НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський Політехнічний Інститут імені Ігоря Сікорського»

Кафедра Технічної кібернетики

Віртуальний асистент “sloBo”

Курсова робота

3 дисципліни «Основи програмування»

Керівник

ас. Тимофєєва Ю.С.

Виконавець

ст. Гавриленко Я.С

зал. книжка № ІТ-0402

«Допущений до захисту» гр. ІТ-04

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(особистий підпис керівника) (особистий підпис виконавця)

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2018р. «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2018р.

Захищений з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оцінка)

Київ – 2018

**Зміст**

1. **Опис завдання**
   1. Про що ваш продукт?
   2. Яку проблему він вирішує?
   3. Функції продукту
   4. Технології
   5. Архітектура
   6. Етапи реалізації
2. **Архітектура підпроектів**
   1. Модуль комунікації з базою даних
   2. Модуль контролю та інтерфейсу користувача
   3. Ієрархія моделей функцій
3. **Архітектура бази даних**
4. **Інструкція з використання**
5. **Результати роботи програми**

**Опис завдання до курсової роботи**

1. **Про що ваш продукт?**

Проект являє собою теоретичний аналог відомих голосових асистентів з рядом функцій, що дозволяють отримувати та передавати інформацію через мережу Internet за допомогою голосових команд.

1. **Яку проблему він вирішує?**

Продукт може пришвидшити пошук інформації за допомогою голосового керування. Також може стати додатком для людей з обмеженими фізичними та розумовими здібностями.

1. **Функції продукту**

Серед функцій продукти зокрема такі:

* Запис, передача та відтворення аудіо-файлів за допомогою сторонньої бібліотеки классів у форматах

1. audio/alaw
2. audio/basic
3. audio/flac
4. audio/g729
5. audio/l16
6. audio/mp3
7. audio/mpeg
8. audio/mulaw
9. audio/ogg
10. Audio/ogg;opus
11. Audio/ogg;vorbis
12. audio/wav
13. audio/webm
14. Audio/webm;opus
15. Audio/webm;vorbis

* Багатофункціональна система команд з можливістю легко додавати нові задачі та макроси для керування головним процесом програми
* Як тестовий матеріал, доступні такі функції

1. Перегляд останніх новин та за заданим у голосовій команді темами
2. Відтворення аудіозаписів, що зберігаються на спеціалізованих сервісах у мережі
3. Огляд погодних умов в околицях пристрою
4. Спеціалізована функція з оглядом ситуації щодо COVID

* Збереження інформації про функції та запити на глобально встановленій базі даних
* Інтерактивне, мінімалістичне графічне меню
* Розпізнавання голосу, можливе в усіх наведених вище форматах

1. **Технології**

* .NET via CS
* NAudio
* Entity Framework
* TSQL via MSSQLServer
* Microsoft Azure
* Docker
* WEB APIs:

1. IBM Watson STT
2. OpenWeatherAPI
3. SpotifyAPI
4. COVID19API
5. NewsAPI
6. **Архітектура**

Архітектура проекта складається з п’яти основних частин:

* Додаткові запозичені та створені самостійно бібліотеки классів
* Інтерфейс користувача
* Інтерфейс бази даних
* Архітектура бази даних
* Модульні тести

Список вбудованих у netstandart бібілотек, які були використані:

* System;
* System.Collections.Generic;
* System.IO;
* System.Linq;
* System.Text;
* System.Threading;
* System.Threading.Tasks;
* System.Runtime.Remoting;
* System.Runtime.InteropServices;
* System.Net.Http;
* System.Diagnostics;

Також використовувалися такі бібліотеки NuGet:

* Newtonsoft.Json;
* NUnit
* NAudio;
* IBM.Watson.SpeechToText.v1;

Окрім цього була створена своя бібліотека DrawRectangle для реалізації інтерфейсу в консолі. Діаграма класів для бібліотеки:

