необходимо создать приложение, позволяющее создавать и управлять заказами, изделиями, накладными, работниками и клиентами, иметь авторизацию и управление ролями, а также иметь возможность вывода информации о сотрудниках, поставках и клиентах авторизованным пользователям.

Для реализации поставленной задачи необходимо определить список технологий и программных средств, позволяющих автоматизировать весь процесс. Программа будет состоять из источника данных, представляющего из себя базу данных, и web-приложения, работающего с конкретной базой данных.

В качестве источника данных предпочтительно использовать СУБД (систему управления базами данных). Среди всех выгодно выделяется *MS SQL Server*. Ее главными преимуществами являются производительность, надежность (можно шифровать данные) и простота. Также эта СУБД разработано компанией *Microsoft*, что говорит о раскрытии высокого потенциала при работе с платформой *.NET Framework, .NET Core* и *Visual Studio* в частности.

Для создания *web*-приложения используется технология *ASP.NET Core*, разработанная компанией *Microsoft* для всех основных операционных систем. Программная модель ASP.NET основывается на протоколе HTTP и использует его правила взаимодействия между сервером и браузером. Поскольку *ASP.NET* *Core* основывается на *Common Language Runtime (CLR)*, разработчики могут писать код для *ASP.NET Core*, используя языки программирования, входящие в комплект *.NET* (*C#, Visual Basic.NET, J#* и *JScript .NET*). В курсовом проекте будет использоваться язык *C#* и среда программирования *Visual Studio*.

1. ЛОГИЧЕСКАЯ И ФИЗИЧЕСКАЯ СТРУКТУРА БАЗЫ ДАННЫХ
   1. Логическая структура базы данных

В информационной системе присутствуют следующие сущности:

– «Клиент»;

– «Работник»;

– «Мебель»;

– «Заказ»;

– «Накладная».

Сущность «Мебель» представляет из себя изделие, которое можно заказать на фабрике. Данный компонент будет иметь следующие атрибуты:

– уникальный *id*;

– наименование;

– описание;

– вид материала;

– цена;

– количество.

Для хранения списка изделий необходимо создать таблицу в БД, в которой должнен храниться список сущностей со значениями атрибутов для каждой записи.

Сущность «Работник» представляет из себя работника предприятия. Данный компонент будет иметь следующие атрибуты:

– уникальный *id*;

– ФИО работника;

– должность;

– образование.

Для хранения списка работников также необходимо создать таблицу в БД, в которой должнен храниться список сущностей со значениями атрибутов для каждой записи.

Сущность «Клиент» представляет из себя заказчика, являющегося другим предприятием с представителем и основной информацией. Данная сущность будет иметь следующие атрибуты:

– уникальный *id*;

– название;

– ФИО представителя;

– номер;

– адрес.

Так как представитель является работником другого предприятия, а данные о нем используются только в качестве описательной характеристики, нет необходимости создавать связь с таблицей работников.

Сущность «Заказ» представляет из себя заказ партии мебели, совершенный пользователем и обслуживаемый работником. Заказ должен иметь следующие атрибуты:

– уникальный *id*;

– данные о клиенте;

– данные об изделии;

– количество единиц изделия на складе;

– цена;

– специальная скидка для клиента;

– отметка о выполнении заказа;

– данные о работнике, обслуживающим заказ.

Так как в заказе присутствуют данные о клиенте, изделии и работнике, необходимо организовать связи между таблицами с клиентами, изделиями и работниками, а в качестве атрибутов создать поля с уникальными *id* записей.

Сущность «Накладная» представляет из себя накладную о поступлении материалов для изделий на склад. Накладная должна иметь следующие атрибуты:

– уникальный *id*;

– номер поставщика;

– наименование поставщика;

– дата поступления материала на склад;

– вид материала;

– цена;

– вес;

– данные об изделии, для которого получены материалы;

– данные о работнике, который обеспечивал поступление.

Так как в накладной присутствуют данные об изделии и работнике, необходимо организовать связи между таблицами с изделиями и работниками, а в качестве атрибутов создать поля с уникальными *id* записей.

* 1. Физическая структура спроектированной базы данных

Используя логическую структуру БД, можно воссоздать ее в СУБД *MS SQL Server*.

