

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЕКОНОМІКО-ПРАВОВИЙ ФАКУЛЬТЕТ
КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

**До захисту допустити:
В.о. зав. кафедри**




Ганна МАРТИНЮК

«23» квітня 2024 р.

«РОЗРОБКА IOS ДОДАТКУ “ГОЛОСОВИЙ ПОМІЧНИК” НА БАЗІ

Кваліфікаційна робота
здобувача вищої освіти першого
(бакалаврського) рівня
освітньо-професійної програми
«Комп’ютерні науки»
Корнюка Микити Васильовича
Науковий керівник:
Мнацаканян Марія Сергіївна,
кандидат технічних наук, доцент,
доцент кафедри системного аналізу та
інформаційних технологій
Рецензент:
Лукашенко Вікторія Вікторівна,
кандидат технічних наук, доцент,
заступник декана факультету
комп’ютерних наук та технологій
Національного авіаційного університету

Кваліфікаційна робота захищена
з оцінкою відмінно 100 (А)
Секретар ЕК 

» червня 2024 р.

Київ - 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему « iOS додаток «голосовий помічник» на базі ChatGPT» поєднує в собі наступні розділи: аналітичний, дослідницький та проєктний.

У першому розділі аналізуються сутність поняття та види штучних інтелектів, проводиться аналіз архітектур та механізм робот, досліджуються основні функції та можливості штучних інтелектів, визначаються типи AI за різними класифікаційними ознаками. Надається характеристика видам штучних інтелектів за сферою застосування. Розглядається використання штучних інтелектів в різноманітніших галузях для різних застосувань і цілей, наводяться приклади їх застосування у світі.

У другому розділі досліджується продукти на ринку штучних інтелектів які представленні компанією OpenAI та структура роботи моделі GPT.

У третьому розділі пропонується структура розробки додатку на базі ChatGPT. Описуючи всі кроки для розробки додатку на мові програмування Swift використовуючи новітні технології та гнучку архітектуру додатку.

SUMMARY

The explanatory note to the qualification work on the topic "iOS application "voice assistant" based on ChatGPT" combines the following sections: analytical, research, and design.

The first section analyzes the essence of the concept and types of artificial intelligence, analyzes architectures and robot mechanisms, explores the main functions and capabilities of artificial intelligence, and defines AI types according to various classification criteria. The types of artificial intelligence are characterized by the field of application. The use of artificial intelligence in various industries for different applications and purposes is considered, and examples of its application in the world are given.

The second section examines the products on the artificial intelligence market represented by OpenAI and the structure of the GPT model.

The third section offers a framework for developing an application based on ChatGPT. Describing all the steps to develop an application in the Swift programming language using the latest technologies and flexible application architecture.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1	7
АНАЛІЗ СУТНОСТІ ШТУЧНОГО ІНТЕЛЕКТУ.....	7
1.1 Поняття штучного інтелекту	7
1.2 Принципи роботи штучного інтелекту.....	8
1.3 Слабкий Штучний інтелект	9
1.4 Сильний штучний інтелект.....	10
1.5 Різниця між слабким штучним інтелектом та сильним штучним інтелектом.....	11
1.6 Розробка штучного інтелекту.....	12
Висновок:	13
РОЗДІЛ 2	15
СТРУКТУРА CHATGPT(OPEN AI).....	15
2.1 Моделі Open AI	15
2.2 GPT3.5 та GPT4.....	19
Висновок:	21
РОЗДІЛ 3	22
РОЗРОБКА ЗАСТОСУНКУ iOS.....	22
3.1 Інструменти та паттерни розробки	22
3.2 Розробка інтерфейсу на UIKit та SnapKit.....	24
3.3 Інтеграція з базою даних.....	26
3.4 Інтеграція openAI API в додаток.....	27
Висновок	30

ВСТУП

У сучасному світі технологій, де кожна секунда на рахунку, виникає гостра необхідність в інструментах, здатних максимально ефективно спрощувати й автоматизувати повсякденні завдання. Серед таких інноваційних рішень особливе місце посідають голосові асистенти, які з кожним днем стають дедалі інтелектуальнішими та багатофункціональнішими. У зв'язку з цим розробка голосового помічника на базі одного з найпередовіших штучних інтелектів, такого як ChatGPT, являє собою актуальний і перспективний напрямок у сфері мобільних додатків.

Метою даної дипломної роботи є розробка програмного засобу для операційної системи iOS, що реалізує функції голосового помічника на основі технології ChatGPT. Це завдання передбачає комплексний підхід, що містить аналіз наявних рішень у цій галузі, проєктування архітектури застосунку, розробку користувацького інтерфейсу, інтеграцію з ChatGPT і, нарешті, тестування готового продукту.

Актуальність проєкту зумовлена зростаючим попитом на зручні та функціональні голосові помічники, здатні не тільки виконувати стандартний набір команд, а й навчатися, адаптуватися до потреб користувача, пропонуючи йому найрелевантнішу інформацію та сервіси. Інтеграція з ChatGPT дає змогу значно розширити можливості голосового помічника завдяки використанню передових алгоритмів опрацювання природної мови і машинного навчання, що робить його здатним вести бесіду з користувачем на широке коло тем, аналізувати його запити і пропонувати оптимальні рішення.

У рамках роботи буде проведено аналіз поточного стану ринку мобільних застосунків, визначено ключові вимоги до функціоналу та інтерфейсу голосового помічника, а також розглянуто основні технології та інструменти, які використовуються під час розроблення подібних застосунків.

Особливу увагу буде приділено питанням безпеки та захисту персональних даних користувачів.

Завершальним етапом роботи стане розробка прототипу застосунку "Голосовий помічник" для iOS, що демонструє всі закладені функціональні можливості та інтеграцію з ChatGPT. Результати тестування і користувацької оцінки прототипу дадуть змогу зробити висновки про його ефективність і перспективи подальшого розвитку.

РОЗДІЛ 1

АНАЛІЗ СУТНОСТІ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Поняття штучного інтелекту

Штучний інтелект (ШІ) - це галузь комп'ютерних наук, що займається створенням систем, здатних виконувати завдання, що вимагають людського інтелекту. Це включає в себе сприйняття, міркування, навчання, взаємодію з навколишнім середовищем і вирішення проблем. Штучний інтелект можна розділити на дві основні категорії: слабкий і сильний.

Слабкий ШІ являє собою системи, розроблені та навчені для виконання конкретного завдання без власної свідомості або розуміння. Приклади включають чат-ботів, системи рекомендацій і розпізнавання образів. Ці системи здаються "розумними", але діють у межах обмеженого набору параметрів і завдань.

Сильний ШІ, або генеральний штучний інтелект (ГІІ), відноситься до теоретичної концепції машини, яка має здатність розуміти, мислити і діяти так, що її не відрізнити від людського інтелекту. Сильний ШІ буде здатний до самонавчання і прийняття рішень у невизначених ситуаціях, адаптуючись до нових обставин так само, як це робить людина.

Розробка ШІ вимагає застосування різних підходів, включно з машинним навчанням, глибоким навчанням, нейронними мережами, алгоритмами опрацювання природної мови та багатьма іншими. Сфери застосування ШІ величезні і охоплюють автомобільну промисловість, охорону здоров'я, фінансові послуги, розваги, освіту та багато іншого.

