**Содержание**

[**Введение** 3](#_Toc90574344)

[1. Аналитический обзор литературы 6](#_Toc90574345)

[2. Архитектура базы данных 9](#_Toc90574346)

[2.1 Таблиц базы данных 9](#_Toc90574347)

[2.3 Индекс базы данных 11](#_Toc90574348)

[3. Реализация задач 12](#_Toc90574349)

[3.1. Управление базой отелей 12](#_Toc90574350)

[3.2. Реализация нескольких ролей 13](#_Toc90574351)

[3.3. Осуществление бронирования 13](#_Toc90574352)

[3.4. Изменение пароля юзера 14](#_Toc90574353)

[3.5. Просмотр забронированных номеров и отмена бронирования. 14](#_Toc90574354)

[4. Импорт и экспорт данных. 16](#_Toc90574355)

[5. Тестирование производительности 18](#_Toc90574356)

[6. Описание технологии 19](#_Toc90574357)

[7. Тестирование 25](#_Toc90574358)

[**Заключение** 28](#_Toc90574359)

[**Список использованных источников** 29](#_Toc90574360)

[**ПРИЛОЖЕНИЕ A** 30](#_Toc90574361)

[**ПРИЛОЖЕНИЕ Б** 32](#_Toc90574362)

# **Введение**

Бронирование — это предварительный заказ мест и номеров. С этого процесса начинается обслуживание гостей в отеле. Бронированием номеров занимаются сотрудники отдела бронирования и службы приема и размещения. Именно в эти подразделения поступают заявки на бронирование от клиентов. Кроме формального сбора заявок отдел бронирования должен изучать спрос на гостиницу. Изучая многолетний опыт работы отелей, учитывая план мероприятий, предусматриваемых в регионе, прогнозируя спрос на места и анализируя загрузку номеров в прошлые и настоящий периоды, отдел бронирования совместно со службой маркетинга планирует деятельность гостиницы.

Способ бронирования через Интернет открывает широкие возможности для гостиниц и клиентов. При бронировании через Интернет заказ мест может быть осуществлено в гостиницах, расположенных в другом городе или даже в другой стране. Преимуществом автоматизированной сети является то, что бронирование может быть переадресовано в любой другой отель, находящийся в этом городе

Есть еще два варианта бронирования с помощью компьютера: некоторые отели имеют собственную Интернет-страницу или являются членами одной из систем Интернет-бронирования (Академсервис, WEB International, Nota Bene и т.п.). Системы становятся все более популярными из-за очень широких возможностей при достаточно небольших затратах. Эффективность этого способа привлечения клиентов тем выше, чем меньше время ответа на полученный заказ. Скорость отправки подтверждения во многом зависит от степени автоматизации процесса обработки заявок и от четкости взаимодействия отеля и агента.

Грамотная организация обмена информацией между двумя этими звеньями позволяет сократить время получения клиентом подтверждения до минимума (в идеале так называемый on-line должно занимать не более семи секунд).

Программное средство «База отелей» реализует работу с отелями и их номерами. «База отелей» имеет несколько режимов доступа: пользователь и администратор. Функциями приложения являются: ведение базы данных отелей, просматривание информации об отелях и номерах, бронирование и разбронирование номеров, просмотр своего аккаунта и изменение пароля (функции бронирования, просмотра аккаунта и изменения пароля только у пользователя), есть возможность добавлять новые отели, редактировать текущие отели, удалять отели с их номерами (добавлять, удалять и редактировать отели может только администратор).

Базой данных называется набор логически связанных данных, который предназначен для удовлетворения информационных потребностей организации/предприятия. Для выполнения задания, поставленного в курсовой работе, необходимо освоить работу по проектированию баз данных и работу с базами данных в выбранной среде программирования.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс — Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом.

В качестве интерфейса программирования приложений был выбран Windows Forms, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом — С#, разработанный для создания приложений на платформе Microsoft .NET Framework.

Кроме того для хранения данных был использован расширяемый язык разметки XML.

Из множества всех существующих СУБД, таких как MySQL, MSSQL, PostgreSQL, MongoDB и другие, в курсовом проекте было принято решение использовать MSSQL, а именно Microsoft SQL Server 2019.

Хоть курсовой проект и является лишь прототипов реального модуля «Hotels Base», однако конфиденциальные данные пользователя всё ещё остаются в приоритете. Именно поэтому пароли пользователей хешируются.

