

База данных включает 4 таблицы.

1. Таблица students содержит поля с данными о студентах.
  - a. Id – первичный ключ
  - b. Name – имя
  - c. Middlename – отчество
  - d. Surname – фамилия
  - e. Birthday – дата рождения
  - f. Phonenumber – номер телефона
  - g. Course – номер курса
  - h. Studentgroup – группа
2. Таблица rooms содержит поля с данными о комнате
  - a. Id – первичный ключ
  - b. Numberofrooms – номер комнаты
  - c. Seatscount – количество мест в комнате
3. Таблица studentsrooms строит отношение между таблицами rooms students и содержит внешние ключи на эти таблицы.
  - a. Id – первичный ключ
  - b. Room\_id – внешний ключ таблицы комнаты
  - c. Student\_id – внешний ключ таблицы студенты
4. Таблица users содержит информацию для формы авторизации
  - a. Id – первичный ключ
  - b. Login – логин
  - c. Password – пароль
  - d. Role – роль, которая принадлежит записи

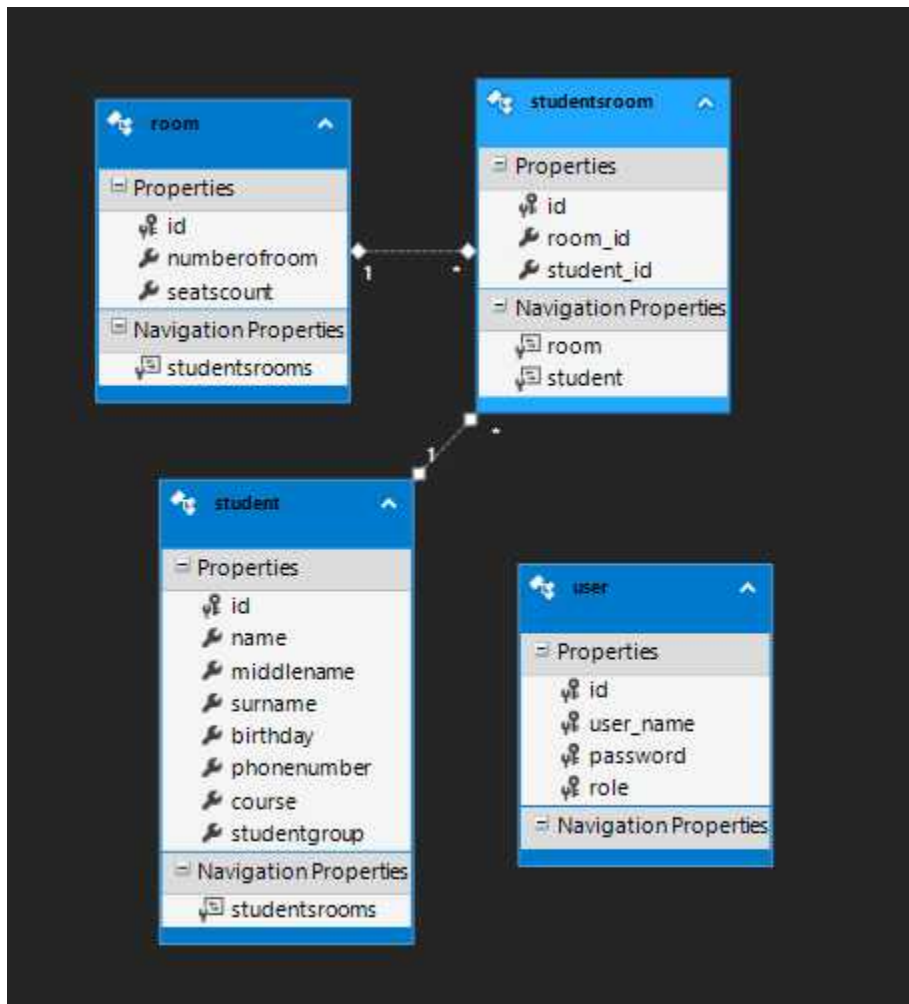


Диаграмма таблиц, со связями.

