

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

**РОЗРОБКА ГОЛОСОВОГО ПОМІЧНИКА ДЛЯ DYNAMICS
365 BUSINESS CENTRAL**

**Текстова частина до курсової роботи
за спеціальністю 121 «Інженерія програмного забезпечення»**

Керівник курсової роботи

Корнійчук М.А.

(підпис)

“ ____ ” _____ 2022 р.

Виконала студентка

Кузів В.Д.

“ ____ ” _____ 2022 р.

Міністерство освіти і науки України
 НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
 Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
 Зав.кафедри інформатики,
 доцент, кандидат фіз.-мат. наук
 _____ С. С. Гороховський
 (підпис)
 „_____” _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
 на курсову роботу

студентці Кузів Васи́ліні

4 курсу
 факультету інформатики

ТЕМА: «Розробка голосового помічника для Dynamics 365 Business Central»

Вихідні дані:

Зміст ТЧ до курсової роботи:

Вступ

Розділ 1. Аналіз предметної області

1.1 Огляд ERP-системи Business Central від Microsoft

1.2 Розгортання системи

Розділ 2. Розробка в середовищі BC

2.1 Розробка розширень на основі Microsoft Dynamics 365 Business Central

2.2 Об'єкти в Business Central

2.3 Розробка розширення. Розробка користувацького функціоналу

2.4 Business Central API-інтеграція зі сторонніми продуктами

Розділ 3. Розробка голосового інтерфейсу

3.1. Аналіз технологій та ресурсів

3.2. Проектування архітектури та розробка додатку

3.3. Подальша розробка та покращення

Висновки

Список джерел

Додатки

Дата видачі „_____” _____ 202_ р. Керівник _____
 (підпис)

Завдання отримано _____
 (підпис)

Календарний план виконання роботи

Тема: «Розробка голосового помічника для Dynamics 365 Business Central»

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	Листопад 2021р.	
2.	Вивчення предметної області.	Листопад 2021р.	
3.	Огляд літератури за темою роботи.	Грудень 2021р.	
3.	Розгортання ВС	Січень 2022р.	
4.	Проектування системи	Січень 2022р.	
5.	Програмування розробленої системи	Лютий – березень 2022р.	
6.	Написання пояснювальної роботи.	Березень 2022р.	
7.	Створення слайдів для доповіді та написання доповіді.	Березень 2022р.	
8.	Аналіз отриманих результатів з керівником, написання доповіді.	Квітень 2022р.	
9.	Остаточне оформлення пояснювальної роботи та слайдів.	Квітень 2022р.	
10.	Захист курсової роботи (проекту)	???.	

Студент Кузів В.Д.

Керівник Корнійчук М.А.

“ _____ ” _____

Зміст

Анотація	5
Вступ.....	6
Розділ 1. Аналіз предметної області.....	9
1.1. Огляд ERP-системи Business Central від Microsoft	9
1.2. Розгортання системи	10
Розділ 2. Розробка в середовищі BC	14
2.1. Розробка розширень на основі Microsoft Dynamics 365 Business Central 14	
2.2. Об'єкти в Business Central	14
2.3. Розробка розширення. Розробка користувацького функціоналу	16
2.4. Business Central API-інтеграція зі сторонніми продуктами	20
Розділ 3. Розробка голосового інтерфейсу	24
3.1. Аналіз технологій та ресурсів	24
3.2. Проектування архітектури та розробка додатку	27
3.3. NPL-імплементация та створення моделі LUIS	30
Висновки	35
Список використаних джерел	37
Додатки.....	40

Анотація

У цій роботі виконується розробка розширення для менеджменту(реєстрації, купівлі\продажу, оренди) транспортних засобів на базі Microsoft Dynamics 365 Business Central і проектування та розробка голосового користувацького інтерфейсу з набором команд для керування цим розширенням. Спочатку виконується ґрунтовний аналіз предметної області: дослідження ринку, збір фактів, огляд ERP-систем та голосових асистентів. Після цього здійснюється розгортання системи Business Central локально та підготовка середовища розробки.

Для подальшої розробки вивчено основні аспекти мови програмування AL та структуру об'єктів у ВС. Наступним етапом є розробка розширення, програмування усіх компонентів та підготовка для інтеграції з зовнішніми додатками за допомогою веб-сервісів.

Останній розділ містить інформацію про створення голосових інтерфейсів, опис необхідних технологій та реалізацію на мові Python.

Вступ

Актуальність, наукове та практичне значення обраної теми

Голосові інтерфейси, які дуже швидко розвиваються протягом п'яти останніх років, змінюють наше уявлення про використання різноманітних технологій. Згідно з дослідженнями компанії Juniper Networks, станом на 2020 рік використовувалося близько 3,25 мільярдів голосових помічників; вважають, що у 2024 цей показник сягатиме 8 мільярдів. Крім того, за даними статистичного порталу Statista, більше 31% користувачів смартфонів використовують голосових пошук хоча б один раз в тиждень. Для деяких людей використання голосового помічника стало невід'ємною частиною повсякденного життя[2].

VUI(Voice User Interface) є надзвичайно корисним для Microsoft Dynamics 365 Business Central. Business Central – система для планування ресурсів підприємств; продукт призначений для керування фінансами, виробництвом, управлінням відносинами з клієнтами, ланцюгами поставок, аналітикою та електронною комерцією бізнесу. Станом на 2021 рік більше 15000 компаній по всьому світу використовує цю ERP-систему, не рахуючи близько 175 тисяч NAV клієнтів, що є попередньою версією BC. Business Central має величезний перелік доступного функціоналу, з яким інколи досить складно розібратися. Голосовий асистент може спростити виконання багатьох функцій.

Основною перевагою VUI для BC є швидкість. Користувачі системи виконують тисячі операцій за день. Деякі процеси, наприклад, створення нового товару або замовлення на покупку\продаж, вимагають багатьох дій користувачів: відкриття різних сторінок, заповнення багатьох полів, запуск процесів. З голосовим асистентом ці процеси відбуватимуться в рази швидше.

Також це дозволяє зберігати час, оптимізуючи повторюванні завдання. У бізнесі час – це гроші.

У цій курсовій роботі буде розроблено розширення для ВС для менеджменту(купівля\продаж, оренда) транспортних засобів, як приклад для імплементації технології голосового інтерфейсу для комерційного продукту.

Мета та завдання курсової роботи

Мета: Розробити розширення на основі ВС для менеджменту(купівлі\продажу, оренди) транспортних засобів та створити голосовий користувацький інтерфейс для цього розширення.

Завдання: Спроекувати дизайн та архітектуру системи менеджменту транспортних засобів та реалізувати голосовий користувацький інтерфейс з набором основних функціональних можливостей.

Об'єкт дослідження

Проектування та розробка розширення на основі Microsoft Dynamics 365 Business Central. Розробка голосового користувацького інтерфейсу на мові програмування Python з доступом до Business Central.

Практичне значення одержаних результатів

Розроблена система є повноцінним комерційним продуктом для ERP-систем Microsoft Dynamics 365 Business Central.

Джерела дослідження

Під час написання курсової роботи було досліджено наукові статті про сучасні патерни програмування різноманітних інформаційних систем. На етапі

аналізу було вивчено аналоги, проведено аналіз різноманітних бібліотек, найкращих підходів та обраних технологій для реалізації схожого функціоналу.

Структура роботи

Курсова робота складається з анотації, вступу, трьох основних розділів, висновків, списку використаних джерел та додатків.

Розділ 1. Аналіз предметної області

1.1. Огляд ERP-системи Business Central від Microsoft

У цій курсовій роботі я працюватиму з ERP-системою від компанії Microsoft – Dynamics 365 Business Central. Планування ресурсів підприємства (Enterprise Resource Planning – ERP) відноситься до типу програмного забезпечення, яке організації використовують для управління повсякденною діловою діяльністю, як-от бухгалтерський облік, закупівлі, управління проектами, управління ризиками та дотримання вимог, а також операції з ланцюга поставок. Повний пакет ERP також включає управління продуктивністю підприємства, програмне забезпечення, яке допомагає планувати, складати бюджет, прогнозувати та звітувати про фінансові результати організації[1].