В нашей информационной системе требуется обеспечение бесперебойного доступа к БД невзирая на всевозможные сбои и отказы оборудования и программ. Задача придать системе "высокую степень доступности" имеет единственное решение в ИТ, а вместо этого имеет гамму разных решений, каждое со своими решениями и ограничениями. Многие из таких решений реализованы и в MSSQL. Одно из самых доступных - организация "горячего резерва". В MSSQL для обозначения горячего резерва используется термин «standby». Именно поэтому в качестве технологии было использовано с создание резервного standby-сервера.

**Реферат**

Курсовая работа изложена на 31 страницах, которые включают в себя 7 разделов, 14 подразделов, введение, архитектуру, реализацию, руководство пользователя, тестирование, заключение, используемую литературу и приложения.

В курсовой работе разработана модель базы данных, которая состоит из 5 таблиц. В базу данных включено 23 процедур, 1 индекса.

Используемая СУБД – Microsoft Sql Server.

Изучаемая технология – резервный standby сервер.

Основной целью курсовой работы является проектирование модели БД и применение технологии резервный standby сервер.

Course work is set out on 31 pages, which includes 7 sections, 14 subsections, introduction, architecture, implementation, user manual, testing, conclusion, literature used and applications.

In the course work, a database model has been developed, which consists of 5 tables. The database includes 23 procedures, 2 indexes.

The DBMS used is Microsoft Sql Server.

The technology under study is a standby server.

The main purpose of the course work is the design of the database model and the use of technology standby server.

## Аналитический обзор литературы

На сегодняшний день на рынке представлено большое разнообразие систем, автоматизирующих процессы принятия и обработки бронирования отелей. Которые индивиду­ально настраиваются под потребности определенного клиента. Данные решения являются платными. В свою очередь, это затрудняет анализ и обзор подобных прототипов. Несмотря на это, я постарался в полном объеме проанализировать найденные мной решения.

Для обзора были выбраны наиболее популярные системы:

1. «booking» – система бронирования жилья.

Как указывает производитель, данное программное средство нацелено на поиск жилья любого типа. Также указаны стоимость проживания, рейтинг, качество номера.

Интерфейс «booking» представлен на рисунке 1.1.

|  |
| --- |
|  |

Рисунок 1.1 - Интерфейс «booking.com»

К вышеперечисленному, решение предоставляет возможность детального заказа авиабилетов, аренду машин, заказ такси, построение маршрута, сопровождающеюся разного рода фильтрами и предоставляемыми услугами для наилучшего выбора номера.

1. «Hotelhunter» – Сравнение цен на отели.

Производитель говорит что данное программное средство позволяет найти эксклюзивные скидки напрямую от отелей и мировых систем бронирования. А также поможет Нам найти все спецпредложения и сэкономить до 80%.

Интерфейс «Hotelhunter» представлен на рисунке 1.2.

|  |
| --- |
|  |

Рисунок 1.2 - Интерфейс «Hotelhunter»

Помимо выше представленного рисунка, данное решение удивляет обилием интерфейсов для различного типа пользователей. Благодаря которым реализован наилучший поиск отелей, просмотра различных отелей.

1. «holiday» – система автоматизации турфирм.

Производитель говорит, что данное программное средство позволяет организовать оперативную организацию тура с выбором турфирмы, забронировать отель.

Интерфейс «holiday» представлен на рисунке 1.3

|  |
| --- |
|  |

Рисунок 1.3 – Интерфейс «holiday»

Производитель также дает возможность помочь в организации свадьбы за городом, покупки сувениров. Не смотря на это данное решение имеет недостаток в виде не отображения фотографий, также предоставляя неточную информацию об отелях.

Проанализировав существующие на рынке решения, был сделан следующий вывод:

Несмотря на обилие представленных средств, которые позволяют максимально автоматизировать процессы выбора отеля и бронирование номера. Стоит учесть, что данные предложения зачастую предоставляют неверную информацию об услугах отелей. Также встречаются платные решения, которые не подходят для многих пользователей.

## Архитектура базы данных

2.1 Таблиц базы данных

Для хранения и предоставления данных пользователю, была разработана база данных, которая содержит в себе 5 таблиц, все связаны друг с другом внешними ключами. Диаграмма базы данных представлена в Приложении Б.