Для того, чтобы разграничить логику приложения от интерфейса управления, был выбран шаблон для построения приложений MVC (model-view-controller).

- Model – модель, база данных приложения
- View – представление, графический интерфейс (windows forms)
- Controller – контроллер, логика приложения, посредник между моделью и представлением

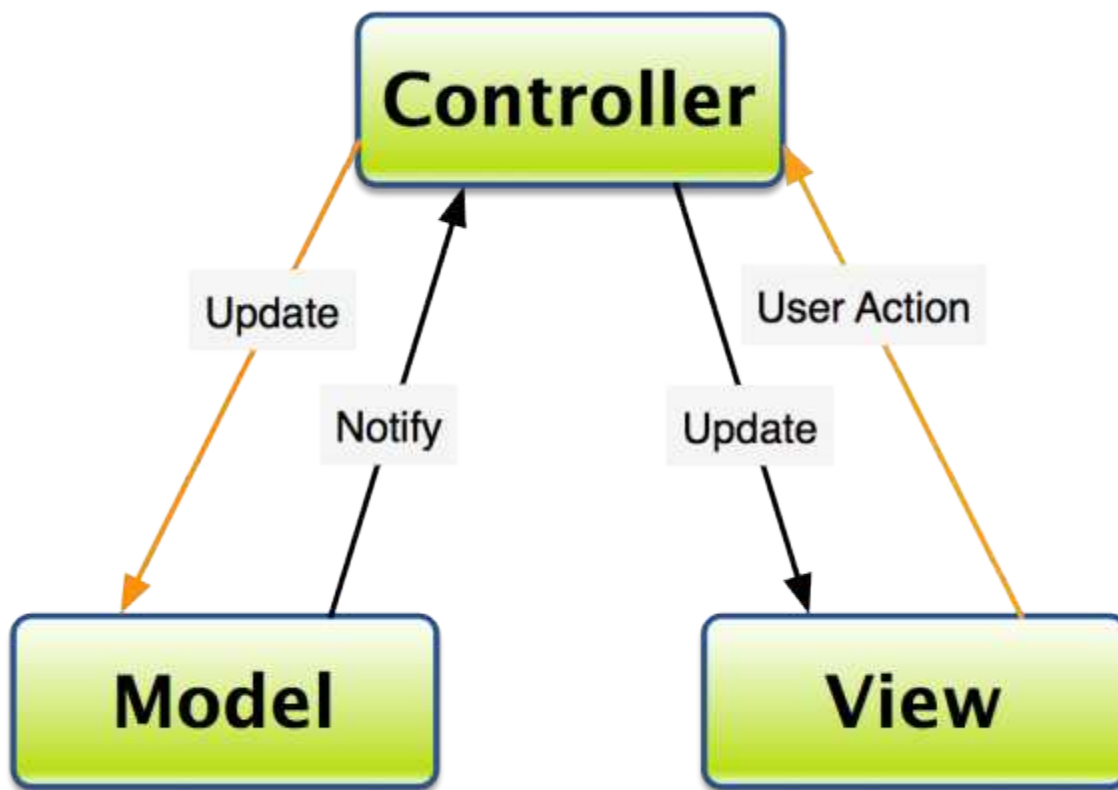
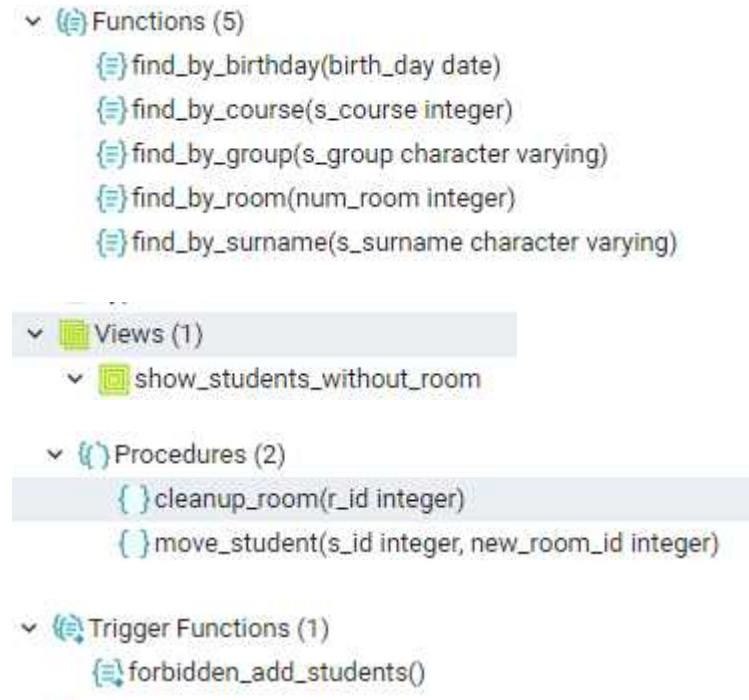


