База данных включает 4 таблицы.

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

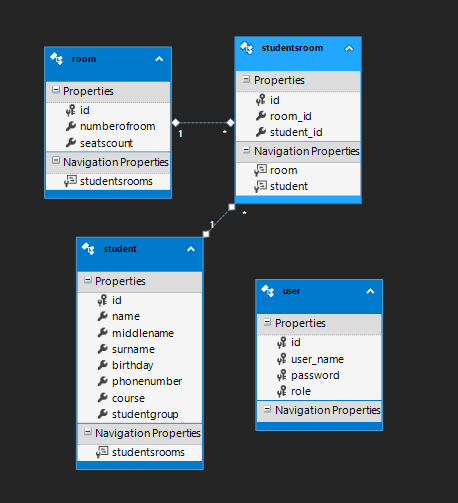


Диаграмма таблиц, со связями.

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

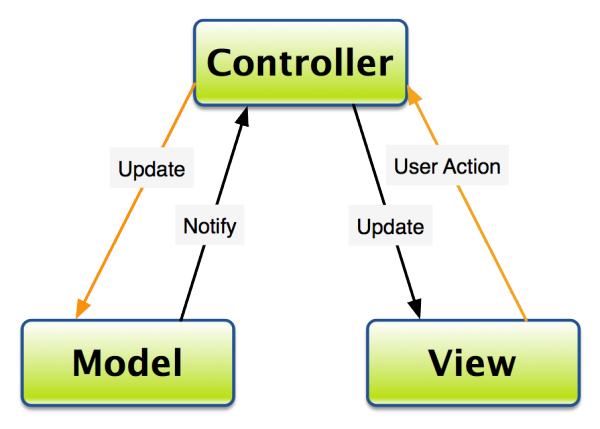
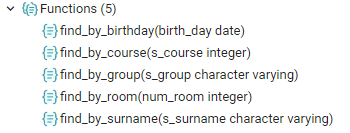


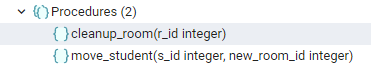
Диаграмма шаблона **MVC**.

**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 чистом, если сравнивать с тем случаем, когда запросы к базе расположены в явном виде.

База данных для приложения была вабрана - **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,StorageProcedure, View, Trigger (хранимые процедуры, представления, триггеры, функции). В самом приложение, используется только непосредственный вызов этих процедур по именам.