Спроектированная БД состоит из следующих таблиц:

– «*Clients*» – хранит информацию о клиентах предприятия;

– «*Employees*» – хранит информацию о работниках предприятия;

– «*Furniture*» – хранит информацию об изделиях;

– «*Orders*» – хранит информацию о заказах;

– «*Waybills*» – хранит информация о накладных;

Каждый внешний и первичный ключ представлен числом формата *int*. Номера клиентов и поставщиков представлены числом формата *int* Названия представлены строкой формата *nvarchar*, которая занимает меньший объем памяти, если длина строки меньше заданной:

– названия компаний клиентов – *nvarchar*(25);

* названия изделий – *nvarchar*(50);

– названия поставщиков – *nvarchar*(100).

Цены накладных и заказов представлены данными формата *decimal*. Поле веса представлено в виде числа формата *float*. ФИО работников и представителей представлены строкой формата *nvarchar*(100). Поле с адресом клиента представлено в виде строки формата *nvarchar(40)*. Поле с должностью работника представлено в формате *nvarchar*(50), а его образование – *nvarchar*(200). Описание мебели представлено в виде строки формата *nvarchar*(200), вид материала – *nvarchar*(60). Количество изделий представлено в виде числа *int*. Статус выполнения заказа из-за особенностей *СУБД MS SQL* представлено в виде числа *int*. Скидка для клиента представлена в формате *int*. Дата поступления материалов представлена форматом *datetime*.

Для минимализации возможности возникновения ошибки при операциях с данными всем полям запрещено хранить *null* значения.

Нормализации данных подвергаются все таблицы. Вторая нормальная форма предполагает, что каждый столбец, не являющийся ключом, должен зависеть от первичного ключа. Так как данные о работниках, клиентах и изделиях не зависят от заказа, то данные о работниках были вынесены в отдельную таблицу «*Employees*», данные об изделиях – в таблицу «*Furniture*», данные о клиентах – «*Clients*», а таблица «*Orders*» содержит внешние ключи записей, соответствующие обслуживающему данный заказ работнику, заказанному изделию и клиенту, сделавшему заказ. Так как данные об изделиях и работниках не зависят от накладной, то данные о них были вынесены в отдельные таблицы «*Employees*» и «*Furniture*», а таблица «*Waybills*» содержит внешние ключи записей, соответствующие принимавшему материалы работнику и изделию, для которого предназначены материалы.

Схема созданной базы данных представлена на рисунке 2.1.

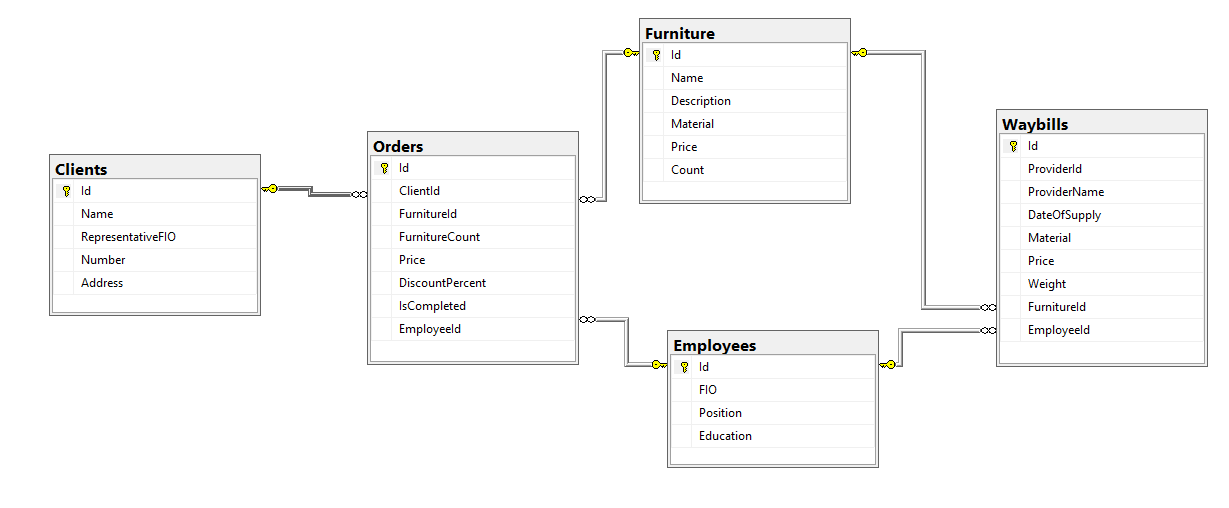


Рисунок 2.1 – Схема базы данных

1. **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

1 Общие сведения

Данное руководство описывает набор необходимых знаний пользователя, особенности работы приложения и минимальные программные требования.

Целью разработки приложения является упрощение доступа к базе данных предприятия в сети Интернет. В результате получилось приложение, способное обрабатывать данные и выполнять функции, установленные в задаче к курсовой работе.