Диаграмма шаблона MVC.

**Представление** ничего не знает ни о контроллере ни о модели, предоставляя только открытые свойства для контроллера, чтобы тот в свою очередь мог получить доступ к элементам управления представления (controls), также включает в себя легкую валидацию и управление дочерними контролами. **Контроллер** хранит в себе ссылки на модель и представление, чтобы обеспечить взаимодействие между ними. **Моделью** является база данных, в приложение модель была сгенерирована с помощью **EntityFramework(Database First)** по существующей базе данных. **EntityFramework** упрощает работу с БД, все базовые операции (insert, delete, update, read) выполняются с помощью методов. Используя такой подход, код приложения получается более чистым, если сравнивать с тем случаем, когда запросы к базе расположены в явном виде.

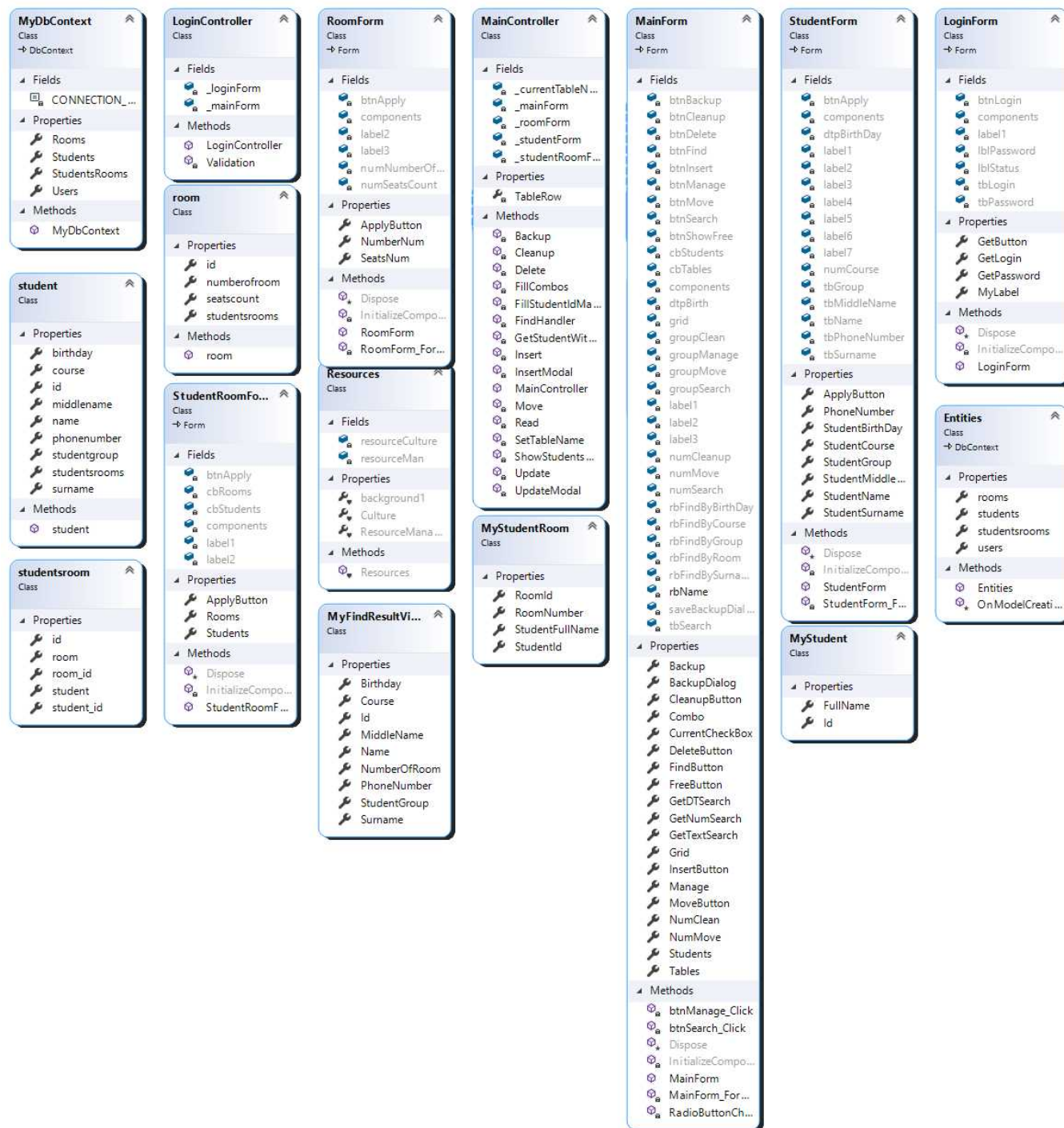
База данных для приложения была выбрана - **PostgreSQL 13.2**, чтобы минимизировать использование сырых запросов внутри приложения, вся логика по поиску, обработке и отображению информации из БД в

**DataGridView** приложения была организована внутри БД, с помощью Function, Storage Procedure, View, Trigger (хранимые процедуры, представления, триггеры, функции). В самом приложении, используется только непосредственный вызов этих процедур по именам.



(можно привести sql запросы из дампа для них)

С помощью встроенных средств Visual Studio ниже отображена диаграмма классов, которая включает в себя все классы приложения и элементы (свойства, поля, методы).