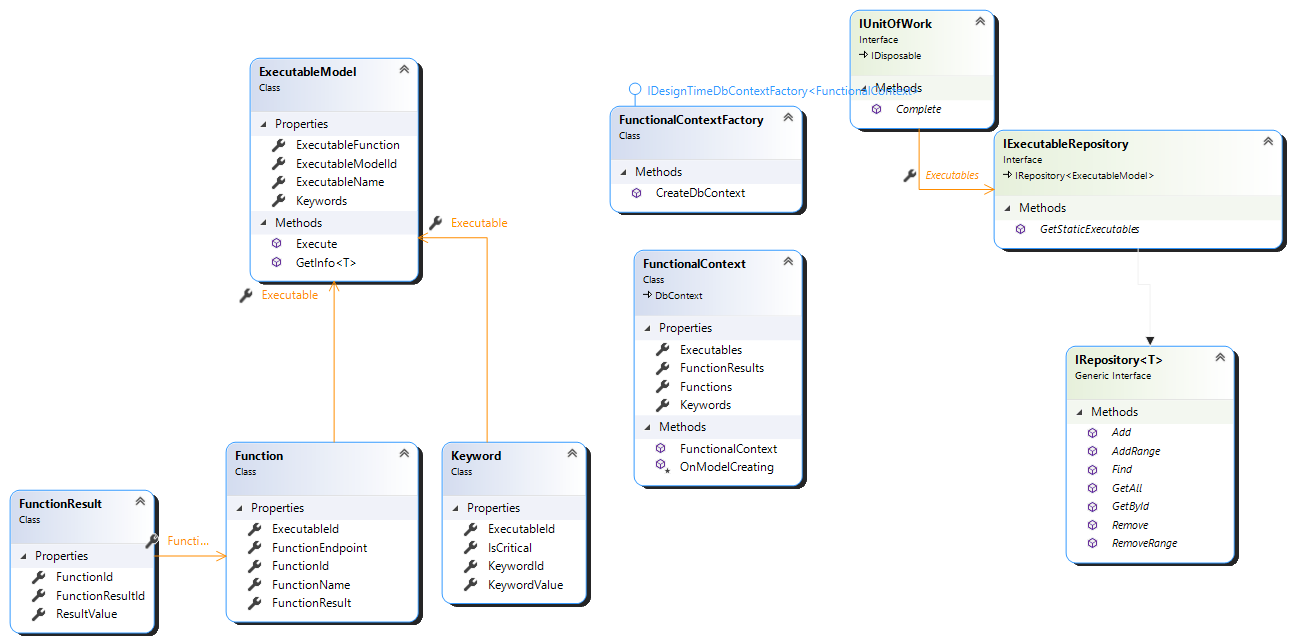
Основна частина програми складається з керування меню, таймінгами перемикання та класів, які представляють функціональний юніт.

1. **Етапи Реалізації**

* Системний аналіз та первинне планування
* Розробка вимог до програмного забезпечення
* Проектування системи
* Розробка програмних компонентів, а саме допоміжних бібліотек, які використовувалися для створення візуальної частини та ієрархії функцій
* Цикли розробки функціоналу за ітераційною моделлю
* Підключення бази даних та перенесення даних, щодо функціоналу, системи підбору та уточнення команд
* Тестування
* Контейнеризація за допомогою Docker