Системи ERP об'єднують безліч бізнес-процесів і забезпечують потік даних між ними. Збираючи спільні транзакційні дані організації з кількох джерел, системи ERP усувають дублювання даних і забезпечують цілісність даних за допомогою єдиного джерела.

ERP — це повні інтегровані платформи, локальні або в хмарі, які керують усіма аспектами виробничого або розподільчого бізнесу. Ці інтегровані системи діють як центральний центр бізнесу для наскрізного робочого процесу та даних, надаючи доступ різноманітним відділам.

Для розробки кастомізацій для Business Central є окреме середовище розробки, де використовується мова AL.

1.2. Розгортання системи

Microsoft Dynamics 365 Business Central містить різні інструменти та компоненти для керування, розробки, розширення та тестування програми(Додаток А).

Існує кілька способів розгортання цієї ERP-системи та середовища розробки. Можна використовувати докер для контейнерування серверів додатків або просто встановити їх на комп'ютер традиційним способом, що і буде розглянуто у цьому розділі.

Спочатку потрібно встановити SQL server та SQL Server Management Studio. Після цього завантажте потрібну збірку з офіційного сайту Microsoft, про різні версії описано у розділі 1. Коли завантаження завершиться, запуститься програма інсталяції. Якщо ви встановлюєте Business Central для розробки, обов'язково потрібно обрати режим Developer. В іншому випадку буде надано доступ лише до інтерфейсу системи.

Коли ВС встановлюється, автоматично створюється тестова база даних. Якщо інсталяція завершується успішно, створюється новий екземпляр Microsoft Dynamics 365 Business Central Серверу, назва якого вказується в налаштуваннях запуску. Для керування сервером можна використовувати програму Business Central Administration, яка встановлюється разом із системою. З її допомогою можна запускати та зупиняти екземпляри серверу та налаштовувати різні компоненти.

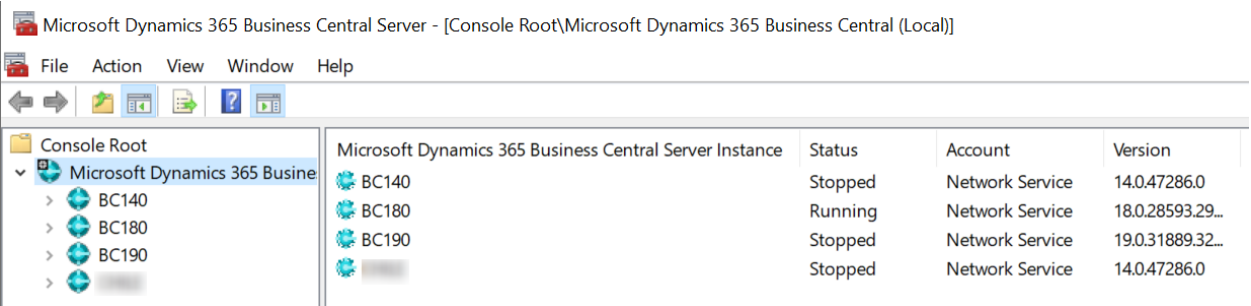


Рисунок 1.2.1 - Список екземплярів серверу

Зауважте, якщо у вас встановлено декілька версій ВС, то керувати кожною з них можна лише з відповідної версії Business Central Administration. Інакше, отримаємо наступну помилку:

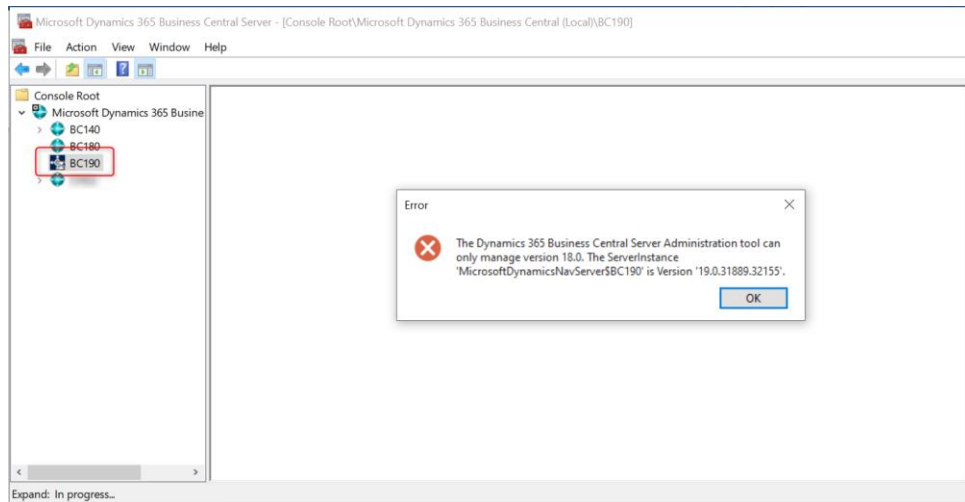


Рисунок 2.2.2 – Повідомлення про помилку невідповідності версій

Якщо сервер запущений, то увійти в систему можна за посиланням: <http://localhost:8080/BC180>, де після номеру порту назва екземпляру сервера. При першій спробі входу в систему, отримаємо таку помилку, оскільки не додано ще жодного користувача.

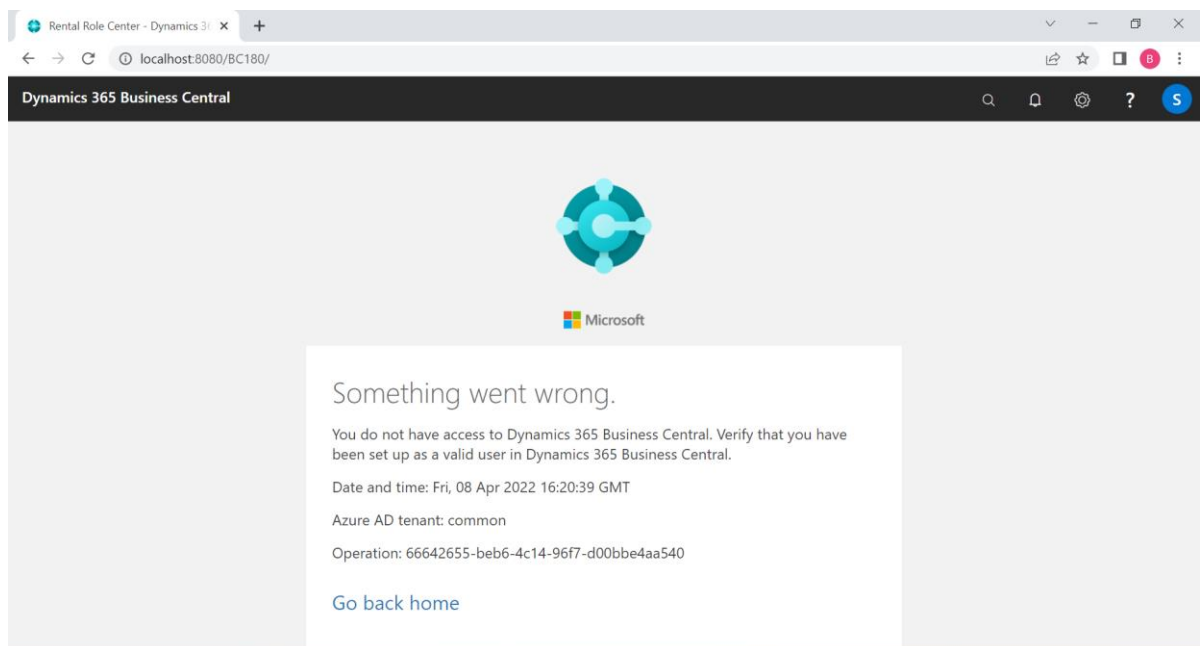


Рисунок 3.2.3 – Повідомлення про відсутність користувача у системі

У Business Central в робочому середовищі користувачі зазвичай додаються через центр адміністрування Microsoft 365. Після створення користувачів у Microsoft 365 їх можна імпортувати з Excel-файлу Користувачі в Business Central. Локальне розгортання Business Central підтримує кілька механізмів авторизації облікових даних для користувачів.