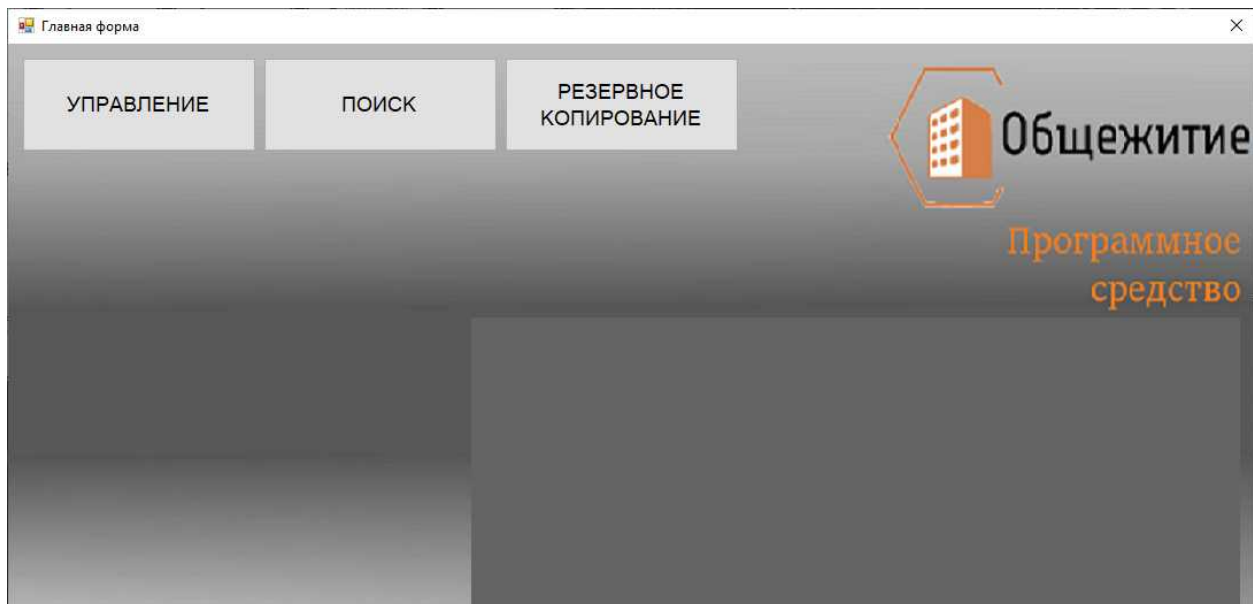


Пароли хранятся в хешированном виде, с использованием так называемой «соли», при которой одинаковые пароли имеют разную хеш сумму. Как можно видеть на скриншоте ниже, хеш сумма разная, для одинакового пароля **pass**