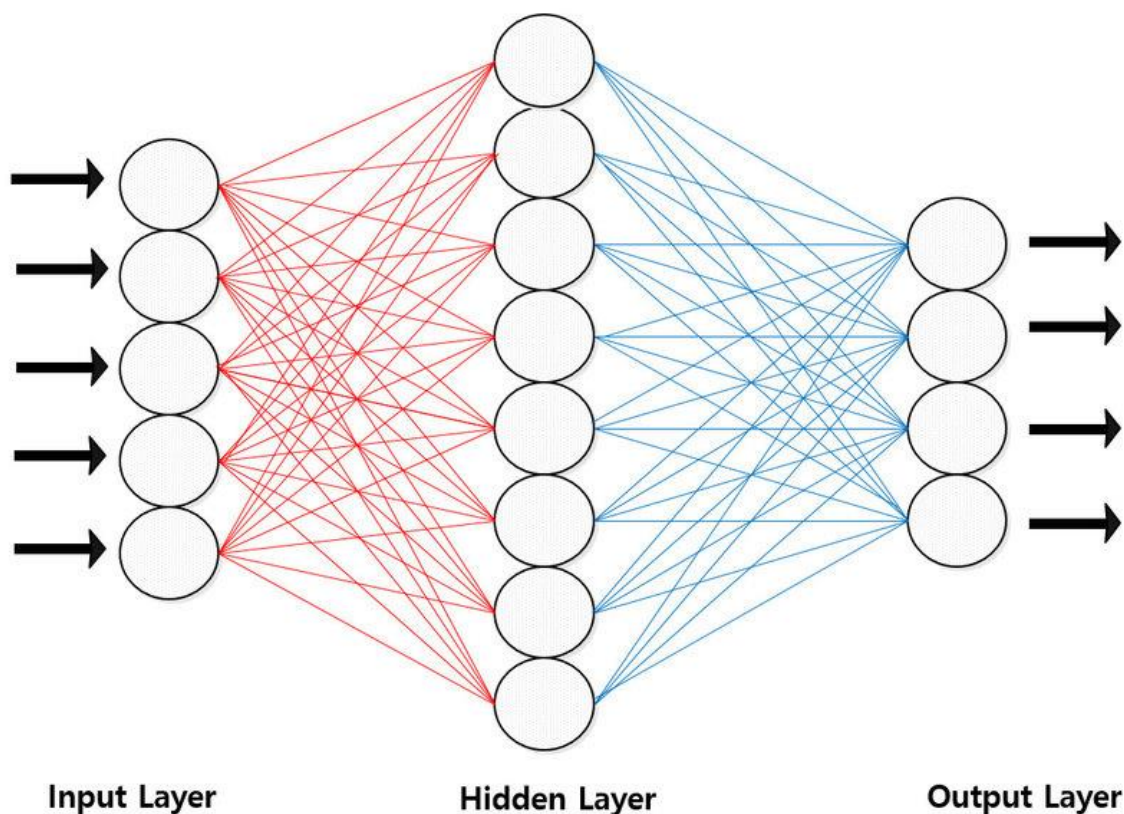


Рисунок 1.1 – Модель штучної нейронної мережі

Основна мета ШІ - автоматизація інтелектуальних завдань, традиційно виконуваних людьми. Однак важливо зазначити, що зі зростанням можливостей ШІ зростає й обговорення етичних і соціальних питань, пов'язаних із його використанням, включно з питаннями приватності, безпеки та впливу на ринок праці.

1.2 Принципи роботи штучного інтелекту

Принципи роботи штучного інтелекту (ШІ) базуються на спробах імітації людського інтелекту за допомогою комп'ютерних систем. Основні аспекти включають навчання, міркування, самовдосконалення, сприйняття та маніпуляцію з об'єктами. Основними ключовими елементами штучного інтелекту є:

1. Навчання: Основа ШІ полягає в здатності системи навчатися і вдосконалюватися з часом. Це може бути досягнуто за допомогою різних

методів машинного навчання, включно з наглядом навчанням (де системі надаються приклади з правильними відповідями), ненаглядом навчанням (де система вчиться ідентифікувати закономірності без вказівок) та підкріплювальним навчанням (де система вчиться на основі винагород за певні дії).

2.Міркування: ШІ використовує міркування для вирішення задач і прийняття рішень. Це може бути логічне міркування, де система використовує правила для виведення висновків, або евристичне міркування, де система робить припущення, що спираються на досвід.

3.Самовдосконалення: Системи ШІ можуть самоудосконалюватися через процеси, такі як глибоке навчання, де вони автоматично виявляють і виправляють помилки у своїх алгоритмах.

4.Сприйняття: ШІ може сприймати навколишнє середовище через датчики, камери, мікрофони тощо. Це дозволяє системам розпізнавати образи, звуки, текст та інші введені дані.

5.Маніпуляція з об'єктами: У певних випадках, особливо у робототехніці, ШІ включає здатність фізично взаємодіяти з об'єктами. Роботи можуть навчатися переміщати та маніпулювати об'єктами в реальному світі.

Технології, які лежать в основі ШІ, включають нейронні мережі, глибоке навчання, обробку природної мови та обчислювальну логіку. Розвиток і застосування цих технологій продовжують розвиватися, покращуючи можливості штучного інтелекту в різних областях.

1.3 Слабкий Штучний інтелект

Слабкий штучний інтелект (Weak AI), також відомий як "вузький ШІ", стосується систем, які призначені та здатні виконувати конкретні завдання без наявності свідомості, розуміння або самосвідомості. На відміну від сильного ШІ, який намагається відтворити загальний інтелект людини та здатний до самонавчання та розв'язання широкого спектру задач, слабкий ШІ спеціалізується на одній задачі або обмеженому наборі завдань.

Прикладами слабого ШІ є шахові програми, які можуть перемагати професіональних гравців, але не можуть виконувати нічого іншого, розпізнавання мови або зображень в смартфонах, рекомендаційні системи в електронній комерції та потокових сервісах. Ці системи ефективно вирішують задачі в своїй області, але не володіють розумінням за межами своєї конкретної задачі і не можуть адаптуватися або навчатися самостійно в інших контекстах.

Розвиток слабого ШІ має значний вплив на суспільство, автоматизуючи завдання, підвищуючи ефективність і створюючи нові можливості для бізнесу та науки. Водночас виклики, пов'язані з етикою, безпекою та впливом на ринок праці, потребують уважного розгляду.

1.4 Сильний штучний інтелект

Сильний штучний інтелект (Strong AI), також відомий як загальний штучний інтелект (AGI, Artificial General Intelligence), відсилає до гіпотетичної версії штучного інтелекту, яка може розуміти, мислити і вчитися так само ефективно, як і людський розум. Відмінність сильного ШІ від слабого ШІ (наприклад, спеціалізованих систем, таких як ігри в шахи або розпізнавання мови) полягає у його здатності до загального інтелектуального розуміння та вирішення проблем, не будучи обмеженим конкретним завданням або контекстом.

Теоретично, сильний ШІ міг би проводити трансфер знань, тобто застосовувати вивчене у одній області для розв'язання проблем у інших областях, як це роблять люди. Це включало б такі здібності, як сприйняття навколишнього середовища, планування, розуміння мови, власне мислення, творчість та емоційний інтелект.

На даний момент сильний ШІ залишається об'єктом теоретичних досліджень і наукової фантастики, оскільки існуючі технології штучного інтелекту здебільшого обмежені конкретними завданнями і не володіють

універсальністю людського розуму. Розробка сильного ШІ ставить перед науковцями численні технічні, етичні та філософські питання.

1.5 Різниця між слабким штучним інтелектом та сильним штучним інтелектом

Слабкий штучний інтелект (СШІ) та сильний штучний інтелект (СІІ) відрізняються за рівнем їх можливостей, обсягом завдань, які вони можуть виконувати, і ступенем наближення до людського інтелекту.