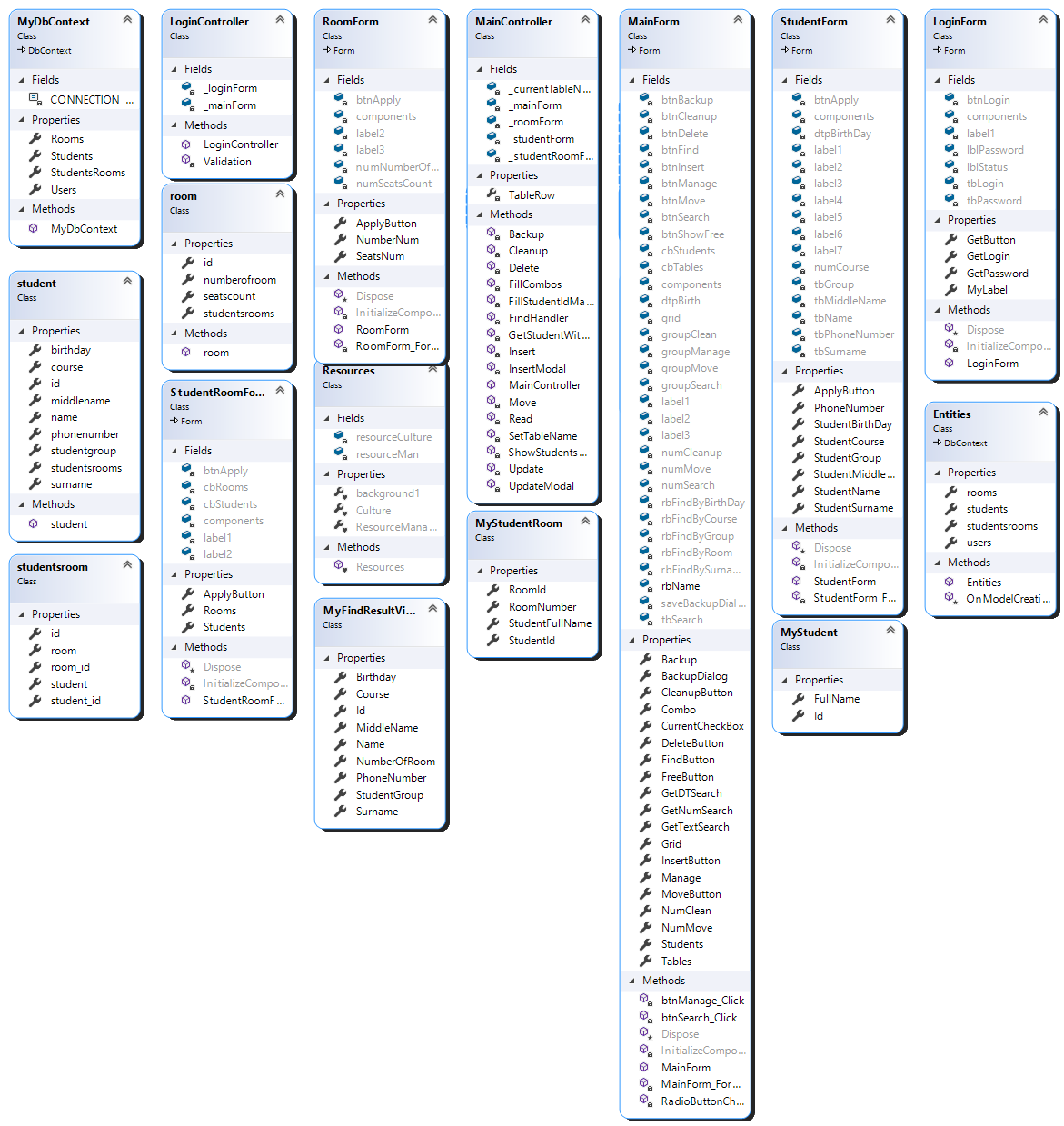




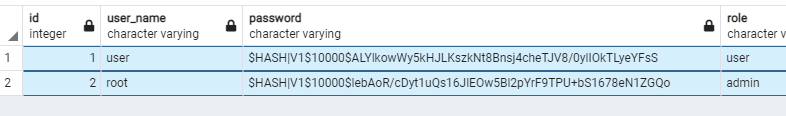


(можно привести sql запросы из дампа для них)

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

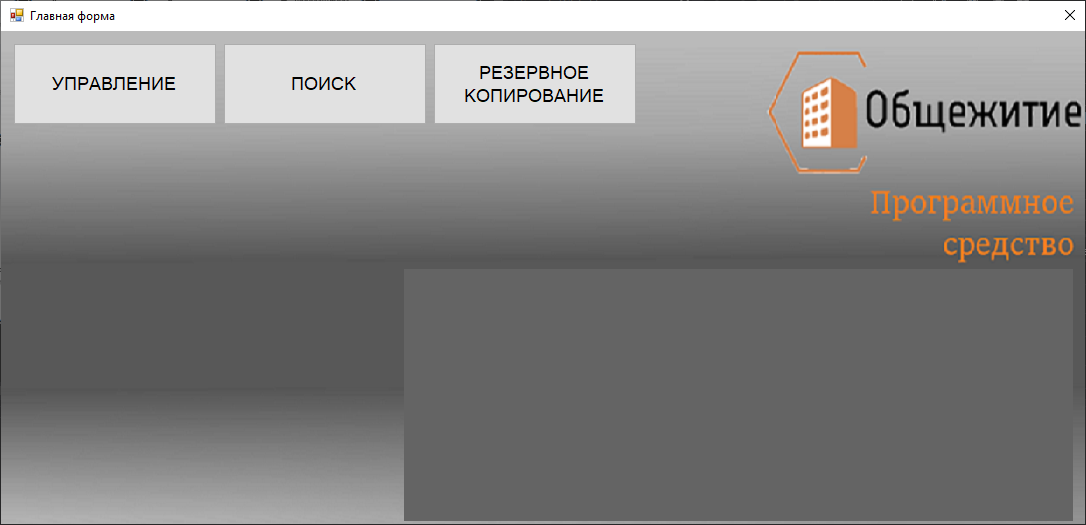


Пароли хранятся в хешированном виде, с использованием так называемой «соли», при которой одинаковые пароли имеют разную хеш сумму. Как можно видеть на скриншоте ниже, хеш сумма разная, для одинакового пароля ***pass***

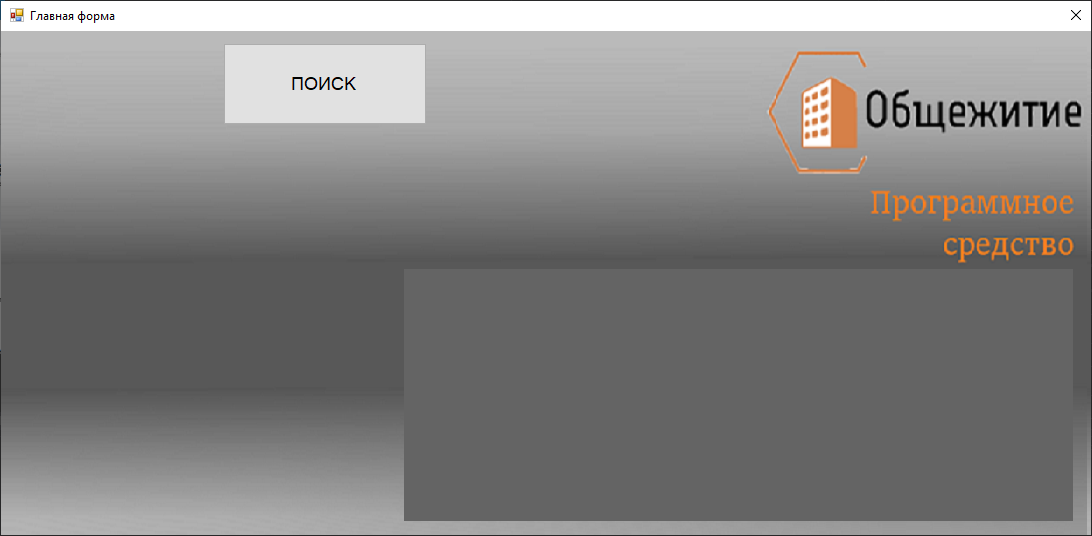


Для хеширования, используется специальный алгоритм, который повторно прогоняет полученную хеш сумму, на заданное число итераций, что усложняет подбор хеша перебором.

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=&quot;

Host=localhost;

Database=StudentResidence;

Username=tomilin;

Password=tomilin&quot;"

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