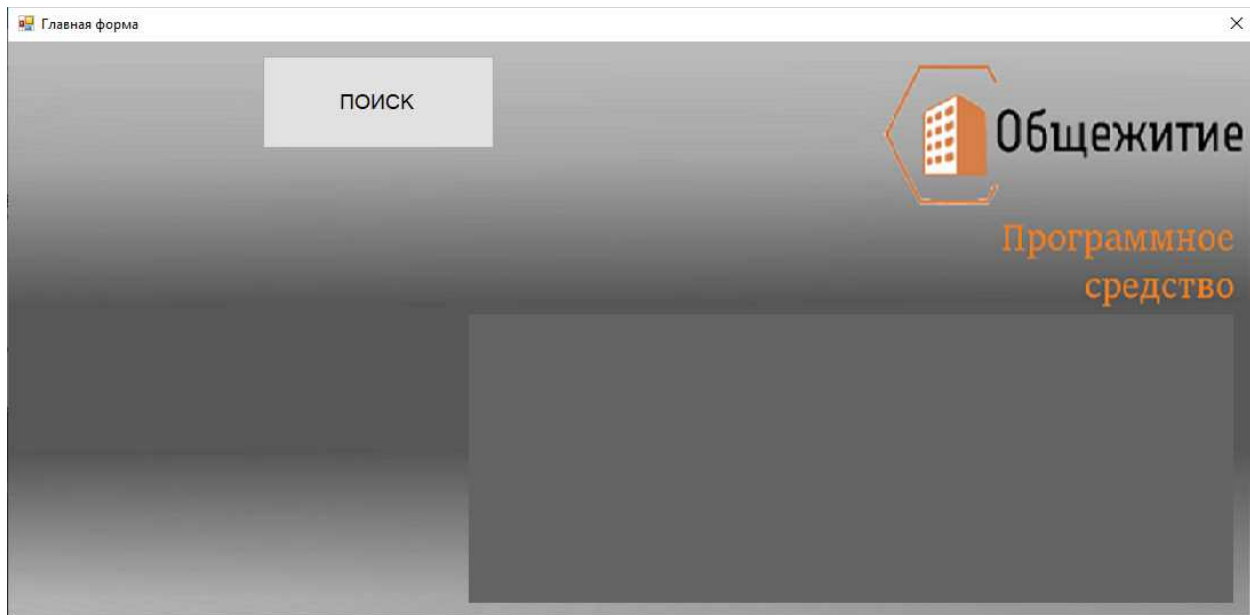
	id integer	user_name character varying	password character varying	role character v
1	1	user	\$HASH V1\$10000\$ALYikowWy5kHJLKszkNt8Bnsj4cheTJV8/0yIIQkTLyeYFsS	user
2	2	root	\$HASH V1\$10000\$lebAoR/cDyt1uQs16JIE0w5BI2pYrF9TPU+bS1678eN1ZGQo	admin