Слабкий ШІ :

- Слабкий ШІ розроблений для виконання конкретних завдань. Наприклад, система розпізнавання мови або рекомендаційна система.
- Він не має власної свідомості або розуміння.
- Слабкий ШІ взаємодіє зі світом за строго визначеними правилами та алгоритмами.
- Прикладами слабого ШІ є персональні асистенти типу Siri або Alexa, а також більшість існуючих систем ШІ, що використовуються в промисловості та бізнесі.

Сильний ШІ :

- Сильний ШІ, в ідеалі, має рівень інтелекту та когнітивних здібностей, що дорівнюють людським або перевершують їх. Він може вирішувати різноманітні завдання, адаптуватись до нових ситуацій і навчатися від досвіду.
- Теоретично сильний ШІ має свідомість, самосвідомість та емоції, хоча наразі це залишається в області наукової фантастики.
- Сильний ШІ здатен до загального інтелектуального мислення, що означає здатність розуміти, навчатися та застосовувати знання в широкому спектрі областей.
- Створення сильного ШІ представляє значні технічні та етичні виклики. Наразі більшість реалізованих систем ШІ належать до категорії слабого ШІ. Сильний ШІ залишається в основному теоретичною концепцією, до реалізації якої досі далеко.

1.6 Розробка штучного інтелекту

Розробка штучного інтелекту (ШІ) - це широке та багатогранне поле, яке охоплює багато різних підходів, технологій та застосувань. Це поле дослідження спрямоване на створення комп'ютерних систем, здатних виконувати завдання, які традиційно вимагають людського інтелекту, такі як розпізнавання мови, навчання, планування, та розуміння природної мови.

Основні компоненти в розробці ШІ включають:

1. Машинне навчання (ML): Це основа багатьох сучасних систем ШІ. Машинне навчання дозволяє комп'ютерам навчатися з даних, вдосконалюючи свої алгоритми з часом, без явного програмування.

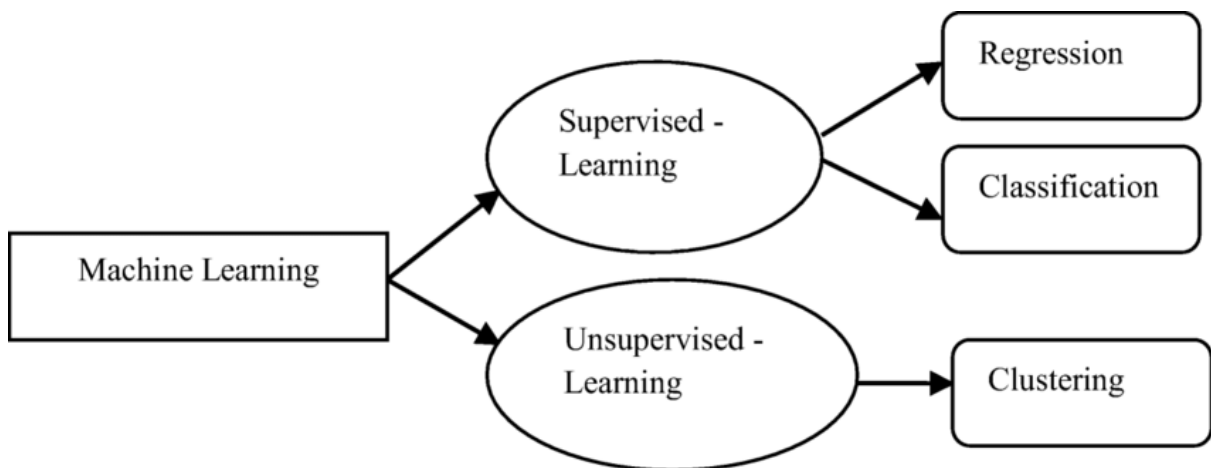


Рисунок 1.2 – Система машинного навчання

2. Глибоке навчання (Deep Learning): Підмножина машинного навчання, яка використовує нейронні мережі з багатьма шарами (так звані "глибокі" нейронні мережі) для аналізу різних видів даних. Глибоке навчання виявилось особливо ефективним у завданнях, пов'язаних з обробкою зображень та розпізнаванням мови.

3. Навчання з підкріпленням (Reinforcement Learning): Це область машинного навчання, де агенти вчаться приймати рішення, взаємодіючи з довкіллям. Цей метод інспірований процесом природного відбору та ідеями з теорії ігор.

4. Обробка природної мови (NLP): Це галузь, яка займається взаємодією між комп'ютерами та людською (природною) мовою. Вона включає в себе завдання, такі як переклад мови, розпізнавання мовлення та генерація тексту.

5. Комп'ютерний зір: Це галузь, яка дозволяє комп'ютерам "бачити" та розуміти зображення та відео. Вона використовує техніки машинного та глибокого навчання для інтерпретації візуальної інформації.

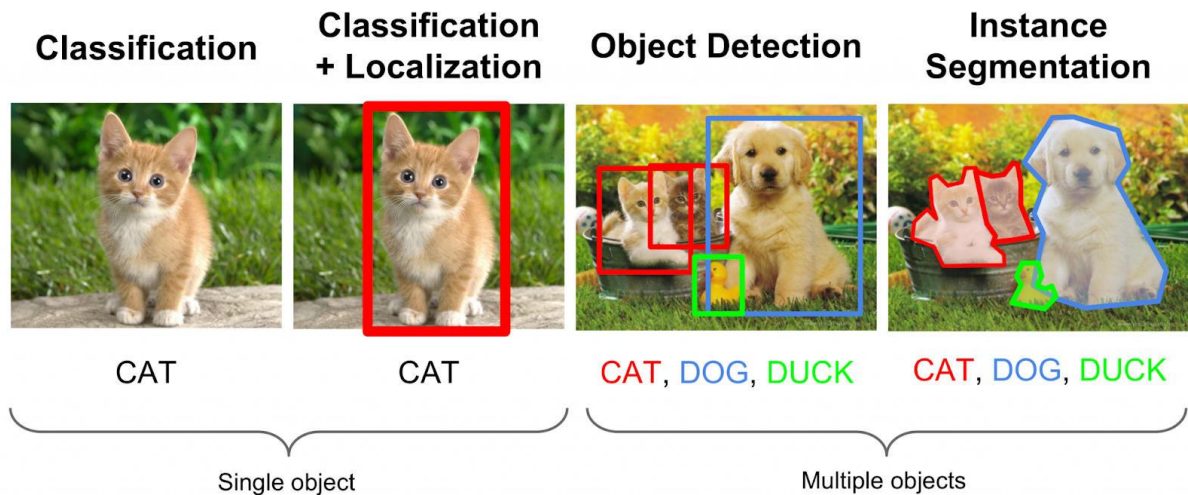


Рис 1.3 – На зображенні показана різниця між різними задачами в комп'ютерному зорі, як для поодиноких, так і для багатьох об'єктів.

Розробка ШІ також включає етичні та соціальні розгляди, особливо з огляду на потенційний вплив на робочі місця, конфіденційність, безпеку та інші аспекти суспільного життя. Розробники ШІ повинні враховувати ці фактори, проектуючи та впроваджуючи свої системи.

Висновок:

Цей розділ надає глибокий аналіз концепцій та різноманітності штучного інтелекту (ШІ), розглядаючи його від слабкого до сильного ШІ. Виділено основні властивості та поділ на слабкий та сильний ШІ, що дозволяє зрозуміти різницю в потенціалі та обмеженнях цих двох форм. Основна увага зосереджена на тому, як штучний інтелект може виконувати завдання, традиційно зарезервовані для людського розуму, такі як вирішення проблем,

навчання, сприйняття, і міркування, а також на способах його реалізації через машинне навчання, глибоке навчання, обробку природної мови та інші технології.