Business Central підтримує такі типи облікових записів:

Тип	Опис
Windows	За допомогою цього типу облікових даних користувачі автентифікуються за допомогою своїх облікових даних Windows. Ви можете вказати Windows як тип облікових даних, лише якщо відповідний користувач існує в Windows. Оскільки вони проходять автентифікацію через Windows, користувачам Windows не буде запропоновано вводити облікові дані під час доступу до Business Central.
NavUserPassword	За допомогою цього параметра автентифікацією керує Business Central Server, але не базується на користувачах Windows або Active Directory. Кожен користувач має ім'я користувача та пароль, які налаштовані лише в Business Central. Під час запуску клієнта користувачу буде запропоновано ввести ім'я користувача/пароль.
UserName	За допомогою цього параметра користувачу буде запропоновано ввести облікові дані імені користувача/пароля під час доступу до Business Central. Ці облікові дані потім перевіряються на автентифікацію Windows за допомогою Business Central Server. У Windows уже має бути відповідний користувач.
AccessControlService	З цим параметром Business Central покладається на Azure Active Directory (Azure AD) для служб аутентифікації користувачів. Сертифікати безпеки необхідні для захисту передачі облікових даних через глобальну мережу.
None	Для внутрішнього використання в системних сесіях і зазвичай не слід використовувати. Якщо вибрати None, екземпляр Business Central Server не може запускатися.

ExchangeIdentity, TaskScheduler, and Impersonate	Тільки для внутрішнього використання.
---	---------------------------------------

Таблиця 4.2.1 – Типи облікових записів у Business Central

Для доступу до локального сервера використовуватимемо Windows автентифікацію. Для того, щоб додати користувача потрібно виконати наступні кроки:

1. Відкрити Windows PowerShell або Business Central Development Shell.
2. Використати команду

Import-Module "{ {Path to BC} } \ { {Server version} } \Service\NavAdminTool.ps1"

Наприклад, **Import-Module "C:\Program Files\Microsoft Dynamics 365 Business Central\140\Service\NavAdminTool.ps1"**

Ця команда додає стандартні функції для роботи з BC.

Обрати команду New-NAVServerUser та заповнити два обов'язкові поля: ServerInstanse та WindowsAccount – ім'я вашого акаунту.

Також необхідно додати набір дозволів(Permission Set) для користувача. Permission Set — це набір дозволів для конкретних об'єктів бази даних. Усім користувачам потрібно призначити один або кілька наборів дозволів, перш ніж вони зможуть отримати доступ до Business Central. Ми додамо для користувача набір SUPER, який має доступ до усіх об'єктів.

Тепер ми маємо доступ до інтерфейсу системи(Додаток Б).

Розділ 2. Розробка в середовищі BC

2.1. Розробка розширень на основі Microsoft Dynamics 365 Business Central

Для розробки додатків на основі Microsoft Dynamics 365 Business Central використовується мова AL(Application Language).

Для написання коду використовується крос-платформний редактор Visual Studio Code. В редакторі необхідно встановити розширення AL Language з повною підтримкою форматування, підсвічування синтаксису та багатим IntelliSense.

Можна розширити Business Central за допомогою додаткових програм, вертикальних або горизонтальних рішень та інтеграції з іншими продуктами та послугами.

Щоб пристосувати Business Central для певного клієнта, розробляються одне або кілька розширень(extensions), які також називають додатками. Extension — це модель програмування, де функціональність визначається як доповнення до існуючих об'єктів і визначає, чим ці об'єкти відрізняються або змінюють поведінку рішення.

Вся функціональність Business Central закодована в об'єктах. Модель розширення базується на об'єктах; ви створюєте нові об'єкти та розширюєте наявні об'єкти залежно від того, що ви хочете робити вашим розширенням.

Всі об'єкти, написані мовою AL зберігаються у файлах із розширенням .al.

2.2. Об'єкти в Business Central

Для кожної сутності в BC необхідно створити об'єкти з типом table. Таблиці є основними об'єктами, які використовуються для зберігання даних у Dynamics 365 Business Central. Незалежно від того, як дані реєструються в продукті, результати цієї транзакції будуть записані в таблицю.

Структура таблиці складається з чотирьох розділів:

1. Перший блок містить метадані загальної таблиці, наприклад тип таблиці, назву, id.
2. Розділ полів описує елементи даних, які складають таблицю, наприклад їх назву та тип даних, які вони можуть зберігати, опис, видимість тощо.
3. Розділ ключів містить визначення первинних та вторинних ключів, визначених для таблиці.
4. В останньому розділі описані тригери та код, які можуть запускатися в таблиці. Для таблиць існує чотири основні тригери: OnInsert – відпрацьовує під час створення нового запису, OnRename – зміна значення ключа, OnModify – модифікація будь-якого поля, OnDelete – видалення.

Для більшості документів(замовлень) створюється одразу 2 таблиці: заголовок та лінії. В заголовку зберігається інформація, яка стосується усіх елементів в замовленні: номер документа, дати, контакти клієнтів тощо. Для кожного елементу у документі(товару, транзакції тощо) створюється окремий запис у таблиці ліній.

Сторінки(Pages) є основним способом відображення та організації візуальних даних у Dynamics 365 Business Central. Вони є основним об'єктом, з яким буде взаємодіяти користувач(у системі не відображаються таблиці), і вони мають різну поведінку залежно від типу. Сторінки розроблені незалежно від пристрою, на якому вони мають відображатися, і таким чином одну й ту саму сторінку можна повторно використовувати на телефонах, планшетах та веб-клієнтах.

Структура сторінки є ієрархічною і розбивається на три розділи. Перший блок містить метадані для всієї сторінки; тип сторінки та вихідну таблицю, з якої відображаються дані. Наступний розділ – макет, описує візуальні частини на сторінці – поля, які відображатимуться, додаткові частини. Останній розділ

детально описує дії, які публікуються на сторінці: кнопки у верхній частині сторінки.

Іншим елементом у Business Central є Codeunit. Це контейнери для коду, який може використовуватися у багатьох різних місцях. Можна викликати функції кодюнітів у тих місцях, де вам потрібна виконати певні бізнес-функції.

Рольові центри(Role Centers) зазвичай створюються для кожної користувацької групи окремо. Це головна сторінка системи, які відкривається при запуску системи. На цій сторінки можна помістити усі потрібні посилання для певного користувача. Наприклад, для менеджера з продажів будуть видимі посилання на замовлення покупки\продажу, створення нових замовлень, для фахівця з транспортних засобів – все, що пов'язано з автомобілями тощо.

Як уже згадували раніше, ВС має набір стандартних об'єктів, проте у нових версіях ERP-системи клієнти не мають дозволу модифікувати ці об'єкти. Для цього необхідно створити їх розширення: table extension або page extension.

2.3. Розробка розширення. Розробка користувацького функціоналу

У цій роботі розроблено розширення на основі стандартного функціоналу ВС для автомобільної компанії, яке містить наступні функції:

- купівля транспортних засобів;
- продаж транспортних засобів;
- створення(реєстрація) нових транспортних засобів;
- створення та редагування класифікації т.з.;
- керування робітниками компанії;
- бронювання т.з.;
- друк усіх відповідних документів.

Для цього створенні основні сутності:

- транспортний засіб – товар;

- транспортний засіб за VIN кодом;
- ACRISS класифікація;
- ECS класифікація;
- замовлення на бронювання(заголовок та лінії).

Повний цикл робочого процесу:

Створення нового товару. Новий новий екземляр сутності Item(27). Item – основна одиниця процесу Купівлі-Продажу. Для того, щоб відрізнити транспортні засоби від інших товарів, які можуть бути у компанії, ми використаємо поле "Item Category Code". Також додамо до таблиці розширення "Vehicle Item"(50000) – об'єкт tableextension. Об'єкт розширення таблиці дозволяє додавати додаткові поля або змінювати деякі властивості стандартної таблиці Dynamics 365 Business Central. Таким чином, можна додати дані до існуючої таблиці та розглядати її як єдину таблицю. Розширення включає наступні поля: ACRISS Code, Maker.

Список усіх транспортних засобів можна переглянути на окремій сторінці Vehicle Items. При натиску на номер товару або кнопку «Переглянути» відкривається сторінка товару з детальним описом та полями з таблиці Item.