**Архітектура підпроектів**

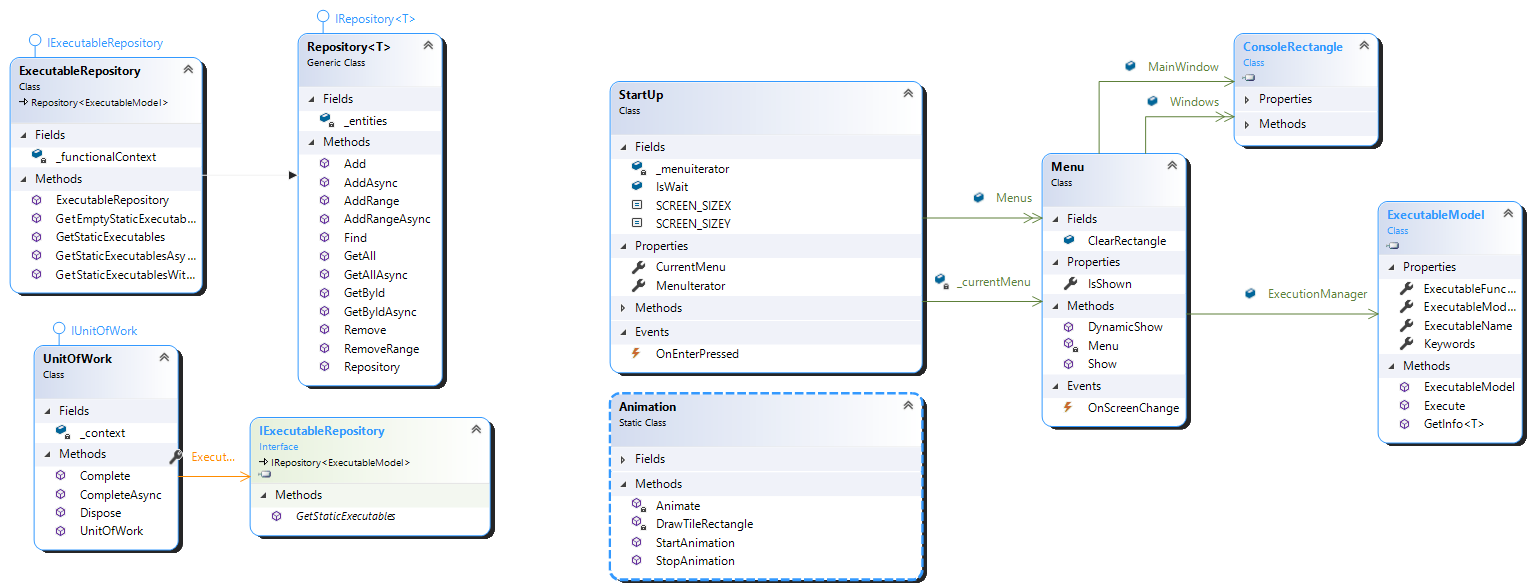
Модуль комунікації з базою даних.



Модуль комунікації з базою даних складається з реалізації патернів Repository та UnitOfWork.

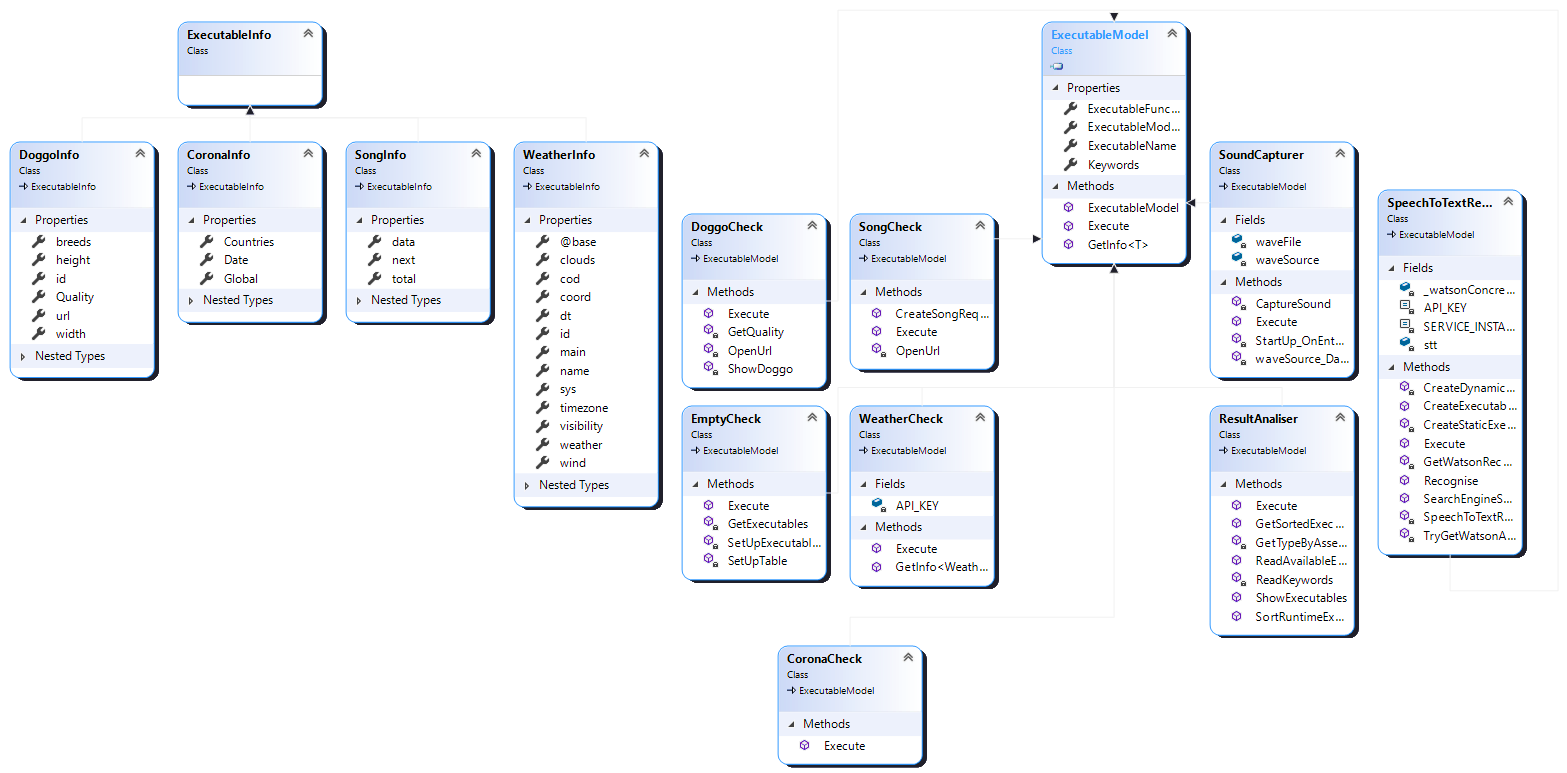
Також для роботи з Entity Framework створено модель бази даних, яка відповідає Entity Relationship діаграмі, зазначеній нижче. Через класс FunctionalContext проходить нижній рівень взаємодії з базою даних у проекті.