Важливим є визнання, що, незважаючи на значний прогрес в галузі ШІ, сильний ШІ залишається в межах теоретичних досліджень та має численні технічні, етичні, і соціальні виклики, що мають бути адресовані. Розвиток та впровадження ШІ в суспільство вимагає обачливого підходу з урахуванням можливих наслідків для приватності, безпеки, і ринку праці, а також етичних дилем, які виникають з його застосуванням.

Закінчуючи, розділ акцентує на величезному потенціалі ШІ для інновацій у багатьох секторах, але також підкреслює необхідність глибокого розуміння його обмежень та комплексного підходу до вирішення пов'язаних з ним викликів.

РОЗДІЛ 2

СТРУКТУРА CHATGPT(OPEN AI)

2.1 Моделі Open AI

Open AI зробили відкриття коли показали свою текстову модель штучного інтелекту(LLM) GPT3.5, це був нонсенс у світі штучного інтелекту. Але Open Ai не зупинились на цьому і показали інші свої моделі, які не тільки текстові.

Ось деякі з найбільш відомих моделей OpenAI:

1.GPT (Generative Pre-trained Transformer): Це серія моделей, які спроможні генерувати текст, що схожий на людський. Від GPT-1 до останньої версії, GPT-4, кожна наступна модель стає все більш потужною та здатною до розуміння складних контекстів і вироблення тексту відповідно до них.



Вы

Напиши код для swift із виведенням у консоль привіт світ



ChatGPT

Ось простий приклад коду на Swift, який виводить фразу "Привіт світ" у консоль:

```
swift Copy code

import Foundation

print("Привіт світ")
```

Цей код можна виконати у будь-якому середовищі Swift, наприклад, у Xcode або онлайн компіляторі Swift. Команда `print` використовується для виведення тексту в стандартний вихід, що зазвичай є консоллю.

Рис 2.1 - Робота з моделлю штучного інтелекту GPT-4

2.DALL·E: Ця модель використовується для генерації зображень на основі текстових описів. DALL·E здатна створювати детальні та виразні зображення з дуже специфічних та абстрактних описів.

WT Вы

Згенеруй рудого котика, що стоїть на столі, серед темного лісу, широкоекранне співвідношення сторін, антиутопія

DALL·E



Here are the images of a ginger cat standing on a table in a dark, dystopian forest. You can view and download them directly from above.

Рис 2.2 – Робота з моделлю штучного інтелекту DALL·E

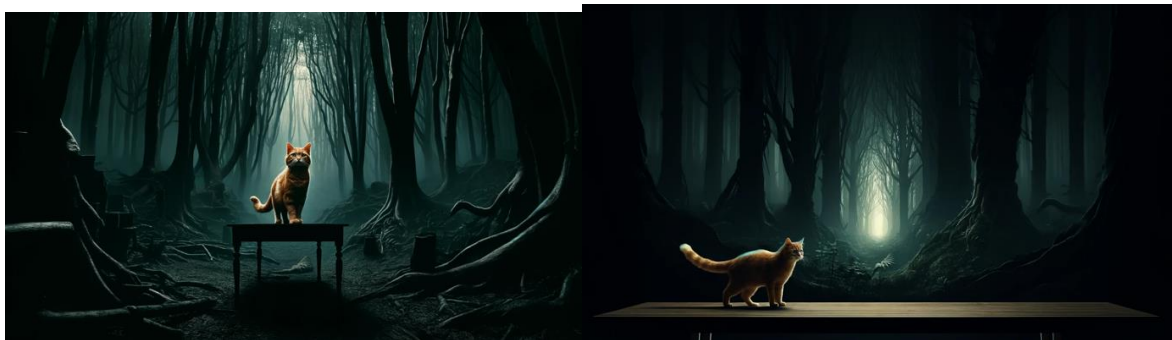


Рис 2.3 – Приклад роботи моделі штучного інтелекту DALL·E

3.CLIP (Contrastive Language–Image Pre-training): CLIP призначена для розуміння та інтерпретації зображень в контексті натуральної мови. Модель використовується для визначення відповідності між зображеннями та текстовими описами, що дозволяє їй виконувати завдання, пов'язані з класифікацією зображень та візуальним пошуком.

Approach

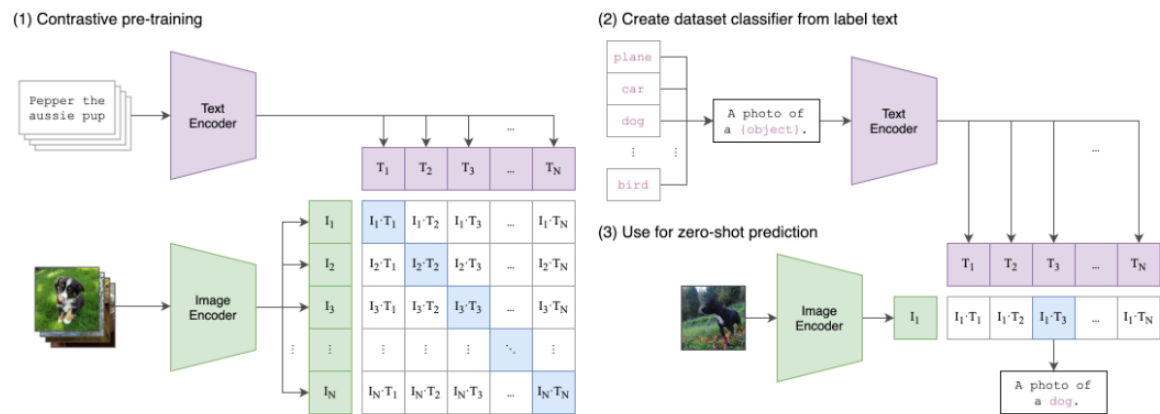


Рис 2.4 – Блок-схема роботи CLIP

4.Codex: Це система AI, оптимізована для розуміння та генерації коду. Codex може допомагати в розробці програмного забезпечення, автоматично генеруючи код на основі природної мови.