* Таблица Users содержит всю информацию о пользователях, а именно: идентификатор пользователя (UserId int pk), логин пользователя (Username nvarchar(64)), имя пользователя (FirstName nvarchar(64)), фамилия пользователя(LastName nvarchar(64)), пароль пользователя (Password nvarchar(128)), роль пользователя (Role int fk).
* Таблица Role содержит всю информацию о ролях пользователей, а именно: идентификатор роли (RoleId int pk), название роли (RoleName nvarchar(64)).
* Таблица Comments содержит всю информацию о комментариях пользователей, а именно: идентификатор комментария (CommentId int pk), идентификатор пользователя (UserId int fk), комментарий (Comment nvarchar(300)).
* Таблица Apartments содержит всю информацию о номерах, а именно: идентификатор номера (ApartmentId int pk), номер номера (Room int), идентификатор пользователя (UserId int fk), название отеля (HotelName nvarchar (64) fk), статус бронирования (Booked int).
* Таблица Hotels содержит всю информацию о отелях, а именно: идентификатор отеля (HotelId int pk), название отеля (Name), количество номеров (Rooms int), вместимость отеля (Capacity int).2.2 Процедуры для решения поставленных задач
* Процедура selectUser позволяет получать всех пользователей.
* Процедура selectUser позволяет выполнять поиск пользователя по его никнейму.
* Процедура selectUserId позволяет выполнять поиск пользователя по его идентификатору.
* Процедура addUser позволяет выполнять добавление пользователя в соответствии с входными параметрами: никнейм, имя, фамилия, пароль. По умолчанию пользователь получает роль «user».
* Процедура adminRole позволяет менять роль пользователя «user» на роль «admin».
* Процедура deleteUsers позволяет удалять пользователя по его идентификатору.
* Процедура updateUser позволяет обновлять данные о пользователе в соответствии с входными параметрами: никнейм, имя, фамилия, пароль.
* Процедура loginUser возвращает пользователя, если входные параметры (никнейм и пароль) соответствуют данным из таблицы Users.
* Процедура updateUserPass позволяет менять пароль пользователя.
* Процедура AllApartments позволяет получать все номера.
* Процедура AllUserApartment позволяет получать забронированные номера всех пользователей.
* Процедура getApartment позволяет получать забронированный номера пользователя.
* Процедура addApartment позволяет добавлять номер.
* Процедура delApartment позволяет удалять номер по его номеру.
* Процедура resApartment позволяет бронировать номер.
* Процедура resetApartment позволяет разбронировать номер.
* Процедура selectHotels позволяет получать все отели.
* Процедура addHoteзволяет добавлять отель в соответствии с входными параметрами: название, количество номеров, вместимость.
* Процедура delHotel позволяет удалять номер по его идентификатору.
* Процедура updHotel позволяет обновлять отель в соответствии с входными параметрами: имя, количество номеров, вместимость.
* Процедура ExportUsersXML позволяет экспортировать таблицу Users в XML файл.
* Процедура ImportUsersXML позволяет импортировать содержимое файла ImportUser.xml в таблицу Users.

### 2.3 Индекс базы данных

Индекс SQL Server является структурой на диске или в памяти, которая связана с таблицей или представлением и ускоряет получение строк из таблицы или представления. Индекс содержит ключи, построенные из одного или нескольких столбцов в таблице или представлении. Для индексов на диске эти ключи хранятся в виде структуры сбалансированного дерева, которая поддерживает быстрый поиск строк по значениям ключей в SQL Server.

Индексы обеспечивают быстрый доступ к строкам таблиц, сохраняя отсортированные значения указанных столбцов и используя эти отсортированные значения для быстрого нахождения ассоциированных строк таблицы. Индексы позволяют находить строку с определенным значением столбца, просматривая при этом лишь небольшую часть общего объема строк таблицы. Таким образом правильное использование индексов сокращает до минимума количество дорогостоящих операций ввода-вывода.

Использование индекса не всегда означает высокую производительность, а высокая производительность не всегда означает эффективное использование индекса. Если бы использование индекса всегда способствовало производительности, то работа оптимизатора запросов была бы очень простой. На самом деле, неверный выбор индекса может привести к неоптимальной производительности. Следовательно, задача оптимизатора запросов состоит в том, чтобы выбрать индекс или комбинацию индексов, если это улучшит производительность, и избежать индексированного поиска, если это ее понизит.

Ниже приведены используемые индексы:

Индекс IX\_Apartments\_HotelName позволяет получать более быстрый доступ к строкам таблицы Apartments, включая столбец Room.

1. Реализация задач
   1. Управление базой отелей

Управление базой отелей осуществляет администратор. Управление базой отелей включает в себя добавление, удаление, обновление отелей. Ниже приведены процедуры, реализующие управление отелями.

