

## I. Използвани технологии

Използвани са най-новите технологии за 2020 година: .NET Core 3, Swagger, ASP.NET Web API Core, Entity Framework Core, MS SQL Server, xUnit Tests. Приложението симулира управлението на Сървъри и прикачени към тях устройства.

## 2. Пускане на приложението

- Създайте база данни наречена FMI.Gateways.
- Отворете приложението и напишете Update-Database във конзолата за да създадете базата данни на локалния сървър.
- Натиснете Ctrl + F5 за да пуснете приложението и да видите наличните операции.

## 3. Тестване на операциите

Отворете началната страница и използвайте бутоните и формите за да изпълнявате команди към сървъра и променяте данните в базата данни.



## Съвременен Стандарт за архитектура на АПИ

*Здравейте!  
Дипломната работа  
демонстрира  
стандартна  
архитектура за  
писането на уеб  
приложения в наши  
дни. Уебсайтовете  
създадени със  
JavaScript рамки  
станаха много  
популярни и  
различни, но  
повечето пъти  
сървърната част е  
написана по един и  
същи начин, както  
е демонстрирано.*

Таньо Иванов

...

Фак. Номер:

...

Дата:

...

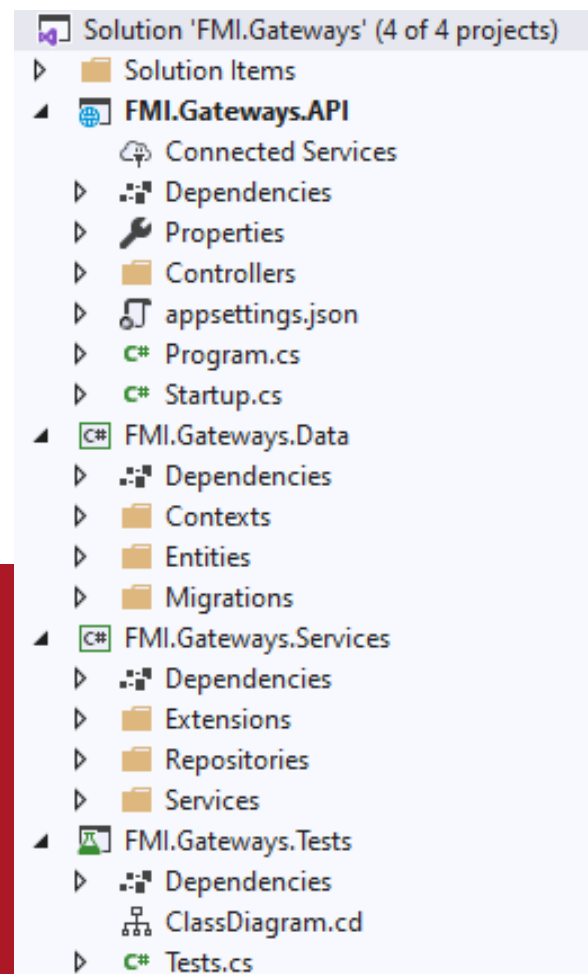
Имейл:

...

ФМИ Пловдив

# ДИПЛОМНА РАБОТА

Таньо Иванов



#### 4. Архитектура: Service

Когато искаме да изпълним операция със даден елемент, използваме Service клас за да можем да създаваме, изтриваме, редактираме и селектираме от базата данни. Например ако искаме да Изтрием устройство можем да изпишем примерно: `new PeripheralDeviceService().Remove(1)`, за да го изтрием.

#### Разпределяне на отговорностите

В системата е използвана методологията за разпределянето на отговорностите както следва:

- Repositories – Отговарят за извършването на първичните операции
- Services – Отговарят за допълнителна логика и операции. Бизнес логика.

## Архитектура на приложението Repository Pattern

Голяма част от днешните приложения ползват модела за дизайн наречен Repository. В него отговорностите по управление на елементите в приложението се разпределят на различни компоненти. Съществува йерархията: Service -> Repository -> DTO -> Entity -> Database Table.



#### 5. Repository

Това ниво от приложението повечето пъти веднъж написано, не се променя. Причината, е че то е съставено от 4те основни операции, които се изпълняват директно върху таблиците, а именно: Създаване, Селектиране, Редактиране и Изтриване. CRUD. Тези операции са написани в чист вид, без ограничения.

#### Gateways

GET /Get

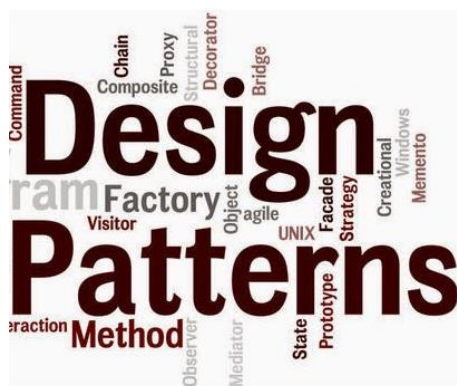
GET /{id}

#### PeripheralDevice

DELETE /{id}

POST /

*„Добрия програмист е този, който се оглежда и в двете посоки, преди да пресече еднопосочен път.“*



#### 6. Entity and Table

Базата данни съдържа таблици със данни. За да можем да ги използваме във кода ние създаваме класове във приложението, които са 1 към 1 като структура със съответните таблици. Тези класове наричаме Entity и използваме за да достъпваме полетата на таблиците и самите тях.

Entity Framework