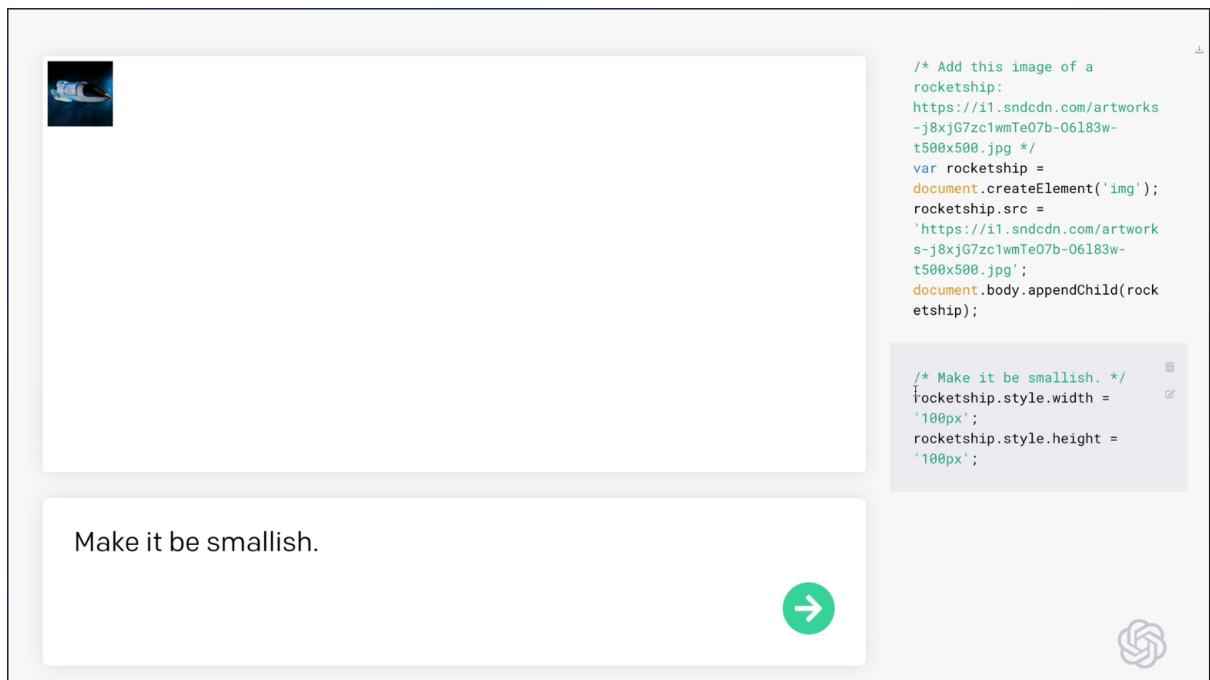


Рис 2.5 – Робота з моделлю штучного інтелекту Codex

5.Whisper: Це система розпізнавання мовлення, розроблена для перетворення аудіо мовлення на текст. Whisper здатна розпізнавати мовлення з різних джерел і в різних умовах, що робить її ефективною для різноманітних застосувань.

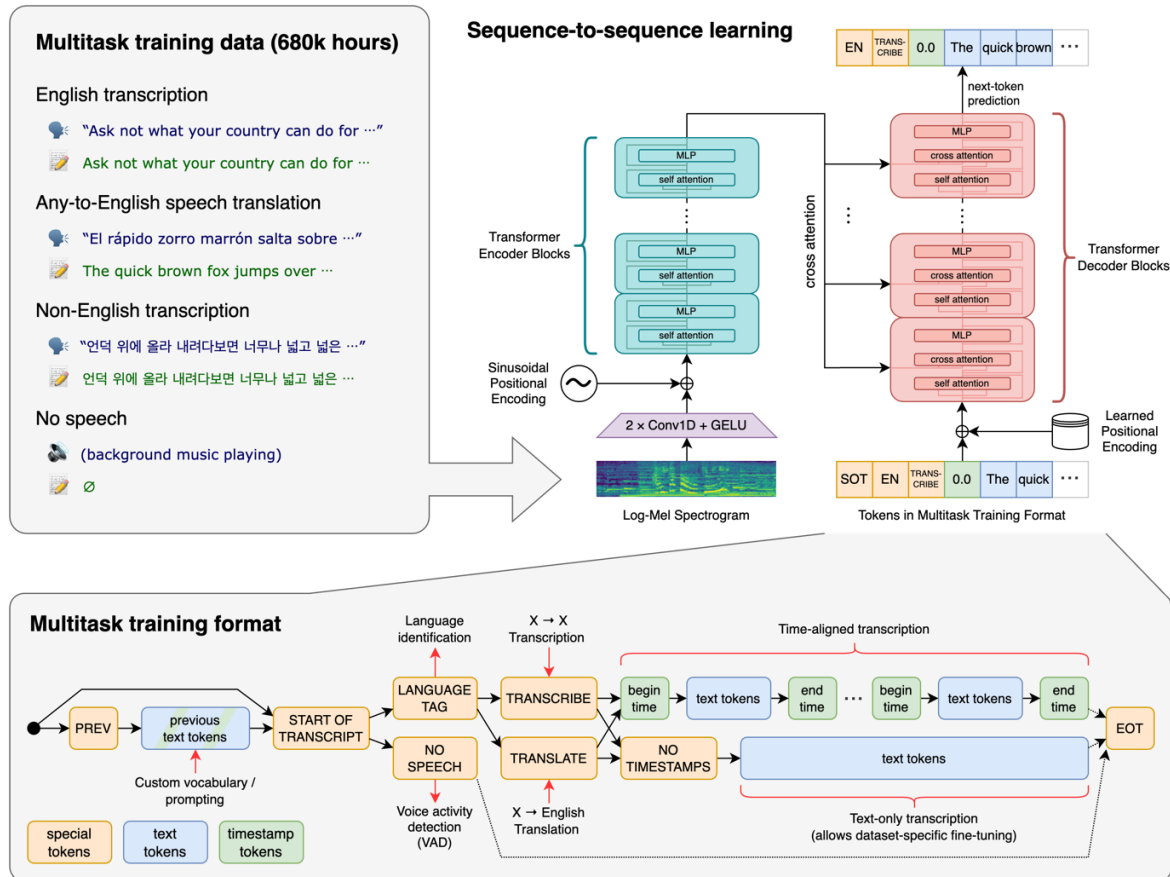


Рис 2.6 – Принцип роботи Whisper

6.Sora: Це вдосконалена модель перетворення тексту у відео, розроблена OpenAI, яка використовує найсучасніші моделі дифузії (SOTA) для перетворення текстових описів у відеокліпи високої чіткості. Її архітектура поєднує дифузійну модель з трансформаторною технологією, що дозволяє ефективно навчати та генерувати відео. Методологія Sora включає уніфіковане представлення, на основі патчів, мережі стиснення відео та дифузійні трансформатори, пропонуючи такі можливості, як підказки до зображень і відео, анімація зображень DALL-E, розширення відео, редагування та генерація зображень. Однак вона також має обмеження, такі як труднощі в моделюванні складних просторів і розумінні причинно-

наслідкових зв'язків, і робить сильний акцент на міркуваннях безпеки, таких як тестування "червоною командою" і виявлення контенту.

7.OpenAI Gym: Це інструментарій для розробки та порівняння алгоритмів навчання з підкріпленням. Він забезпечує широкий спектр середовищ, від простих завдань до складних завдань, де AI моделі можуть вчитися та вдосконалюватися.

8.Safety Gym: Це набір інструментів, розроблений для випробування та покращення безпеки алгоритмів навчання з підкріпленням, зосереджуючись на тому, щоб моделі не тільки досягали своїх цілей, але й робили це безпечно.

9.Baseline: OpenAI Baseline: Це набір високоякісних імплементацій алгоритмів навчання з підкріпленням. Вони забезпечують стандартний відлік для наукових досліджень, дозволяючи вченим легко порівнювати нові алгоритми з наявними.

10.OpenAI Microscope: Це колекцією візуалізацій для широкого спектру моделей машинного навчання, розроблених OpenAI. Ця платформа дозволяє детально вивчати, як працюють ці моделі на нейронному рівні.

11.RoboSumo: Це середовище для симуляції та тренування роботизованих агентів, щоб змагатися у сумо. Це дозволяє досліджувати складні динамічні взаємодії між агентами, що конкурують у фізичному просторі.

Кожна з цих моделей має свої унікальні характеристики та області застосування, і вони продовжують розвиватися з часом, надаючи все більше можливостей для інновацій у галузі штучного інтелекту.

2.2 GPT3.5 та GPT4

Як ми вже знаємо що GPT3.5 та GPT4 це мовні моделі. В них є свої архітектури на яких вони базуються.