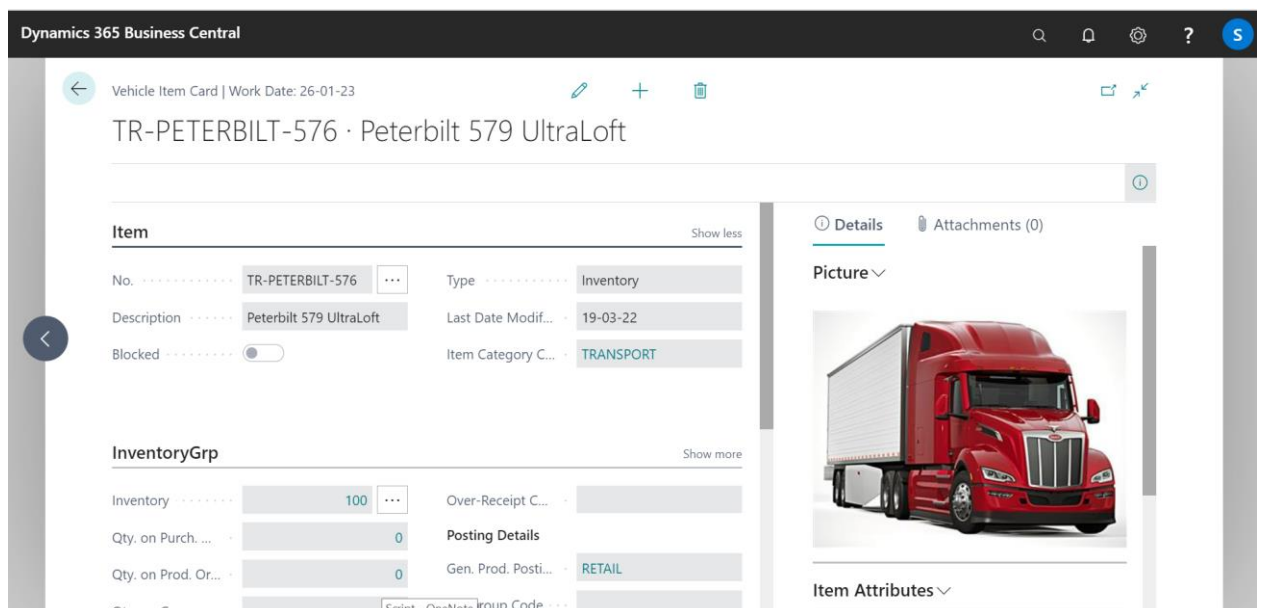


Рисунок 2.3.1 – Сторінка "Vehicle Item Card"

Класифікація. У цьому додатку додано 2 типи класифікації: ACRISS та ECS(Euro Car Segment).

The Association of Car Rental Industry Systems Standards (ACRISS) є галузевим органом компаній з прокату автомобілів у Європі, на Близькому Сході та в Африці. Код класифікації автомобілів був створений, щоб дозволити компаніям з прокату автомобілів повідомляти характеристики автомобіля за допомогою стандартизованої системи, щоб запобігти оманливій інформації під час бронювання автомобіля напрокат. Код описує категорію, тип кузова, трансмісію, колеса (FWD, RWD або AWD), тип палива (або електричний/гібридний автомобіль).

Створено окремі чотири непередані таблиці для кожного елемента класифікації, які заповнюються один раз після створення та використовуються у таблиці, де реєструється код з 4-ох символів для кожного транспортного засобу. Для кожної таблиці створена окрема таблиця з типом – Список.

ACRISS Type | Work Date: 26-01-23

<ul style="list-style-type: none"> > ACRISS Type > ACRISS Category > ACRISS Fuel / Air-con > ACRISS Classification > ACRISS Trans / Driven wheels 	<div style="border: 1px solid #ccc; padding: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; margin-bottom: 5px;"> 🔍 Search ➕ New ✎ Edit List 🗑 Delete 📄 Open in Excel </div> <table> <thead> <tr> <th style="text-align: left;">Code ↑</th> <th style="text-align: left;">Name</th> </tr> </thead> <tbody> <tr style="background-color: #e0f2f1;"> <td>→ B</td> <td>2-3 Door</td> </tr> <tr> <td>C</td> <td>2/4 Door</td> </tr> <tr> <td>D</td> <td>4-5 Door</td> </tr> <tr> <td>L</td> <td>Limousine/Sedan</td> </tr> <tr> <td>S</td> <td>Sport</td> </tr> <tr> <td>V</td> <td>Passenger Van</td> </tr> <tr> <td>W</td> <td>Wagon/Estate</td> </tr> </tbody> </table> </div>	Code ↑	Name	→ B	2-3 Door	C	2/4 Door	D	4-5 Door	L	Limousine/Sedan	S	Sport	V	Passenger Van	W	Wagon/Estate
Code ↑	Name																
→ B	2-3 Door																
C	2/4 Door																
D	4-5 Door																
L	Limousine/Sedan																
S	Sport																
V	Passenger Van																
W	Wagon/Estate																

a)
b)

Рисунок 2.3.2. а – Список сторінок для ACRISS класифікації; б – Перелік значень компоненти Type

🔍 Search + New Edit List 🗑 Delete 📄 Open in Excel 🔍							
Item No. ↑		Category	Type	Trans / Driven wheels	Fuel / Air-con	Full Code	
→ TEST2	⋮	C	S	A	Q	CSAQ	
TR-PETERBILT-576		F	W	C	D	FWCD	

Рисунок 2.3.3 – Приклад конфігурації ACRIS-коду

Налаштувати ці 2 типи класифікації можна на сторінці товару транспортного засобу, натиснувши на відповідні кнопки.

Для того, щоб додати кнопку на сторінку, необхідно додати елемент action у одну з чотирьох секцій:

1. Promoted action categories – відображаються, як вкладка Home;
2. Actions;
3. Navigate – пов’язані сторінки або архівні записи;
4. Report – друкування та перегляд текстових або Excel-документів.

Всередині триггеру OnAction() описуються функції, які будуть викликані, коли користувач натисне на кнопку.

Купівля транспортних засобів у постачальників. Необхідно створити замовлення на купівлю(Purchase Order) та фіналізувати його. Для цього необхідно обрати постачальника, товар або декілька різних товарів, який бажаємо придбати, та оплатити замовлення. Після цього замовлення переміститься в архів, а товар з’явиться в наявності у нашій компанії.

Реєстрація транспортних засобів за VIN кодом. Після того, як автомобіль з’являється у наявності, його необхідно зареєструвати. Кожен автомобіль має свій унікальний VIN код. Для їх обліку використовується сутність Vehicle(50007). Користувач має ввести код автомобіля, як товару(“Item No.”) та VIN-код з 17-ти символів.

Після цього можна ввести інші параметри вручну або натиснути кнопку Decode VIN. Ця кнопка викликає функціонал з кодуюниту Rental Utils, який

формує GET-запит до зовнішнього API ресурсу, щоб отримати інформацію про транспортний засіб. Якщо такий VIN-код існує, та ми отримали успішну відповідь від сервера, ми парсимо JSON-файл, який надійшов у відповіді та заповнюємо поля у таблиці отриманими значеннями.

Прокат\оренда. Для того, щоб орендувати транспортний засіб, клієнт створює замовлення на прокат(Rental Order), де вказує, які саме т.з. хоче взяти, дату і час початку і закінчення оренди. Після заповнення обов’язкових полів автоматично обраховується вартість замовлення та з’являється можливість роздрукувати PDF-документ.

Для зручності було створено новий рольовий центр, звідки можна перейти до усіх створених сторінок, а також елементів, які можуть знадобитися менеджеру під час проходження основного робочого циклу. Для того, щоб цей Рольовий центр був видимим для певної групи користувачів та відкривався одразу ж при запуску системи, необхідно призначити його на сторінці профілю(Profiles).

2.4. Business Central API-інтеграція зі сторонніми продуктами

У попередньому розділі було розроблено розширення у Business Central, додано користувацький функціонал та створення і бронювання транспортних засобів. Тепер необхідно підготувати систему для інтеграції з іншою частиною додатку – голосовим помічником.