* Процедура для созданию отеля представлена ниже:

Create procedure addHotel

@name nvarchar(64),

@rooms int,

@capacity int,

@hotelId int out

AS

begin

insert into Hotels(Name, Rooms, Capacity)

values(@name, @rooms, @capacity)

SET @hotelId=SCOPE\_IDENTITY()

end

* Процедура для удаления отеля представлена ниже:

go

Create procedure delHotel

@hotelId int

AS

begin

delete from Hotels where HotelId = @hotelId

end

Процедура для обновления отеля представлена ниже:

go

create procedure updHotel

@hotelId int,

@name nvarchar(64),

@rooms int,

@capacity int

as

begin

update Hotels set Name = isNull(@name, Name),

Rooms = isNull(@rooms,Rooms), Capacity = isNull(@capacity,Capacity)

where HotelId = @hotelId

end

* 1. Реализация нескольких ролей

В приложении доступно две роли: роль юзера и роль администратора.

Юзер может просматривать номера отелей, бронировать и разбронировать номера, изменять свои личные данные(пароль).

Администратор может просматривать всю информацию отелей, добавлять, удалять, редактировать отели. Выполнять импорт и экспорт пользователей.

* 1. Осуществление бронирования

Осуществление бронирования производится юзером. Чтобы сделать бронирование ему нужно выбрать понравившийся номер и нажать на кнопку «reserve».

* Процедура для осуществления бронирования представлена ниже:

go

create procedure resApartment

@userId int,

@apartmentId int

AS

begin

update Apartments set UserId = @userId, Booked = 1

where ApartmentId = @apartmentId

end

* 1. Изменение пароля юзера

Чтобы изменить пароль нужно под темой «Change password», ввести пароль в поле для ввода и подтвердить его в соседнем поле для ввода. После этого нажать на кнопку «Save pass». Если введенные данные совпадают пароль юзера будет изменен.

* Процедура для осуществления изменения пароля юзера представлена ниже:

go

Create procedure updateUserPass

@userId int,

@userPass nvarchar(128)

as

begin

Update Users set Password = @userPass where UserId = @userId

end

3.5. Просмотр забронированных номеров и отмена бронирования.

Просмотр забронированных номеров производится юзером. Чтобы осуществить просмотр нужно нажать на кнопку «Reservation», после нажатия откроется соответствующее окно. Где будут показаны все забронированные номера пользователя. Чтобы отменить бронированние номера нужно выбрать соответствующий номер и нажать на кнопку «de-reserve».

* Процедура для получения забронированных номеров представлена ниже:

go

create procedure getApartment

@userId int

as

begin

Select ApartmentId, Room, HotelName from Apartments where UserId = @userId

End

* Процедура для получения отмены бронирования номера представлена ниже:

go

create procedure resetApartment

@apartmentId int

AS

begin

update Apartments set UserId = NULL, Booked = 2

where ApartmentId = @apartmentId

end

1. Импорт и экспорт данных.

Реализация импорта и экспорта данных находится в самом приложении, написанного на С#. Импортировать и экспортировать данные может только администратор. Для этого на форме администратора есть специальная вкладка «import/export users», на которой есть 2 кнопки: «Import» и «Export», соответственно при нажатии на «Import» происходит загрузка данных пользователей в файл XML, при нажатии на «Export» происходит сохранение данных о пользователях в файл XML.

* Процедура импорта представлена ниже:

go

create procedure ImportUsersXML

AS

begin

insert into Users (Username, FirstName, LastName,Password, Role)

select

usXML.[User].query('Username').value('.', 'nvarchar(64)'),

usXML.[User].query('FirstName').value('.', 'nvarchar(64)'),

usXML.[User].query('LastName').value('.', 'nvarchar(64)'),

usXML.[User].query('Password').value('.', 'nvarchar(128)'),

usXML.[User].query('Role').value('.', 'int')

from (select cast(usXML as xml)

from openrowset(bulk 'D:\Laboratory\CourseWork\HotelsBase\XML\ImportUser.xml', single\_blob) as T(usXML)) as T(usXML)

cross apply usXML.nodes('Users/User') as usXML ([User]);

end

* Процедура экспорта представлена ниже:

go

create or alter procedure ExportUsersXML

AS

begin

Select \*from Users

for xml path('User'), root('Users')

EXEC sp\_configure 'show advanced options', 1

RECONFIGURE WITH OVERRIDE

EXEC sp\_configure 'xp\_cmdshell', 1

RECONFIGURE WITH OVERRIDE

DECLARE @fileName varchar(100)

DECLARE @sqlStr varchar(1000)