Архітектура Transformer

Механізм Уваги (Attention Mechanism): Ядро архітектури Transformer. Дозволяє моделі зосереджуватись на різних частинах вхідних даних при

обробці кожного слова чи токена, що покращує здатність моделі вловлювати контекст та зв'язки в тексті.

Блоки Self-Attention: У кожному шарі Transformer блоки self-attention оцінюють, наскільки важливими є всі інші токени в тексті для кожного токена, що допомагає моделі краще зрозуміти залежності та структуру мови.

Позиційне Кодування (Positional Encoding): Оскільки моделі Transformer не мають внутрішньої розуміння порядку слів, позиційне кодування додається до вхідних даних для забезпечення інформації про позицію токена в послідовності.

Масштабування та Навчання

Кількість Шарів та Параметрів: Більші моделі, як GPT-4, містять значно більшу кількість шарів Transformer та параметрів, ніж їхні попередники, що дозволяє їм зберігати більше інформації та виконувати більш складні мовні завдання.

Техніки Оптимізації: Використовуються різні техніки оптимізації, такі як регуляризація, дропаут та розширення даних, для покращення здатності моделі до узагальнення та зменшення ризику перенавчання.

Процедура Навчання: Моделі навчаються на величезних корпусах тексту, використовуючи метод навчання з вчителем, де модель намагається передбачити наступний токен на основі попередніх.

Вдосконалення та Нововведення

Адаптивність до Контексту: Завдяки великим об'ємам даних та масштабованій архітектурі, GPT-3.5 та GPT-4 мають здатність адаптуватися до широкого спектру контекстів та стилів тексту, роблячи їх здатними до вирішення багатьох різних завдань.

Вдосконалені Механізми Уваги: У GPT-4 були внесені покращення в механізми уваги, що дозволяє моделі ефективніше обробляти довші послідовності та краще розуміти складні зв'язки в тексті.

Багатоцільність: Моделі спроектовані таким чином, щоб виконувати багато різних завдань без потреби у специфічних для задачі налаштуваннях

або тренуванні, що робить їх надзвичайно гнучкими інструментами для обробки мови.

Висновок:

У другому розділі описується вражаючі досягнення OpenAI в розвитку моделей штучного інтелекту, демонструючи їх широкий спектр застосувань від генерації тексту до візуалізації зображень і розуміння природної мови. GPT-3.5 та GPT-4, з їх передовою архітектурою трансформерів і механізмом уваги, стали символом значного прогресу в технології обробки мови, забезпечуючи глибше розуміння контексту та зв'язків у тексті. Моделі як DALL·E, CLIP, і Codex відкривають нові можливості для креативності та автоматизації, а Whisper і Sora підсилюють взаємодію між різними типами медіа. Цей розділ підкреслює не лише технічні досягнення OpenAI, але й їхнє зосередження на безпеці, адаптивності і мультифункціональності, що робить їх інструментами нового покоління для інновацій у галузі штучного інтелекту. Це не просто технологічний прогрес, це відкриття нових можливостей для дослідження і розвитку в сфері AI.

РОЗДІЛ 3

РОЗРОБКА ЗАСТОСУНКУ iOS

3.1 Інструменти та паттерни розробки

При розробці голосового помічника для платформи iOS, вибір правильних інструментів та паттернів проектування є ключовим для створення надійного та ефективного додатку.

Інструменти розробки

1. Swift - це сучасна, потужна і зручна мова програмування, розроблена Apple у 2014 році. Вона призначена для розробки додатків на iOS, macOS, watchOS, iPadOS, visionOS і tvOS. Swift поєднує в собі простоту і безпеку, пропонуючи функції, як-от керування пам'яттю без збирача сміття, підтримку об'єктно-орієнтованого і функціонального програмування, а також сувору типізацію. Мова вирізняється високою продуктивністю та інтерактивним середовищем, що робить її популярним вибором серед розробників.

2. Xcode - це інтегроване середовище розробки (IDE) від Apple, призначене для розробки програмного забезпечення для macOS, iOS, watchOS і tvOS. Воно містить у собі редактор коду, компілятор і різні інструменти для проектування інтерфейсів, а також підтримує мови програмування Swift і Objective-C. Xcode також надає інструменти для аналізу продуктивності та налагодження програм, що робить його основним інструментом для розробників додатків для продукції Apple.

3. UIKit - це фреймворк для розроблення графічного інтерфейсу, що надається Apple в рамках iOS. Він містить набір інструментів для управління елементами користувацького інтерфейсу, подіями введення та управлінням життєвим циклом додатків. Розробники використовують UIKit для створення та управління ієрархіями візуальних елементів, таких як кнопки, текстові поля, слайдери та інші керуючі елементи. UIKit тісно інтегрований з мовою

програмування Swift, надаючи великі можливості для розробки інтуїтивно зрозумілих та інтерактивних додатків.

4. `Speech` - це потужний інструмент для розпізнавання та обробки мови, розроблений Apple. Вона дає змогу додаткам використовувати розпізнавання мови для перетворення аудіо на текст і підтримує безліч мов і діалектів. Бібліотека `Speech` інтегрується з голосовими командами Siri і може працювати як у режимі онлайн, так і офлайн.

5. `AVFoundation` - це потужний фреймворк в iOS і macOS для роботи з аудіо та відео. Він дає змогу користувачам записувати, редагувати та відтворювати аудіо та відео файли. Фреймворк надає широкий спектр можливостей, включно з керування потоками медіа-даних, зміною швидкості відтворення, обробкою та аналізом аудіо- та відеоконтенту.

6. `SnapKit` - це DSL (domain-specific language, предметно-орієнтована мова) для авторозподілу в Swift, що спрощує додавання обмежень в iOS і macOS. Він надає чистіший і зрозуміліший синтаксис порівняно з нативними `NSLayoutConstraint`, що робить процес розроблення інтерфейсів більш інтуїтивним. Бібліотека підтримує як код, написаний на Swift, так і Objective-C, і використовується для динамічного визначення розмірів і позиціонування елементів призначеного для користувача інтерфейсу залежно від зміни умов навколишнього середовища, наприклад, зміни орієнтації пристрою.

Паттерни розробки

1. `MVVM` (Model-View-ViewModel) - це шаблон проектування, який використовується для побудови застосунків на Swift, особливо популярний у розробці iOS. Цей підхід допомагає відокремити логіку представлення (UI) від бізнес-логіки та даних застосунку, що спрощує тестування та підтримку коду.

-`Model` (Модель): Містить основні дані та бізнес-логіку програми. Моделі не залежать від компонентів користувацького інтерфейсу.

-`View` (Подання): Відображає візуальні елементи та інтерфейс, через який користувач взаємодіє з додатком. У контексті iOS, це зазвичай усе, що пов'язано з `UIKit` або `SwiftUI`.

-ViewModel (В'ю-модель): Шар, який містить логіку управління поданнями, що не включає безпосередньо бізнес-логіку даних. ViewModel перетворює моделі на значення, які можуть бути легко відображені в Поданнях.

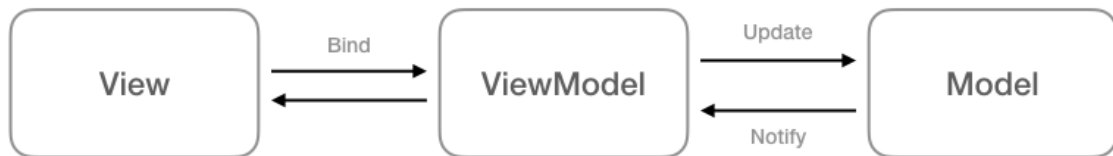


Рис. 3.1 - принцип роботи архітектури MVVM

2.Delegate Pattern - використовується для визначення комунікації між об'єктами в iOS. Часто застосовується для реагування на події інтерфейсу, такі як кліки по кнопках або зміни у текстових полях.