Веб-сервіси – стандартний для підприємств спосіб зробити функціонал Business Central доступним для зовнішніх систем і користувачів. Business Central підтримує три типи веб-сервісів: API, SOAP та OData. Для протоколів веб-служб можна створити додатки з типом Page, Codeunit або Query. Сторінки(Pages) можуть використовуватися в усіх трьох протоколах для виконання CRUD операцій. Кодюніти підтримуються лише протоколами SOAP та OData, а запити(Queries) доступні лише в OData і API веб-сервісах в Read-only режимі. Коли об’єкти Business Central, а саме Pages, Codeunits та

Queries, публікуються як веб-сервіси, вони відразу стають доступними в мережі.

Для інтеграції з розробленим у цій роботі розширенням використовуються API веб-сервіси. Стек API Business Central оптимізовано для продуктивності і є кращим способом інтеграції з Business Central. ERP-система постачається з певним набором вбудованих API, які доступні за посиланням <https://<base URL>:<port>/<serverinstance>/api/beta>.

Щоб розпочати роботу з веб-сервісами, необхідно через додаток Business Central Administration увімкнути служби OData та API.

Для того, щоб мати доступ до веб-сервісів на локальному сервісі з Windows автентифікацією треба також увімкнути NTLM-автентифікацію. Windows NT LAN Manager (NTLM) — це протокол автентифікації виклик-відповідь, який використовується для автентифікації клієнта на ресурсі в домені Active Directory.

Для того, щоб зробити нові сутності видимими для python додатку, необхідно опублікувати їх, як веб-сервіси. Для кожної сутності створено окрему сторінку з типом API. Розберемо на прикладі елементу Vehicle. Необхідно вказати наступні характеристики:

```
APIGroup = 'rental';
APIPublisher = 'kuziv';
APIVersion = 'v1.0';
Caption = 'vehicleEntity';
DelayedInsert = true;
EntityName = 'vehicle';
EntitySetName = 'vehicles';
PageType = API;
SourceTable = Vehicle;
```

APIVersion, APIPublisher, APIGroup, EntityName і EntitySetName впливають на користувацьку кінцеву точку:

[https://api.businesscentral.dynamics.com/v1.0/<user domain name>/api/<API publisher>/<API group>/<API version>/companies\(<company id>\)/vehicles](https://api.businesscentral.dynamics.com/v1.0/<user domain name>/api/<API publisher>/<API group>/<API version>/companies(<company id>)/vehicles).

EntitySetName і EntityName встановлюють назву множини і єдину назву об'єкта, з якою сторінка відображатиметься в кінцевій точці API.

Як таблиця-джерело в нас вказана Vehicle, це означає, що на цій сторінці доступні усі поля, які визначені на рівні таблиці. Якщо потрібно додати додаткові поля, необхідно спочатку їх описати в таблиці.

Встановлюємо для властивості ODataKeyFields значення SystemId. Поле SystemId — це поле типу даних GUID, яке визначає унікальний незмінний ідентифікатор (тільки для читання) для записів у таблиці. Ми можемо використовувати для API сторінки й інші поля для визначення ключів, наприклад, первинний ключ таблиці, проте це може спричинити проблему, коли запис було перейменовано в іншому додатку. Тому рекомендують, для ODataKeyFields визначати незмінне унікальне системне поле.

На сторінку ми можемо додати всі поля або лише ті, які нам необхідно зробити доступними для сторонніх систем. Також можна визначити права на операції створення, редагування або видалення для кожної сторінки.

Після того, як ми опублікуємо цю сторінку в BC, вона стане доступною для CRUD-операцій. Сторінка, яку ми розглядали, буде доступною за адресою: [http://localhost:7048/BC140/api/kuziv/rental/v1.0/companies\({{companyId}}\)/vehicles](http://localhost:7048/BC140/api/kuziv/rental/v1.0/companies({{companyId}})/vehicles) у локальному середовищі.

За аналогією створюємо API сторінки для інших сутностей.

Через API-запити можна не лише додавати та змінювати дані у таблиці, а й викликати різні функції. Для цього необхідно створити нову процедуру у третій частині API-сторінки з тегом [ServiceEnabled]. Після публікації джерела, функцію можна викликати за допомогою запиту з методом POST та url-адресою:

[<baseUrl>/companies\(<company id>\)/<page caption>/Microsoft.NAV.<procedure name>](http://localhost:7048/BC140/api/kuziv/rental/v1.0/companies(<company id>)/<page caption>/Microsoft.NAV.<procedure name>).

При цьому назву функції(procedure name) необхідно вказати з маленької літери.

Тепер, коли API-сторінки опубліковані як веб-сервіси, інші додатки можуть звертатися до цього розширення за допомогою HTTP-запитів. Для тестування запитів перед розробкою голосового інтерфейсу використовувалась програма Postman.

Розділ 3. Розробка голосового інтерфейсу

3.1. Аналіз технологій та ресурсів

Наступною частиною проекту є розробка голосового інтерфейсу, який трансформуватиме звукову інформацію у команди, виконуватиме їх та «озвучуватиме» результат виконання. Голосові інтерфейси користувача (VUI) дозволяють користувачеві взаємодіяти з системою за допомогою голосових або мовних команд. Прикладами VUI є віртуальні помічники, такі як Siri, Google Assistant і Alexa. Основна перевага VUI полягає в тому, що він дає змогу користувачам взаємодіяти з продуктом, зосередивши свою увагу деінде, без використання рук і очей.

Є декілька способів додати голосовий інтерфейс у інший додаток.

Спосіб 1 – Інтеграція з уже існуючим додатком, таким, як Siri, Cortana та Google Assistant. У такому випадку необхідно визначити набір потрібних запитів, які будуть категоризовані у домени.

Наприклад, для Siri існує спеціальний SiriSDK, який містить два фреймворки. Перший містить набір задач, які підтримуються додатком, другий інформує на рівні візуального представлення статус виконання команд. На відміну від Siri, у Google Assistant немає таких строгих правил стосовно дизайну та обмеженого набору команд. Користувачі цього VUI можуть створювати власні команди за допомогою спеціальних API.

Спосіб 2 – Використання спеціальних API та інструментів AI. Найпопулярнішими з них є: Jasper, Melissa, Dialogflow, API.ai та Wit.ai. Ці інструменти можуть перетворювати голосові дані у текст та навпаки, зберігати нотатки, відкривати інші програми тощо.

Спосіб 3 – Розробка власного голосового асистента з визначеним набором клієнтських функцій. Часто для систем необхідний VUI з таким

набором команд, який не підтримується у попередніх двох способах. Єдиним виходом залишається створення власного віртуального помічника.

Перед створенням власного голосового інтерфейсу потрібно насамперед обрати правильну платформу з доступними сервісами, необхідними для забезпечення майбутнього функціоналу. Важливим етапом є визначення вимог, набору доступних команд. Щоб зробити програму швидкою та ефективною потрібно додати лише вадливі для користувача функції.

Розробка голосового користувацького асистенту складається з таких технологій:

- **STT (Voice/Speech to Text)**

Це перетворення голосових сигналів у цифрові дані, наприклад, у текстові дані. Мовлення зазвичай надходить у вигляді потоку даних або звукового файлу.

- **TTS (Text to Speech)**

Це протилежність процесу STT, він трансліює текст в людську мову.

- **NLP(Natural Language Processing)**

NLP – обробка природної мови – ключовий компонент голосового інтерфейсу. За визначенням IBM, обробка природної мови (NLP) відноситься до галузі комп’ютерних наук, а точніше, до галузі штучного інтелекту, яка займається наданням комп’ютерам здатності розуміти текст і вимовлені слова приблизно так само, як і люди[3].

У даному випадку NLP можна розділити на дві частини: Natural language understanding(NLU) та Natural language generating(NLG).

Перший компонент допомагає перетворити текст, сказаний користувачем, у одну з визначених функцій. Наприклад, одну команду “getItemUnitCost”(з англ. – “взяти вартість товару”) можна озвучити багатьма способами:

1. «Яка вартість товару?»
2. «Скільки коштує товар?»
3. «Яка ціна товару?»

4. «Скільки вартує товар?»

NLU надає можливість машині зрозуміти, що це один і той же запит.