Для хеширования, используется специальный алгоритм, который повторно прогоняет полученную хеш сумму, на заданное число итераций, что усложняет подбор хеша перебором.

Также используется простейшее разграничение по ролям, приложение имеет две роли user и admin. User не имеет возможности делать бекап и редактировать базу, ему доступен только поиск, admin может все.



Главная форма для пользователя с ролью admin



Вид главной формы для пользователя с ролью user.

Чтобы не использовать жестко заданные параметры подключения к БД в программном коде, строка подключения была вынесена в специальный конфигурационный xml файл App.config, данные для создания backup базы данных, также извлекаются из него.

```
<connectionStrings>
  <add name="StudentResidenceEntities"
        connectionString="metadata=res://*/BL.Model.StudentResidence.csdl|
res://*/BL.Model.StudentResidence.ssdl|res://*/BL.Model.StudentResidence.msl;
        provider=Npgsql;
        provider connection string="
Host=localhost;
Database=StudentResidence;
Username=tomilin;
Password=tomilin";
        providerName="System.Data.EntityClient" />
</connectionStrings>
```

Пример строки подключения из конфигурационного файла.

Все пакеты, которые не входят в стандартную библиотеку .NET Framework были получены с помощью интегрированного пакетного менеджера NuGet





Browse

Installed

Updates10

NuGet Package Manager: StudentResidence


Search (Ctrl+L)




☐ Include prerelease

Package source: 

nuget.org






EntityFramework

by Microsoft

v6.4.0

Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with ma...

v6.4.4




EntityFramework6.Npgsql

by Shay I

v6.4.1

PostgreSQL provider for Entity Framework 6





Microsoft.Bcl.AsyncInterfaces


by M

v1.1.0

Provides the IAsyncEnumerable<T> and IAsyncDisposable interfaces and helper typ...

v5.0.0






Npgsql

by Shay Rojansky, Yoh Deadfall, Bra

v5.0.3

Npgsql is the open source .NET data provider for PostgreSQL.

v5.0.4




System Buffers


by Microsoft

v4.5.0

Provides resource pooling of any type for performance-critical applications that allo...

v4.5.1






System.Memory


by Microsoft

v4.5.3

Provides types for efficient representation and pooling of managed, stack, and native...

v4.5.4






System.Numerics.Vectors

by Micros

v4.5.0

Provides hardware-accelerated numeric types, suitable for high-performance proce...





Browse

Installed

Updates 10

NuGet Package Manager: StudentResidence

Search (Ctrl+L)



Include prerelease

Package source: nuget.org



**System.Numerics.Vectors** by Micros v4.5.0  
Provides hardware-accelerated numeric types, suitable for high-performance proce...



**System.Runtime.CompilerServices** v4.6.0  
Provides the System.Runtime.CompilerServices.Unsafe...



**System.Text.Encodings.Web** by Mic v4.6.0  
Provides types for encoding and escaping strings for use in JavaScript, HyperText Mar...



**System.Text.Json** by Microsoft v4.6.0  
Provides high-performance and low-allocating types that serialize objects to Jav...



**System.Threading.Channels** by Mic v4.7.0  
Provides types for passing data between producers and consumers.



**System.Threading.Tasks.Extensions** v4.5.3  
Provides additional types that simplify the work of writing concurrent and asynchron...



**System.ValueTuple** by Microsoft v4.5.0  
Provides the System.ValueTuple structs, which implement the underlying types for...