3.Observer Pattern - дозволяє одному об'єкту спостерігати і реагувати на події, що відбуваються в іншому об'єкті. Використовується для ефективного сповіщення компонентів додатка про зміни даних.

4.Singleton Pattern - паттерн, який обмежує ініціалізацію класу одним об'єктом. Це корисно для компонентів, які повинні мати лише один екземпляр у додатку, наприклад, менеджери налаштувань або з'єднання з мережею.

3.2 Розробка інтерфейсу на UIKit та SnapKit

Розробка інтерфейсу користувача (UI) є одним з найважливіших аспектів створення додатків, особливо коли мова йде про голосові помічники, де інтуїтивно зрозумілий та естетично приємний інтерфейс може значно покращити взаємодію з користувачем. UIKit та SnapKit є двома основними інструментами, які використовуються для розробки UI в додатках iOS.

UIKit: Основи та Компоненти

UIKit — це фреймворк розробки, який надає розширені інтерфейсні елементи та класи для управління та відображення користувацького

інтерфейсу на iOS. Він включає широкий спектр елементів управління, таких як кнопки, слайдери, перемикачі, лейбли, і віджети для складніших завдань, як-от навігаційні контролери, вікна вибору дати та інші. UIKit також забезпечує управління жестами, анімацією, рисуванням графіки та підтримку багатозадачності.

SnapKit: Фасилітація Auto Layout

SnapKit є DSL (domain-specific language) для Auto Layout, який значно спрощує задачу створення констрейнтів для інтерфейсів в коді. Цей інструмент допомагає розробникам швидше та легше створювати адаптивний дизайн, який працює на всіх пристроях Apple. SnapKit зменшує кількість коду, необхідного для встановлення констрейнтів, та робить код чистішим і легшим для розуміння.

Процес Розробки:

1.Планування інтерфейсу: Перш ніж перейти до кодування, необхідно зробити мокапи інтерфейсу, визначити основні компоненти UI та їх взаємодії.

2.Реалізація з UIKit: Створення базових елементів інтерфейсу відбувається з використанням компонентів UIKit. Важливо забезпечити, що елементи інтерфейсу добре адаптуються до різних розмірів екранів та орієнтацій.

Використання SnapKit для Layout: SnapKit використовується для додавання констрейнтів до елементів UI, що забезпечує їх коректне розташування на екрані без зайвих зусиль. Наприклад, якщо потрібно, щоб кнопка була в центрі екрану і мала певні відступи від країв, SnapKit дозволяє реалізувати це в кілька рядків коду.

3.Тестування інтерфейсу: Після реалізації інтерфейсу слід провести тестування на різних пристроях та в різних умовах (наприклад, при різному освітленні та за різних налаштувань доступності), щоб забезпечити його коректну роботу та зручність для кінцевих користувачів.

Використання UIKit разом із SnapKit надає потужні можливості для створення динамічних, адаптивних і доступних інтерфейсів, що є невід'ємною частиною розробки сучасних мобільних додатків.

3.3 Інтеграція з базою даних

Інтеграція голосового помічника з базою даних є фундаментальним елементом для забезпечення його функціональності, оскільки дозволяє зберігати та використовувати користувацькі дані, історію запитів, відповіді та іншу інформацію, що збільшує корисність додатку. Нижче наведено ключові кроки та рекомендації для ефективної інтеграції з базою даних.

Вибір Системи Управління Базами Даних (СУБД)

Локальні СУБД: Для мобільних додатків часто вибирають локальні бази даних як SQLite або Realm через їх легкість та ефективність на мобільних пристроях. Вони дозволяють зберігати дані безпосередньо на пристрої користувача.

Інтеграція СУБД

1. Модель даних: Визначте схему бази даних, яка включатиме всі необхідні таблиці та поля для збереження даних, наприклад, користувацькі налаштування, історію запитів та відповіді. Модель повинна бути гнучкою, щоб підтримувати майбутнє розширення додатку.

2. Робота з ORM: Використання ORM-бібліотек (наприклад, Core Data в iOS), що дозволяє взаємодіяти з базою даних на високому рівні абстракції, спрощує роботу з даними як з об'єктами, мінімізуючи кількість SQL-коду.

3. Безпека: Забезпечення безпеки даних має вирішальне значення. Використовуйте шифрування бази даних та безпечні методи автентифікації та авторизації для захисту інформації користувачів.

Тестування та оптимізація

1. Тестування продуктивності: Важливо тестувати базу даних на продуктивність, особливо при великих об'ємах даних та запитів, щоб забезпечити, що додаток залишається швидким та ефективним.

2.Оптимізація запитів: Оптимізування SQL-запити для підвищення ефективності, особливо ті, що виконуються найчастіше. Використання індексів, правильне структурування запитів та уникнення непотрібних даних у відповідях допоможе знизити навантаження на базу даних.

3.Масштабування: Масштабування бази даних з урахуванням зростаючої кількості користувачів та даних. Використання обlačних рішень, які легко масштабуються і забезпечують високу доступність та надійність. Інтеграція з базою даних є важливою складовою розробки голосового помічника, яка вимагає уваги до деталей, глибокого розуміння технологій та врахування потреб кінцевих користувачів. Забезпечення правильної реалізації інтеграції підвищить ефективність додатку.

3.4 Інтеграція openAI API в додаток

Інтеграція API OpenAI у розробку голосового помічника для iOS надає значні переваги для покращення його функціоналу, зокрема у сфері аналізу та генерування мови. Завдяки можливостям GPT (Generative Pre-trained Transformer) від OpenAI, можна створити взаємодію з користувачами, що враховує контекст та глибше розуміє запити, забезпечуючи відповідність та релевантність відповідей.

Інтеграція API OpenAI

1.Реєстрація та доступ: Перший етап передбачає реєстрацію на платформі OpenAI та отримання ключів API, що дозволяє додатку взаємодіяти з сервісом. Цей процес включає налаштування заходів безпеки та збереження ключів у захищеному середовищі.

Використання API у додатку: З отриманими ключами, необхідно вбудувати HTTP-клієнт у iOS додаток для виконання запитів до API. Використовувати такі інструменти, як URLSession або Alamofire для налагодження запитів, обробки відповідей і управління помилками.

2.Обробка отриманих відповідей: Відповіді зазвичай надходять у форматі JSON. Модуль для аналізу цих даних і перетворення їх на формати,

які бенефіціють ваш додаток, а також обробка помилок або неочікуваних відповідей сервера.

3.Інтеграція з користувальницьким інтерфейсом: Використовуйте відповіді від API для демонстрації тексту, який генерує помічник, в UI компонентах додатку, звертаючи увагу на зменшення затримок між запитами та відповідями для забезпечення плавності користувальницького досвіду.

Безпека та приватність

1.Конфіденційність даних: Забезпечте анонімізацію і відокремлення всіх даних, що передаються до OpenAI, від особисто ідентифікованої інформації.