NLG відповідає за генерування відповіді на команди. Системи LG можуть перетворювати числа в наративи на основі попередньо встановлених шаблонів.

- **Інтелектуальне тегування & Прийняття рішень**

Інтелектуальне тегування використовується для інтерпретування запитів клієнта та виконання відповідних дій. Для прикладу, користувач питає, що йому приготувати на вечерю. Тоді ця технологія має позначити усі найпопулярніші страви та рецепти відповідно до вподобань користувача.

- **Звуковий інтерфейс**

Це представлення того, що користувачі чують або бачать, як відповідь на свій запит. Інколи це може розширюватися графічним інтерфейсом.

- **Біометрія голосу**

Під час розробки комерційних VUI необхідно звернути увагу на цю технологію. З її допомогою голосовий асистент здатен розрізняти хто саме віддає команди та чи потрібно їх виконувати. Таким чином можна уникнути ситуацій, коли додаток виконує команди, коли чує телевізійні діалоги тощо.

- **Контроль шуму**

Шуми, що надходять від електронних приладів, трафіку, людей поруч тощо, можуть зробити голос користувача нечітким та важким для сприйняття голосовим асистентом. Ця технологія пом'якшує та видаляє фонові звуки, щоб зробити основний потік мовних даних чіткішим.

- **Стиснення звукових сигналів**

Для покращення продуктивності програми голосові дані переформатовуються на стороні клієнта та надсилаються на сервер у більш стислому форматі. Стандарт G.711 ідеально підходить для цього механізму[6].

У цій курсовій виконано розробку технологій STT, TTS, інтелектуального тегування команд, звукового інтерфейсу та частково розглянуто NLP.

3.2. Проектування архітектури та розробка додатку

Для реалізації додатку обрано мову програмування частина Python через наявність багатьох бібліотек для роботи з текстом та звуком, а також для можливості подальшого удосконалення з використанням NLP, для якого ця мова наразі є найкращим вибором. Розробка VUI у межах цієї роботи відбувалася за такими ітераціями:

1. Трансформація голосових даних у текст;
2. Перетворення тексту у звук;
3. Створення коротких команд у вигляді чат бота для формування повного запиту;
4. Створення звукового інтерфейсу;
5. Розбивка тексту на команди та параметри;
6. Імплементації NLP.

Перший етап – розпізнавання мовлення – це міждисциплінарна галузь NLP, що розробляє методології та технології для трансформації розмовних даних у текст. Для перетворення голосу у текст – використовується бібліотека `speech_recognition`. Бібліотека `speech_recognition` діє, як обгортка для багатьох популярних мовленнєвих API і тому надзвичайно гнучка. До того ж ця бібліотека має вбудовані функції для доступу до мікрофону та обробки звукових файлів, тому немає необхідності писати додаткові функції для цього.

Перш за все, ініціалізуємо клас розпізнавання. Використовуємо мікрофон, як джерело звуку та записуємо отримані звідти дані:

```
recognizer = sr.Recognizer()
with sr.Microphone() as source:
    audio_text = recognizer.listen(source)
```

Рисунок 3.2.1 – Частина коду для STT-технології

Бібліотека `speech_recognition` підтримує різні системи розпізнавання мовлення та API, серед яких: CMU Sphinx (працює оффлайн), Google Speech Recognition, Google Cloud Speech API, Wit.ai, Microsoft Bing Voice Recognition, Houndify API та IBM Speech to Text. Оскільки `SpeechRecognition` постачається з ключем API за умовчанням для Google Web Speech API, ми використовуватимемо саме його. Інші шість API вимагають аутентифікації за допомогою ключа API або комбінації імені користувача/пароля. Ця система також підтримує більшість поширених мов, серед яких і українська.

Є декілька способів трансформувати текст в аудіо. Я використовую Google Text to Speech API, широко відомий як gTTS. З його допомогою можна зберегти файл у форматі mp3 різними мовами та з декількома варіантами швидкості відтворення. Після цього збережений файл відтворюється через команди бібліотеки `os` - інтерфейси операційної системи.

У межах цієї курсової роботи розроблено набір основних команд для робочого процесу у розширенні, створеного у попередньому розділі.

Прийняття рішень відбувається за наступним алгоритмом:

1. Вказівка, сказана користувачем та перетворена у текст, зводиться за вмістом певних слів до однієї конкретної команди;
2. Система перевіряє, чи є ця команда у її наборі «вмінь»;
3. Якщо така команда існує, то викликається функція, відповідальна за її виконання. Ця функція може містити певну бізнес-логіку, до прикладу, перевіряє, чи є усі потрібні параметри для надіслання запиту.

4. Після забезпечення усіх вимог, викликається інший функціонал, куди передаються усі потрібні параметри. Ця частина створює HTTP-запит, надсилає його, та повертає результат.
5. Тепер, аналізуючи код статусу відповіді та отриманий результат, формується відповідь голосового інтерфейсу.

Я використовую 3 основні методи HTTP запитів для виконання CRUD-операцій та виклику функцій Business Central, а саме:

- GET – в поєднанні з різними параметрами для Read-операцій, для отримання актуальних даних з таблиць. Наприклад, для відповіді на питання «Скільки товару X є в наявності?», надіслати GET-запит на API сторінку items із параметром “filter = ItemNo eq ‘X’”. Використання фільтрації безпосередньо в запиті до системи значно пришвидшує час виконання програми, оскільки не потрібно робити додаткових дії на стороні додатку.
- POST – використовується для створення нових записів у таблицях та виклику процедур на API сторінках. Для вставлення запису необхідно додати тіло запиту у вигляді JSON-об’єкту.

```
body = {
    "number": ItemNo,
    "displayName": description,
    "type": "Inventory",
    "itemCategoryCode": "TRANSPORT",
    "blocked": "false",
    "priceIncludesTax": "false"
}

response = requests.post(url, json = body, auth = HttpNtlmAuth(userName,password))
return True if ResponseSuccess(response) else False
```

Рисунок 3.2.2 – Фрагмент коду для запиту з методом POST

При додаванні нових записів через API веб-сервіси, виконується вся валідація, визначена на полях та тригерах сторінки.

- PATCH – для оновлення уже існуючих записів.

Оскільки майже усі команди потребують певних параметрів, то для початку було розроблено взаємодію з користувачем у вигляді боту з

короткими вказівками. Наприклад, команда «Create new Vehicle Item CANTER 3C15 with description ‘Mitsubishi Fuso Canter standard tipper van’, which has unit price 13 500 euro and inventory – 10 pieces.» («Створи новий транспортний засіб CANTER 3C15 з назвою ‘Mitsubishi Fuso Canter standard tipper van’ вартістю 13 500 євро у наявності 10 штук») буде розбито на такі частини:

1. Яка дія? – Створити новий транспортний засіб.
2. Який код товару? - CANTER 3C15.
3. Який опис? – ‘Mitsubishi Fuso Canter standard tipper van’.
4. Яка ціна? - 13 500 євро.
5. Скільки штук у наявності? – 10.

У цьому прикладі вказані обов’язкові поля для створення нового товару. В залежності від кастомізацій кількість параметрів змінюватиметься.

3.3. NPL-імплементация та створення моделі LUIS

Подальша розробка фокусуватиметься здебільшого на імплементации штучного інтелекту, а саме NLP, всередину додатку.

Існує декілька способів виконати цю частину проекту. Наприклад, можна самостійно реалізувати базовий конвеєр команд(pipeline) NLP і перетворити його в потрібний формат, що буде передано, як вхідні дані для будь-якого алгоритму машинного або глибокого навчання.