Пользователь системы должен обладать таким минимальным набором знаний:

– пользовательские навыки работы с ЭВМ;

– знание предметной области и знакомство с руководством пользователя.

Список документов, предоставляемых к обязательному ознакомлению перед началом работы:

– настоящее руководство пользователя;

– инструкциями по технике безопасности (для работы с ЭВМ).

**2 Требования к запуску приложения**

Для запуска приложения требуется:

– операционная система *Windows* версией *Windows* 7 или выше;

– устнановленный *.NET Core* и наличие всех необходимых библиотек;

– установленный *MS SQL Server*, а также наличие необходимой базы данных.

**3 Запуск приложения**

Приложение можно запустить через *exe*-файл напрямую или же через среду программирования. После запуске приложения пользователь может сразу же продолжить работу без использования какой-либо роли. Для завершения программы необходимо закрыть страницу сайта.

**4 Непредвиденные ситуации**

Так как формы ограничивают ввод неверных данных и предупреждает пользователя об ошибках, а неавторизованный доступ к функциям пресекается, ошибки могут возникнуть с базой данных, а также с оформлением сайта и расположением элементов. Это такие ошибки, как:

– потеря соединения с базой данных;

– доступ к данному ресурсу недоступен;

* потеря соединения с сервером;
* отсутствие элементов формы в связи с неправильной загрузкой таблиц стилей или элементов страницы;
* отсутствие реакции кнопок форм на нажатие в связи с отсутствием скриптов.

При возникновении любых непредвиденных ситуаций рекомендуется проверить соединение с базой данных и наличие всех таблиц и полей в базе данных, а также наличие всех необходимых таблиц стилей и скриптов. Лучше всего будет обратится к системному программисту. Если после повторного использования программы возникают ошибки, сообщите разработчику или программисту, который в данный момент поддерживает программу для её устранения.

1. **РУКОВОДСТВО ПРОГРАММИСТА**

**1 Назначение и условия применения программы**

Приложение предназначено для работы с базами данных.

Основные функции приложения:

– предоставление сведений из таблиц базы данных в удобном для пользователя виде;

– осуществление авторизации и аутентификации;

– добавление данных в таблицы авторизованными пользователями;

– удаление данных из таблиц авторизованными пользователями;

* изменение данных в таблицах авторизованными пользователями;

– управление ролями.

Для запуска приложения должен быть установлен *.NET Core*. Для считывания данных устройство должно соединятся с базой данных *MS SQL Server*. Для корректного отображения информации необходимо наличие всех таблиц стилей, скриптов и изображений интерфейса.

**2 Характеристики программы**

Приложение написано в среде разработки *Visual Studio 2019*, с помощью технологии *.NET Core* на языке *C#*. База данных поддерживается с помощью СУБД *MS SQL Server*, а доступ к данным осуществляется с помощью технологии *Entity Framework Core*. Графический интерфейс и работа в браузере реализована с помощью технологии *ASP.NET Core MVC*.

**3 Обращение к программе**

Обращение к программе происходит при запуске .*exe* файла или в среде разработки и настройки соединения с СУБД *MS SQL Server*.

Для дополнения программного обеспечения новым функционалом можно использовать любую среду разработки на языке программирования *C#*. Понятная иерархия позволяют модифицировать приложение без опасности нарушить основной функционал программы. Так как используется технология *Entity Framework Core* для связи с базой данных, не рекомендуется использовать другую СУБД.

**4 Входные и выходные данные**

Для добавления входных данных необходимо через приложение использовать формы для управления данными в базе данных, расположенные на страницах таблиц БД.

Выходными данными для приложения являются отображаемые пользователю таблицы в представлении с актуальными данными.

**5 Сообщение в ходе работы приложения**

При работе программа может оповещать пользователя о следующих неполадках в работы системы:

– некорректные учетные данные пользователя;

– несуществующий пользователь;

– ошибки корректности ввода данных;

– остутствие необходимых изображений;

– потеря соединения с базой данных;

– доступ к данному ресурсу недоступен;

– остутствие таблиц или полей.

Оповещения о неполадках с базой данных производится путем сообщений в специальном виде ошибки с описанием. Остальные же ошибки отображаются в приложении в виде сообщения с предупреждением и остановкой выполнения некорректной функции без завершения работы приложения.

ЗАКЛЮЧЕНИЕ

Список используемых источников

ПРИЛОЖЕНИЕ А

(обязательное)

**Иерархическая схема классов**

ПРИЛОЖЕНИЕ Б

(обязательное)

**Код программы**