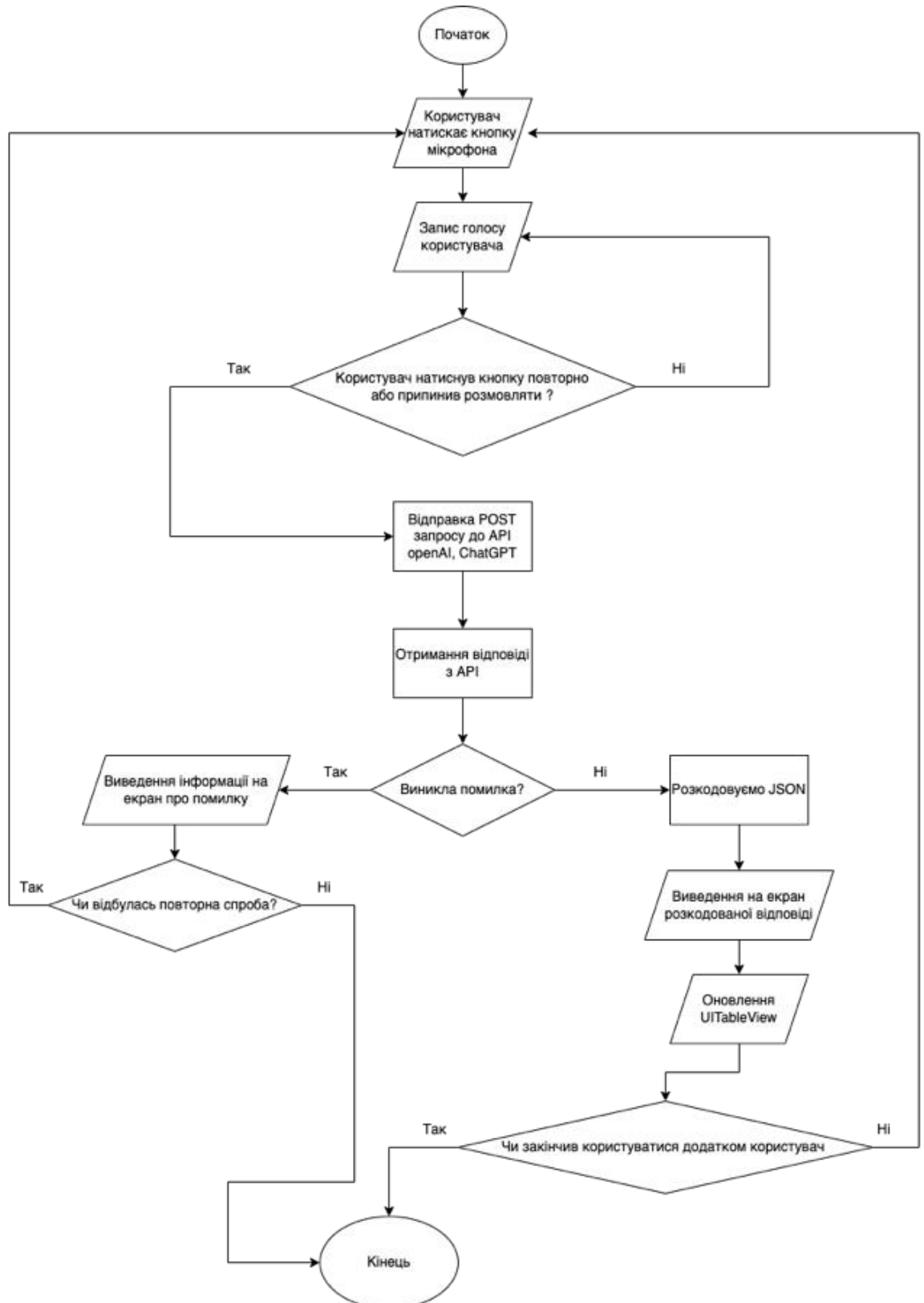
2.Шифрування: Забезпечте захист даних при їх передачі за допомогою шифрування, щоб уникнути перехоплення.

3.Моніторинг використання: Стежте за частотою і характером використання API, адже це може впливати на витрати та обмеження встановлені OpenAI.

4.Оптимізація запитів: Оптимізуйте вміст запитів, щоб зменшити витрати і збільшити ефективність отриманих відповідей.

Ці кроки з інтеграції API OpenAI дозволять розширити можливості голосового помічника, зробивши його більш респонсивним і здатним до адекватної реакції на складні запити користувачів.

3.5. Принцип роботи додатку



Висновок

Розробка голосового помічника на базі ChatGPT для платформи iOS ставить перед розробниками велику кількість технічних завдань, що включають інтеграцію з передовими технологіями обробки мови, забезпечення безпеки та зручності користувачів. Завдяки продуманому підходу до архітектури, вибору інструментів і методів тестування, можливо створити ефективний та надійний продукт.

ВИСНОВОК

У ході виконання кваліфікаційної роботи на тему «iOS додаток «голосовий помічник» на базі ChatGPT» було досягнуто кілька важливих результатів, які свідчать про актуальність та перспективність проєкту.

1. Аналіз та дослідження:

- Проведено глибокий аналіз поняття штучного інтелекту (ШІ), його видів, архітектур та механізмів роботи.
- Визначено основні функції та можливості штучних інтелектів, а також їх типи за різними класифікаційними ознаками.
- Досліджено застосування штучного інтелекту в різних галузях, таких як охорона здоров'я, фінанси, автомобільна промисловість та розваги, що дозволило окреслити потенційні сфери використання голосового помічника.

2. Вивчення ринку:

- Проаналізовано існуючі продукти на ринку штучних інтелектів, представлені компанією OpenAI.
- Розглянуто структуру та можливості моделей GPT-3.5 та GPT-4, що є передовими розробками в області обробки природної мови.
- Оцінено потенціал та обмеження даних технологій, що дозволило зрозуміти їхні сильні та слабкі сторони у контексті розробки голосових помічників.

3. Проєктування додатку:

- Розроблено детальну структуру додатку, що включає всі необхідні етапи розробки: від планування та проєктування до реалізації та тестування.

- Вибрано мову програмування Swift та використано фреймворки UIKit і SnapKit для створення користувацького інтерфейсу, що дозволяє забезпечити зручність та інтуїтивність взаємодії користувача з додатком.
- Визначено основні паттерни розробки, такі як MVVM (Model-View-ViewModel), що допомагають у підтримці чистого та зрозумілого коду.

4. Інтеграція технологій:

- Визначено можливості інтеграції технологій обробки природної мови від OpenAI, зокрема моделей GPT-3.5 та GPT-4.
- Розглянуто питання реалізації розпізнавання та генерації мовлення, що забезпечує високу якість відповіді на запити користувачів.
- Описано процес інтеграції API OpenAI, включаючи отримання ключів доступу, налаштування HTTP-запитів та обробку відповідей.

5. Безпека та конфіденційність:

- Обговорено аспекти забезпечення безпеки та захисту персональних даних користувачів, включаючи шифрування даних та захист від несанкціонованого доступу.
- Визначено важливість забезпечення конфіденційності даних, що передаються до OpenAI, та вжиття заходів для анонімізації інформації користувачів.

6. Тестування та оцінка:

- Розроблено план тестування додатку, що включає перевірку функціональності, продуктивності та зручності користування.

- Проведено тестування прототипу додатку, яке показало його відповідність запланованим функціональним можливостям.
- Отримано відгуки від тестових користувачів, що дозволило зробити висновки про ефективність розробленого продукту та виявити можливі напрямки для покращення.

Таким чином, виконана кваліфікаційна робота підтвердила доцільність використання технології ChatGPT для розробки голосових помічників та продемонструвала потенціал створення інноваційного продукту, який відповідає сучасним вимогам ринку мобільних додатків. Подальші дослідження та розробка можуть бути спрямовані на вдосконалення функціоналу, оптимізацію роботи додатку та розширення його можливостей, що забезпечить ще більш високий рівень взаємодії з користувачами та підвищить його конкурентоспроможність.