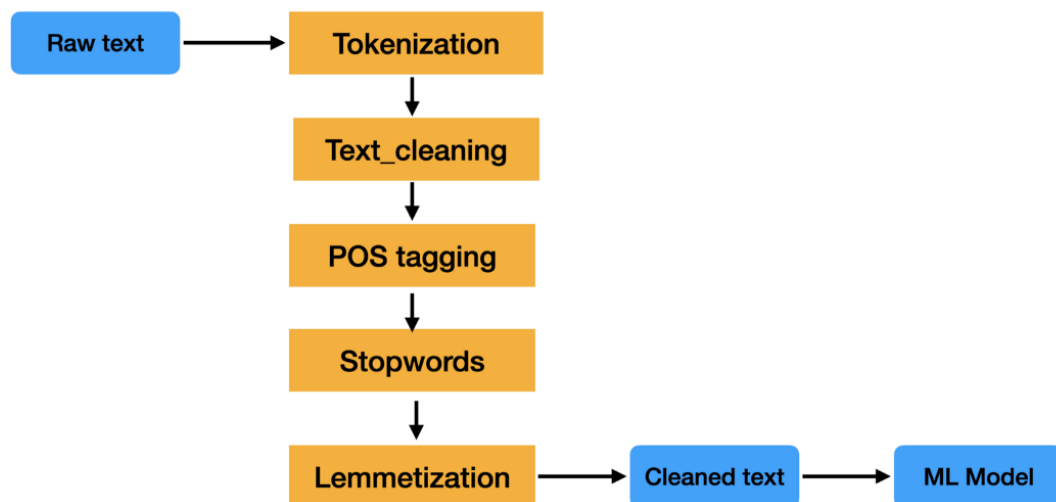


Рисунок 3.3.1 – Базовий конвеєр команд NLP

Як показано на фігурі вище, є п'ять основних кроків у пайплайні NLP. Звичайно, є набагато більше проміжних елементів, такі, як кореферентна роздільна здатність, розпізнавання іменованих сутностей тощо, проте на початковій стадії важливими є саме цих 5 частин.

Перший етап – токенизація – поділ тексту на менші частини, які називаються токенами. Це може бути виконано на рівні речень або слів.

Далі відбувається очистка тексту за допомогою регулярних виразів або вже готових бібліотек. Числа, великі літери, лапки та знаки пунктуації видаляються з токенів, щоб допомогти підвищити ефективність моделі машинного навчання.

Наступна фаза - POS(Part-Of-Speech)-тегування – позначає кожне слово в реченні відповідною частиною мови на основі як його визначення, так і його контексту.

Після цього ідентифікуються та видаляються усі стоп-слова. Стоп-слова – це слова, які не додають багато цінності сенсу загального тексту. Стоп-слова поширені в усіх мовах. Деякі бібліотеки, наприклад, Nltk для Python, містять перелік стоп-слів для багатьох поширених мов, таких як: англійська, німецька, данська і так далі. Перевагою виконання цього кроку насамперед є те, що розмір датасету зменшиться і таким чином зменшиться час, необхідний для навчання моделі МН. Також це може підвищити продуктивність і точність класифікації, оскільки залишається менше токенів, але з більшим смисловим навантаженням.

На останньому етапі проводиться лематизація, тобто усі слова зводяться до їх кореневих слів в залежності від контексту. Уже згадана раніше бібліотека Nltk містить WordNet Lemmatizer, який використовує базу даних WordNet для пошуку лем слів.

Після виконання цих кроків текст буде у потрібному форматі, який можна передавати як вхідні дані для моделі ML.

Інший спосіб додати AI полягає у використанні уже готових сервісів, як-от LUIS. Language Understanding (LUIS) – це хмарний сервіс розмовного

штучного інтелекту, який застосовує машинне навчання до тексту натуральної мови, щоб передбачити загальне значення та надати релевантну інформацію.

Процес створення моделі LUIS можна побачити на наступному зображенні:

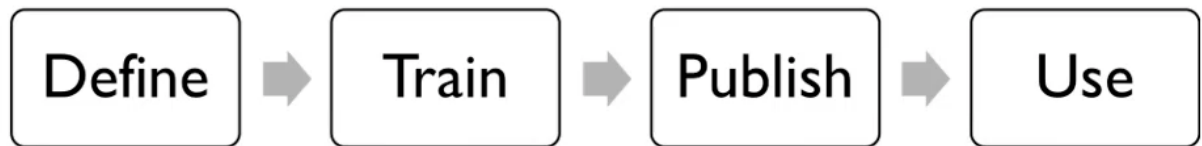


Рисунок 3.3.2 – Процес створення моделі LUIS

Для початку необхідно створити новий застосунок на офіційному сайті, використовуючи акаунт Microsoft. Основним кроком є визначення інтентів(intents – з англ. - наміри). Інтенти відображають завдання або дії, які користувач хоче виконати.

Intents ⓘ	
+ Create + Add prebuilt domain intent ≡ Rename 🗑 Delete 🔍	
Name ↑	Examples
buy	3
create	3
find	12
modify	3
None	0
sell	5

Рисунок 3.3.3 – Список визначених інтентів

Для кожного наміру необхідно додати декілька висловлювань – прикладів вхідних даних від користувача. Вони необхідні для тренування моделі розпізнавання. Важливою є різноманітність висловлювань, потрібно вказати приклади з синонімами, різним розміщенням слів, довжиною, пунктуацією тощо.

Confirm all entitiesMove toDelete

View options

<div>what is customer 10000s due balance</div> <div><div>Table</div><div>keyPhrase</div><div>Field</div><div>keyPhrase</div></div>	0.957	0.01	create	0.95
<div>find customer 10000 ' s balance</div> <div><div>Table</div><div>keyPhrase</div><div>Custo...</div><div>Vendor</div><div>number</div><div>keyPhrase</div><div>Field</div></div>	0.979	0.01	create	0.97
<div>show me customer 10000 ' s balance</div> <div><div>Table</div><div>keyPhrase</div><div>Custo...</div><div>Vendor</div><div>keyPhrase</div><div>Field</div></div>	0.980	0.01	modify	0.97

Рисунок 3.3.4 – Приклади висловлювань для інтенту find

На цьому етапі можна також створити сутності, які будуть використані у намірах. Вони визначають дані, які можуть бути виокремлені з висловлювань. Цей крок не обов’язковий, але дуже рекомендується його виконати, оскільки це покращує результати.

Entities

+ Create

Add prebuilt entity

Add prebuilt domain entity

Renam

Name ↑	Type
Customer	List
datetimeV2	Prebuilt
email	Prebuilt
Field	List
geographyV2	Prebuilt
Item	List
keyPhrase	Prebuilt
money	Prebuilt

Рисунок 3.3.5 – Список визначених сутностей

Тепер можна переходити до тренування та тестування моделі. Усі алгоритми уже описані для цього і доступні просто у вигляді двох кнопок “Train” та “Test”. Спочатку необхідно запустити тренування, що може зайняти певний час в залежності від розміру вхідних даних на попередньому етапі. Після цього можна до тесту. Ці процеси зазвичай є повторюваними, доки

результати тестування не будуть задовільними. Тренуємо, тестуємо, вносимо зміни і знову тренуємо.

Коли застосунок готовий і не потребує інший покращень, його треба опублікувати, щоб зробити доступним для інтеграції з іншими додатками. Після публікації на головній сторінці з'явиться адреса кінцевої точки цієї моделі. Тепер її можна використовувати у інших програмах за допомогою GET-запитів на цю адресу у комбінації з ідентифікатором додатку та запитом, який треба розпізнати.

У відповідь надійде JSON-файл з переліком інтенів з коефіцієнтом співпадіння у порядку спадання та використані сутності.

Це лише 2 варіанти можливості імплементації штучного інтелекту у голосовий інтерфейс. Технології варіюються в залежності від обраної платформи, мови програмування, системи, з якою необхідно інтегровувати та набором підтримуваного функціоналу.

Висновки

У цій курсовій роботі було виконано розробку розширення для менеджменту(реєстрації, купівлі\продажу, оренди) транспортних засобів на основі Microsoft Dynamics 365 Business Central, а також зроблено проектування та розробку голосового користувацького інтерфейсу з набором команд для керування цим розширенням.

На першому етапі було проведено ґрунтовний аналіз предметної області та зібрано інформацію про ERP-систему Microsoft Dynamics 365 Business Central. У процесі було встановлено та розгорнуто систему локально; усі кроки та особливості цього процесу описано у Розділі 1.

Після ознайомлення з функціоналом Business Central було вивчено основні аспекти розробки у цьому середовищі, а саме синтаксис та правила мови AL та властивості основних об'єктів. Було визначено основні функціональні вимоги до майбутнього розширення та основні сутності, які беруть участі у процесі. Результатом цієї частини є реалізація повноцінного робочого циклу: від створення та реєстрації нового автомобіля до оформлення для нього замовлення на оренду.