DECLARE @sqlCmd varchar(1000)

SET @fileName = 'D:\Laboratory\CourseWork\HotelsBase\XML\ExportUsers.xml'

SET @sqlStr = 'USE HotelsBase; select \* from Users for xml path(''User''), root(''Users'') '

SET @sqlCmd = 'bcp "' + @sqlStr + '" queryout ' + @fileName + ' -w -T'

EXEC xp\_cmdshell @sqlCmd;

End

1. Тестирование производительности

* Процедура вставки 100000 строк представлена ниже:

go

Create procedure Ins100000rows

as

begin

declare @i int = 1

while @i < 100001

begin

set @i = @i+1

insert into dbo.Apartments(Room, HotelName) values

(@i, 'Moscow');

end

end;

Данная процедура вставляет в таблицу Apartments строки с фиксированным значением. Время выполнения – примерно 14 секунд.

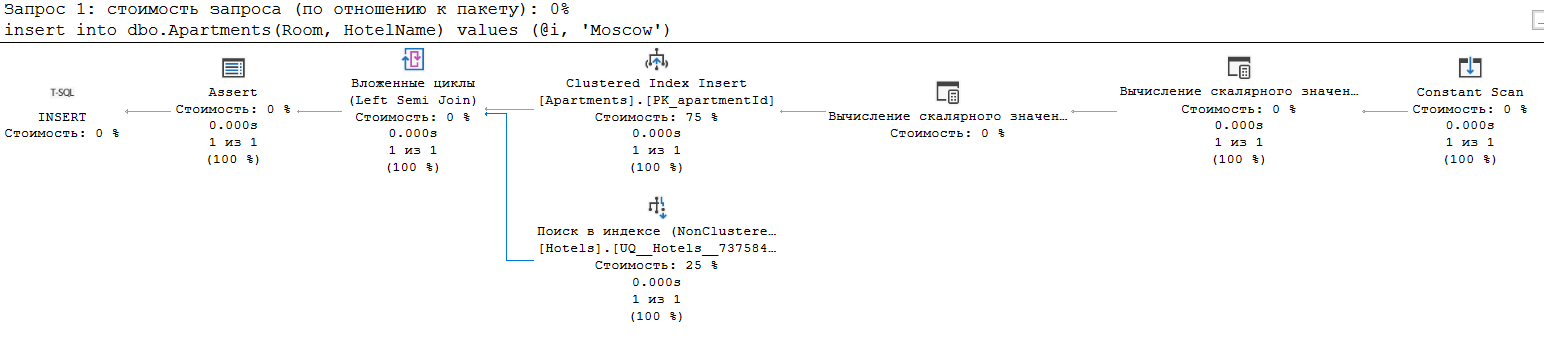


Рисунок 5.1 – Тестирование производительности

1. Описание технологии

Представим себе ситуацию, когда наш проект, использующий в качестве СУБД MSSQL, неожиданно (или с надеждой ожидаемо) стал критически важным для бизнеса (соответственно, появилась готовность выделять средства на обеспечение надежности системы).

До этого момента мы вполне обходились ежедневным или даже еженедельным бэкапом («горячим» или «холодным» копированием, а может и просто экспортом данных) и нас устраивало время восстановления системы порядка суток (будем считать, что данных у нас на пару терабайт).

И вот оказалось, что на восстановление системы нам отводится не более часа, и никакие данные нам терять нельзя.

Итак, все указывает на то, что нам придется поднимать standby сервер.

Настройка горячей резервной копии (standby) базы данных Microsoft SQL Server выполняется при помощи программного комплекса "BACKUP RENT", который должен быть установлен на сервере (компьютере), где работает MS SQL Server.

Для организации standby базы данных MS SQL Server необходимо запустить Мастер настройки резервирования (кнопка на панели инструментов) в программе Конфигуратор (меню Пуск → Все программы → Backup Rent → Конфигуратор).

Шаг 1

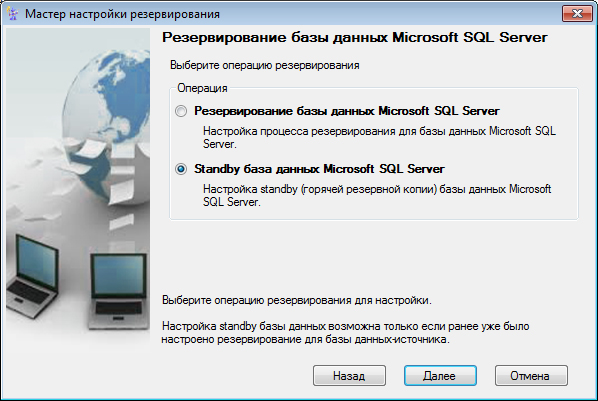


Рисунок 6.1 – Выбор резервирования

Выбираем Standby база данных MSSQL и жмем далее

Шаг 2

Выбираем компьютер для резервирования.