Модуль контролю



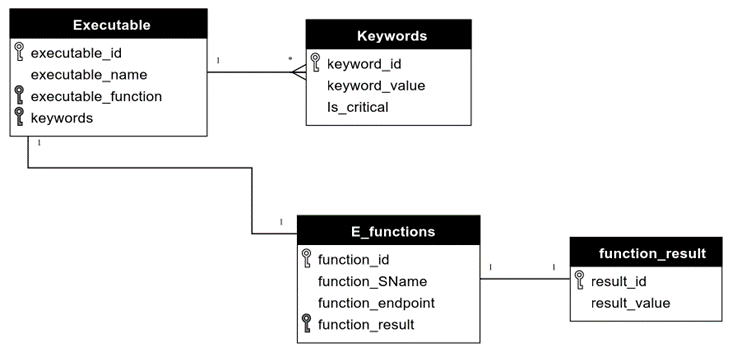
Тут визначено чітку реалізацію інтерфейсів користування базою даних. Класс StartUp є центральним, у якому викликаються тригери переключення меню та збереження деякої критичної інформації про стан системи.В кожен окремий момент роботи програми може бути відображене лише один об’єкт класу Menu, що напряму взаємодіє з бібліотекою для створення користувацького інтерфейсу.

Ієрархія фунцій



Праве ієрархічне дерево відповідає функціям проекту. Так праве піддерево – це функції-меню, а ліве – кінцеві можливості для користувача. Ліве дерево відповідає ієрархії моделей інформації для користувацькиї функцій.

**Архітектура бази даних**



**Інструкція з використання**

Ітерація меню відбувається за допомогою клавіші *Enter.*

Під час перегляду одного з меню, вимкнути додаток можна за допомогою клавіші *Esc.* Також в будь-який момент перервати виконання можливо комбінацією *Ctrl+C.*

Після однократного виконання функцій, можливо одразу зробити наступний запит.