Також у Розділі 2 було проаналізовано та описано можливості інтеграції Microsoft Dynamics 365 Business Central із сторонніми додатками. Створено API сторінки та функції, які потрібні для взаємодії з голосовим асистентом.

У останньому розділі наведено перелік та особливості основних технологій, які потрібні для створення голосового користувацького інтерфейсу. На основі цього виконано проектування архітектури та розробку додатку, який складається з таких компонентів: STT, TTS, інтелектуальне тегування, користувацький інтерфейс та NLP. Реалізовано базовий набір команд, необхідний для роботи з функціоналом BC, розробленим у попередньому розділі.

Для подальшої розробки було визначено конвеєр задач NLP, проаналізовано найкращі підходи та технології. Основні етапи та дії при створення моделі LUIS, яка використовується як компонента NLU, також описані у третьому розділі.

Отже, результатом курсової роботи є комплексна розробка розробку розширення для менеджменту(реєстрації, купівлі\продажу, оренди) транспортних засобів на основі Microsoft Dynamics 365 Business Central та голосовий помічник з набором команд для керування цим розширенням.

Список використаних джерел

1. Microsoft Dynamics 365 - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://dynamics.microsoft.com/en-us/business-central/>.
2. Digital voice assistants in use - офіційний ресурс [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.juniperresearch.com/press/digital-voice-assistants-in-use-to-8-million-2023>.
3. Natural Language Processing (NLP) - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.ibm.com/cloud/learn/natural-language-processing>.
4. Microsoft Dynamics 365 Business Central - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://dynamics.microsoft.com/en-us/business-central/overview/>.
5. Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. - O'Reilly Media, 2009 – (ISBN 978-0-596-51649-9). – С. 89–93.
6. G.711 : Pulse code modulation (PCM) of voice frequencies - офіційний сайт [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.itu.int/rec/T-REC-G.711>.
7. Mark Richards. *Software Architecture Patterns*. O'Reilly Media, Inc. 2015.
8. Wilhelm Hasselbring. Component-based software engineering. *International Journal of Software Engineering and Knowledge Engineering*. May 2002.
9. Google DialogFlow. Google Cloud Platform. - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://cloud.google.com/dialogflow/docs/>.
10. LUIS - офіційний ресурс [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.luis.ai/>.

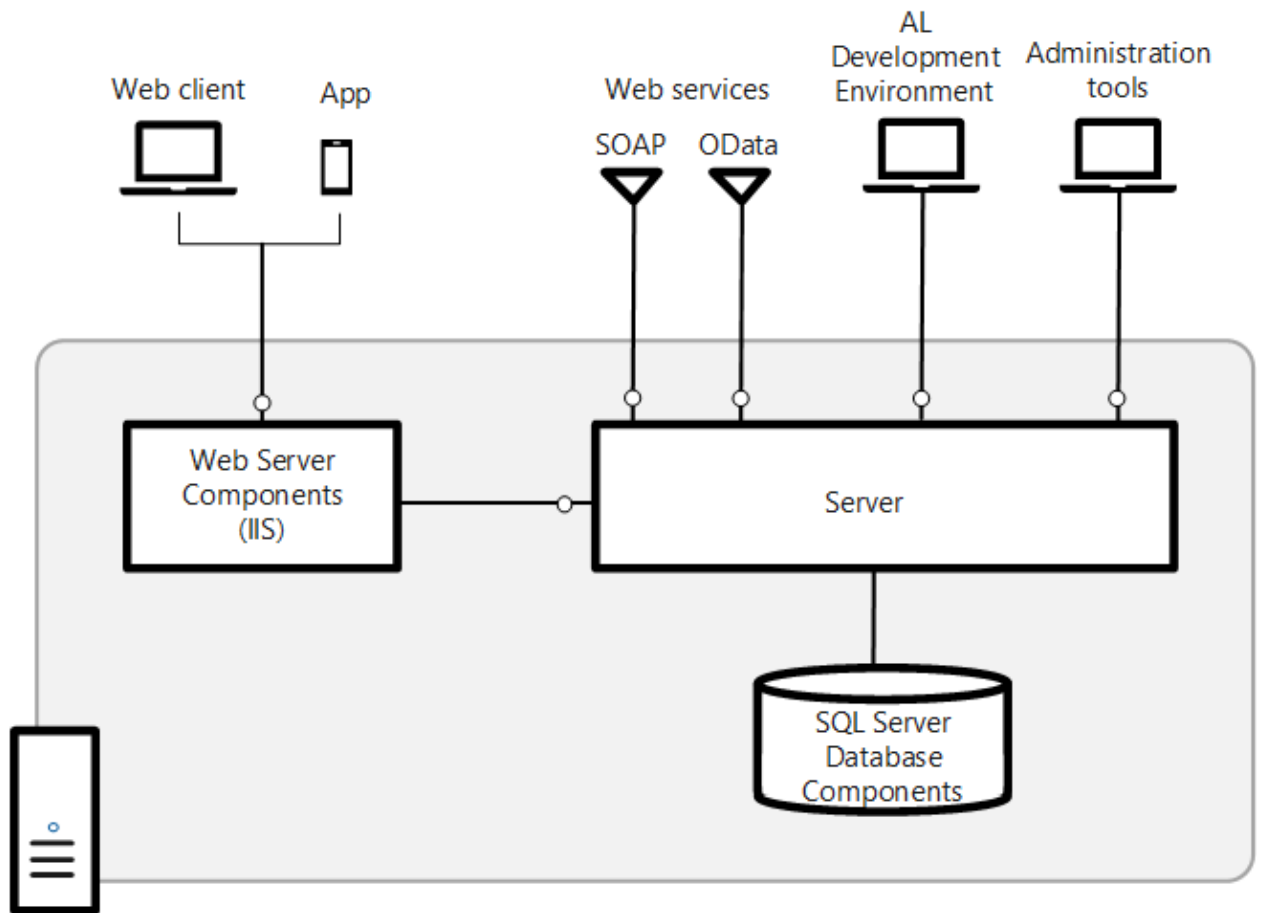
11. Development and Administration for Dynamics 365 Business Central - офіційний ресурс [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/>.
12. Sunny Srinidhi. Lemmatization in Natural Language Processing (NLP) and Machine Learning - [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://towardsdatascience.com/lemmatization-in-natural-language-processing-nlp-and-machine-learning-a4416f69a7b6>.
13. Dr. Monica. Basic Steps In Natural Language Processing Pipeline - [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://medium.com/mlearning-ai/basic-steps-in-natural-language-processing-pipeline-763cd299dd99>.
14. Natalia Kh. 3 Efficient ways to supply your app with a virtual assistant - [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.cleveroad.com/blog/how-to-create-virtual-assistant-apps-like-siri-and-google-assistant>.
15. Heidi Vella. NLG: technology that gives machines a voice - [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://www.raconteur.net/technology/artificial-intelligence/technology-that-gives-machines-a-voice>.
16. Natural Language Processing with Python - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <http://www.nltk.org/book/>.
17. Speech Recognition - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://pypi.org/project/SpeechRecognition/>.
18. STT Engines and TTS Voices - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://cognitivevoice.io/docs/available_STT_Engines_and_TTS_Voices.html
19. NTLM authentication - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.windows-active-directory.com/ntlm-and-kerberos-authetication-protocols>

20. Download 18.5 for Microsoft Dynamics 365 Business Central 2021 Release Wave 1 - офіційний сайт [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.microsoft.com/en-us/download/>.
21. O'Leary, Daniel E. Enterprise Resource Planning Systems Systems, Life Cycle, Electronic Commerce, and Risk. - Cambridge University Press, 2000. - p. 232. - ISBN 978-0-52-179152-6.
22. Demiliani S. Dynamics 365 Business Central Development / S. Demiliani, D. Tacconi. - Packt Publishing, 2018. – 224 с.

Додатки

Додаток А

Багаторівнева топологія системи Microsoft Dynamics 365 Business Central



Додаток Б

Базовий рольовий центр Microsoft Dynamics 365 Business Central