Шаг 3

Выбираем из списка каталог, куда выполняется резервирование исходной базы данных Microsoft SQL Server средствами программы BACKUP RENT.

Шаг 4

В поле Пароль для расшифровки резервных копий указываем пароль, который используется для шифрования и защиты от несанкционированного доступа резервных копий. В случае если для резервных копий не используется шифрование, оставляем данное поле пустым.

Шаг 5

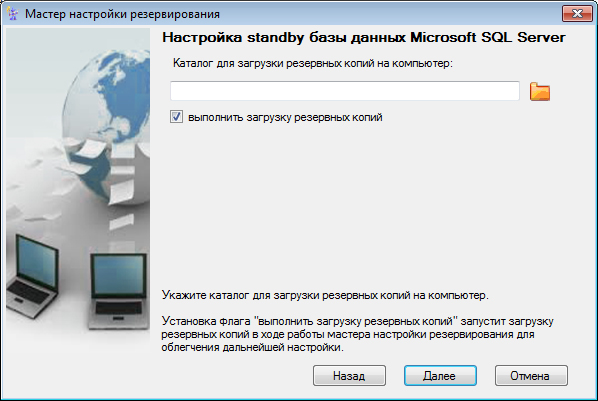


Рисунок 6.2 – Выбор каталога резервных копий

Указываем локальный каталог, который будет использоваться для загрузки резервных копий из хранилища BACKUPRENT.

Для облегчения процедуры настройки установите флажок Выполнить загрузку резервных копий. Установка этого флага разрешает Мастеру настройки резервирования выполнить загрузку резервных копий в момент своей работы для автоматического определения параметров standby базы данных.

Шаг 6

Укажите имя исходной базы данных для создаваемой standby базы данных и нажмите кнопку Далее.

Шаг 7

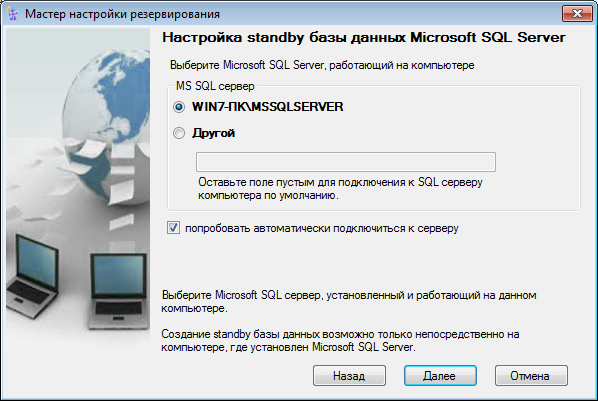


Рисунок 6.3 – Выбор сервера MSSQL

Выберите Microsoft SQL Server который будет обслуживать создаваемую standby базу данных. Если в списке Мастера резервирования отсутствует требуемый MS SQL Server, то можно ввести его имя вручную выбрав пункт "Другой".

Шаг 8

В случае определения пользователя для подключения к серверу вручную необходимо обратить внимание на права пользователя – пользователь должен быть администратором базы данных или как минимум иметь роль DB\_BACKUPOPERATOR.

Шаг 9

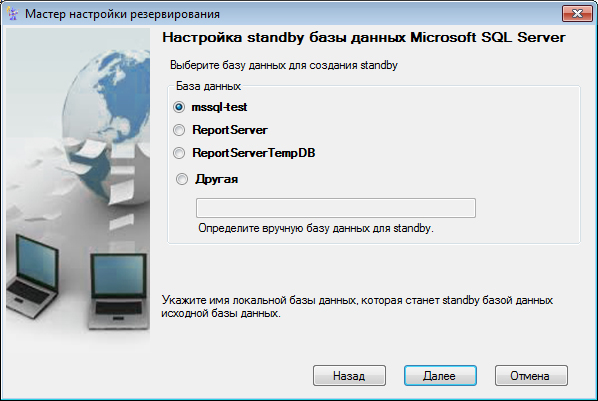


Рисунок 6.4 – Выбор БД

Укажите имя для создаваемой локально на компьютере standby базы данных. В случае если необходимо создать standby базу данных с именем, которого нет в списке, то выберите пункт "Другая" и задайте вручную имя для standby базы данных.