**Результати роботи програми**

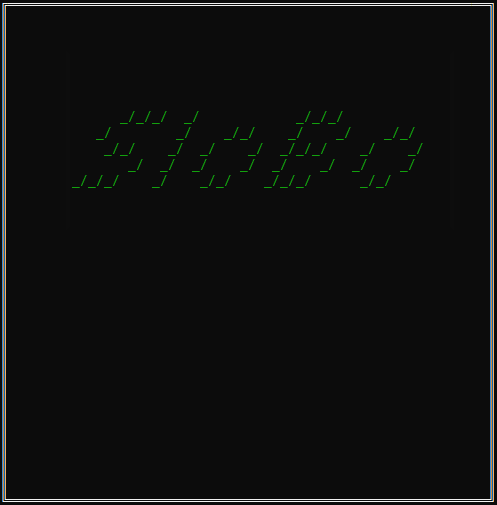


Рис.1 Початковий екран програми

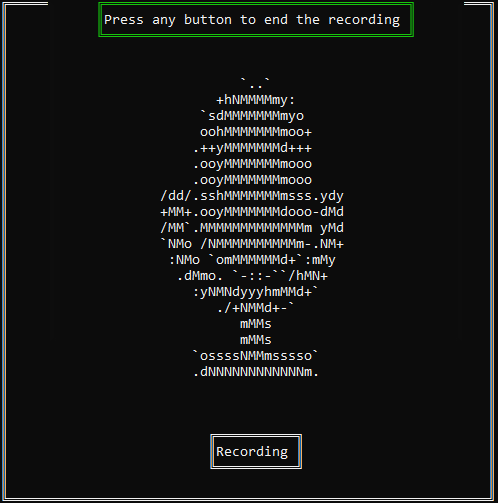


Рис.2 Екран запису звуку

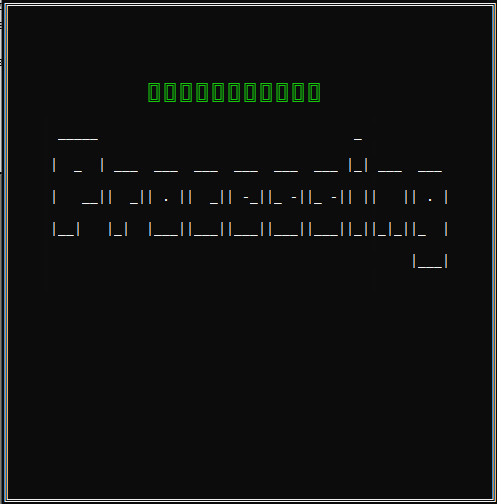


Рис.3 Екран обробки. Кадр анімації 1

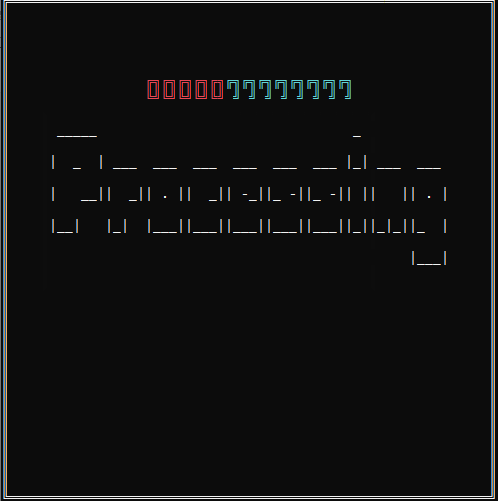


Рис.4 Екран обробки. Кадр анімаціїї 2

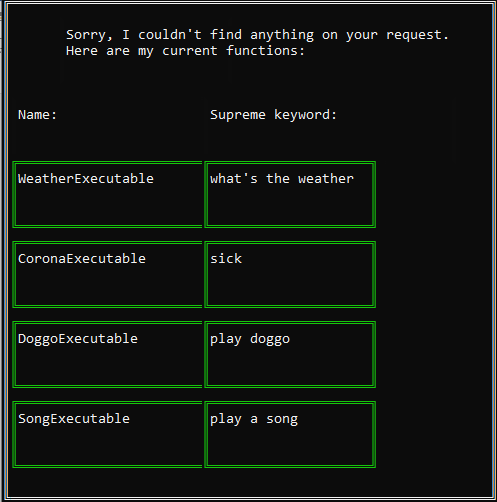


Рис.5 Екран результату 1. Пустий результат

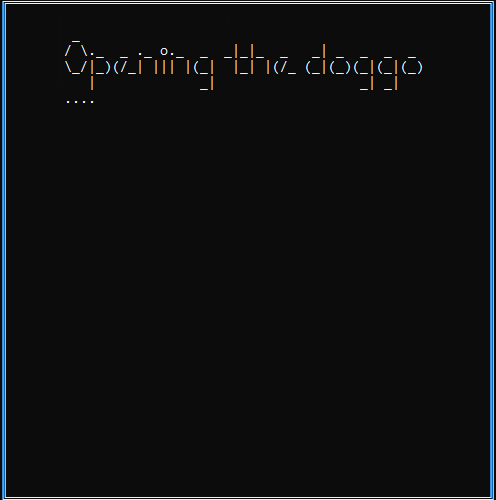


Рис.6 Екран результату 2. Міні-гра “Doggo”



Рис.7 Екран результату 2. Результат міні-гри “Doggo”