Шаг 10

Задайте расположение файлов базы данных на локальном диске компьютера и нажмите кнопку Далее.

ВНИМАНИЕ

Standby база данных должна содержать как минимум один файл для данных и один файл для журнала транзакций.

Шаг 11

Задайте периодичность попыток загрузки резервных копий исходной базы данных в standby базу данных.

ВНИМАНИЕ

Период синхронизации с хранилищем для загрузки крайних резервных копий должен соотноситься с настройками периодичности резервирования журнала исходной базы данных Microsoft SQL сервер.

Шаг 12

Задайте периодичность попыток загрузки резервных копий исходной базы данных в standby базу данных.

ВНИМАНИЕ

Период синхронизации с хранилищем для загрузки крайних резервных копий должен соотноситься с настройками периодичности резервирования журнала исходной базы данных Microsoft SQL сервер.

Шаг 13

Укажите имя для создаваемой задачи горячего резервирования (standby) базы данных Microsoft SQL сервер

Шаг 14

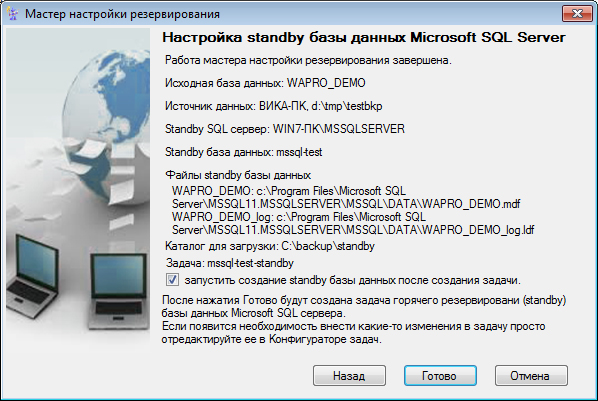


Рисунок 6.5 – Проверка и создание standby

Мастер настройки резервирования еще раз выводит для проверки все ключевые параметры создаваемой задачи. Если все отображенные на этом этапе сведения верны, то после нажатия кнопки Готово будет создана задача, реализующая создание standby базы данных Microsoft SQL сервер.

1. Тестирование

Тестирование программного обеспечения (Software Testing) - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. Для предотвращения возникновения исключительных ситуаций, которые препятствуют нормальной работе пользователя, их необходимо обработать. Функциональное тестирование проводится для проверки качества программного продукта, а точнее того, какие функции реализованы и насколько верно это сделано. Иными словами, это способ узнать, сможет ли разработанное ПО при определённых условиях выполнять необходимые задачи. В моём приложении обрабатываются базовые ошибки.

Попытка логина с неверными данными:

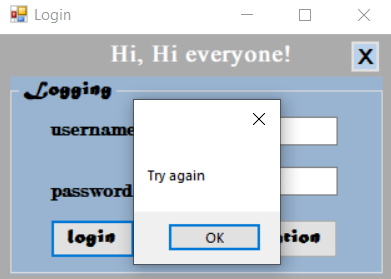


Рис 6.1 – Пользователь ввел неверные данные

Попытка регистрации уже существующего пользователя:

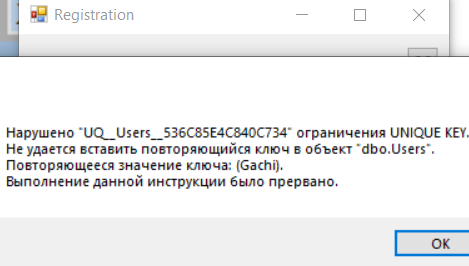


Рис 6.2 – Существующий пользователь

Попытка неправильного подтверждения пароля:

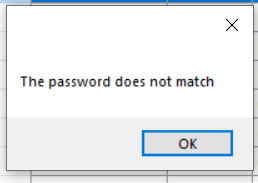


Рис 6.3 – Пароли не совпадают

Попытка добавления уже существующего отеля:

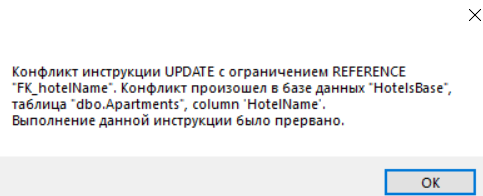


Рис 6.4 – Существующий отель

Попытка обновления имени на имя существующего отеля:

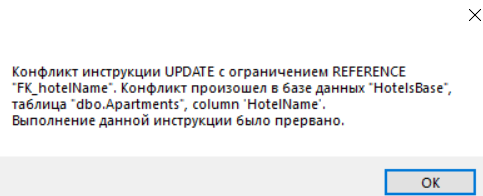


Рис 6.5 – Существующий отель

Попытка загрузки несуществующего файла:

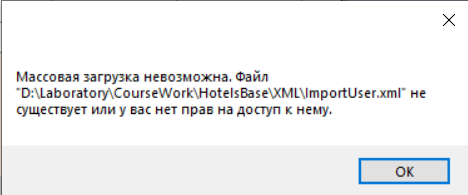


Рис 6.6 – Ошибка с файлом импорта

Попытка загрузки уже существующих пользователей:

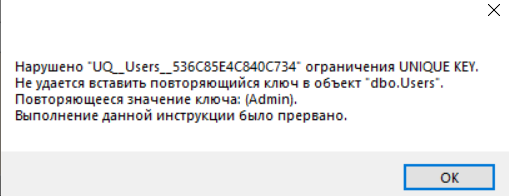


Рис 6.7 – Ошибка с файлом импорта

# **Заключение**

В ходе курсовой работы основные задачи были выполнены: была спроектирована база данных для базы отелей используя СУБД MSSQL Server 2019. Приложение было разработано с использованием технологии standby сервер. Основные требования, которые были поставлены к работе, были выполнены:

* Был реализован ряд процедур, благодаря которым в итоговом приложении поддерживается функция бронирования номеров. Все введённые и/или выбранные пользователем номера (в данном случае именуемым юзером) принимаются в хранимые процедуры в качестве параметров и далее обрабатываются;
* Была реализована возможность операций с отелями: добавление, удаление и обновление.
* Реализован импорт данных из XML файла и экспорт данных в XML файл.
* Реализация нескольких ролей представляет из себя создание двух пользователей: администратора и юзера. В зависимости от роли пользователя, ему предлагаются те или иные функции;
* В приложении была реализована технология создания standby сервера.
* База данных была протестирована на производительность.

# **Список использованных источников**

1. Интернет-портал [Электронный ресурс] /<https://metanit.com>. – Режим доступа: <https://metanit.com/sharp/windowsforms/1.1.php>. – Дата доступа: 28.11.2021.
2. Пацей, Н.В. Курс лекций по языку программирования C# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с. – Дата доступа: 28.10.2021.
3. Блинова Е.А. Курс лекций по базам данных / Е.А. Блинова – Дата доступа: 17.11.2021
4. Microsoft Справочник по Transact-SQL [Электронный ресурс]/ docs.microsoft.com– Режим доступа: https://docs.microsoft.com/ru-ru/tsql/language-reference?view=sql-server-ver15 – Дата доступа 02.12.2021
5. Интернет-портал [Электронный ресурс] / [https://metanit.com](https://vk.com/away.php?to=https%3A%2F%2Fmetanit.com&cc_key=) – Руководство по ADO.NET и работе с базами данных: https://metanit.com/sharp/adonet/ – Дата доступа: 10.10.2021

# **ПРИЛОЖЕНИЕ A**

Файл для импорта данных пользователей:

<Users>

<User>

<Username>Admin</Username>

<FirstName>Daniil</FirstName>

<LastName>Alsevich</LastName>

<Password>46/tAEewgFnQ+toQ9ADB5Q==</Password>

<Role>1</Role>

</User>

<User>

<Username>Admin2</Username>

<FirstName>Mark</FirstName>

<LastName>Shagal</LastName>

<Password>Ie7U8umrIU/b8AoqCR1jxA==</Password>

<Role>1</Role>

</User>

<User>

<Username>Gachi</Username>

<FirstName>Billy</FirstName>

<LastName>Heringhton</LastName>

<Password>iyb/uT4DWW5lFOlghfZA2Q==</Password>

<Role>2</Role>

</User>

<User>

<Username>Capital</Username>

<FirstName>Karl</FirstName>

<LastName>Marks</LastName>

<Password>xWzMahMh5m3dlP5DC4ofgg==</Password>

<Role>2</Role>

</User>

<User>

<Username>StarWars</Username>

<FirstName>Obi-van</FirstName>

<LastName>Kenobi</LastName>

<Password>WvKQSkxz9NaeOyl0YkQMFQ==</Password>

<Role>2</Role>

</User>

</Users>

# **ПРИЛОЖЕНИЕ Б**

|  |
| --- |
|  |