

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система сервісної підтримки користувачів на базі
модульного чат-бота в Microsoft Teams»

на здобуття освітнього ступеня магістра
зі спеціальності 124 Системний аналіз
освітньо-професійної програми «Інтелектуальні системи управління»

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Олексій ЛИТВИНЕНКО
(підпис)

Виконав: здобувач вищої освіти групи САДМ-61

_____ Олексій ЛИТВИНЕНКО

Керівник: _____ Ровіл НАФСЄВ
к.ф.-м.н., доцент

Рецензент: _____
науковий ступінь, Ім'я, ПРІЗВИЩЕ
вчене звання

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра інформаційних систем та мереж

Ступінь вищої освіти магістр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Інтелектуальні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

«_____» ____20__р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Литвиненко Олексій Вікторович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система сервісної підтримки користувачів на базі модульного чат-боту в Microsoft Teams

керівник кваліфікаційної роботи Ровіл НАВЄЄВ к.ф.-м.н., доцент кафедри ІСТ,
(Ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «_____» _____20__р. № _____

2. Строк подання кваліфікаційної роботи «_____» _____20__р.

3. Вихідні дані до кваліфікаційної роботи: Технічна документація, параметри розробки модульних чат-ботів, результати тестування, методи інтеграції ботів із Microsoft Teams, вимоги до автоматизації обробки заявок і оптимізації внутрішніх сервісів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Що таке сервіс, внутрішній сервіс, система підтримки та їх роль у організаціях.

2. Початкові вимоги до сервісного чат-боту.

3. Розробка архітектури чат-бота з урахуванням потреб компанії.

4. Валідація функціональності чат-бота та аналіз ефективності його використання.

5. Перелік ілюстративного матеріалу: презентація

6. Дата видачі завдання «_____» _____20__р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз проблематики сфери		
2	Підготування даних та матеріалів		
3	Прототипування програми		
4	Реалізація системи		
5	Висновки, презентація		
6	Перевірка роботи на анти плагіат		
7	Передзахист		
8	Захист роботи		

Здобувач(ка) вищої освіти

(підпис)

Олексій ЛИТВИНЕНКО

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Ровіл НАФЄЄВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра:
124 стор., 25 рис., 8 табл., 40 джерел.

Мета роботи – створення модульного чат-бота для інтеграції в корпоративне середовище Microsoft Teams, що дозволяє автоматизувати обробку заявок, підвищити ефективність внутрішніх сервісів, зменшити витрати на підтримку систем і забезпечити зручність користувачів.

Об'єкт дослідження – внутрішні сервіси та інструменти для їх підтримки в корпоративних середовищах.

Предмет дослідження – процес автоматизації обробки заявок і підтримки внутрішніх сервісів за допомогою чат-бота.

Короткий зміст роботи: У роботі досліджено, внутрішні сервіси, їх роль та важливість у організаціях. Проведено аналіз існуючих інструментів для роботи внутрішніх сервісів. Розроблено архітектуру системи як альтернативи готових рішень, що включає інтеграцію з Jira, PostgreSQL, та іншими інструментами. Також розглянуто питання персоналізації функціоналу та масштабованості чат-бота. Практична частина включає впровадження системи у компанії Innovecs та оцінку її ефективності. Досліджено фактори, які вплинули на сприйняття системи користувачами, та запропоновано шляхи вдосконалення. Результати роботи демонструють економічну ефективність і перспективність використання розробленого рішення.

КЛЮЧОВІ СЛОВА: внутрішні сервіси, технічна підтримка, модульний чат-бот, Microsoft Teams, автоматизація, Jira, Python, PostgreSQL, ефективність.

ABSTRACT

Text part of the master's qualification work:

124 pages, 25 pictures, 8 table, 40 sources.

The purpose of the work is to develop a modular chatbot for integration into the corporate Microsoft Teams environment, enabling the automation of ticket processing, increasing the efficiency of internal services, reducing system support costs, and ensuring user convenience.

Object of research is internal services and tools for their support in corporate environments.

Subject of research – is the process of automating ticket processing and supporting internal services using a chatbot.

Summary of the work: The study examines internal services, their role, and importance in organizations. An analysis of existing tools for internal service operations was conducted. The architecture of the system was developed as an alternative to ready-made solutions, including integration with Jira, PostgreSQL, and other tools. The issues of functional personalization and chatbot scalability were also considered. The practical part includes the implementation of the system at Innovecs and an evaluation of its efficiency. Factors influencing user perception of the system were studied, and ways to improve it were proposed. The results demonstrate the economic efficiency and potential of the developed solution.

KEYWORDS: internal services, technical support, modular chatbot, Microsoft Teams, automation, Jira, Python, PostgreSQL, efficiency.

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

**ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра**

Направляється здобувач(ка) Литвиненко О. В. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 124 Системний аналіз
(код, найменування спеціальності)
освітньо-професійної програми Інтелектуальні системи управління
(назва)
на тему: «Система сервісної підтримки користувачів на базі модульного чат-бота в Microsoft Teams».

Кваліфікаційна робота і рецензія додаються.

Директор ННІ ІТ

(підпис)

Катерина НЕСТЕРЕНКО

(Ім'я, ПРІЗВИЩЕ)

Висновок керівника кваліфікаційної роботи

Здобувач Олексій ЛИТВИНЕНКО виконав роботу на високому рівні. Робота має практичну значущість та відповідає поставленим завданням. Здобувач продемонстрував належні знання та вміння у розробці програмного забезпечення.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Олексія ЛИТВИНЕНКА на оцінку «_____» та присвоїти йому кваліфікацію магістр з системного аналізу.

Керівник кваліфікаційної роботи _____

(підпис)

Ровіл НАФЄЄВ

(Ім'я, ПРІЗВИЩЕ)

«____» _____ 20____ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач(ка) Литвиненко О. В. допускається до захисту даної роботи в Експертній комісії.

Завідувач кафедри ІСТ

(назва)

(підпис)

Каміла СТОРЧАК

(Ім'я, ПРІЗВИЩЕ)

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня магістра

здобувача(ки) вищої освіти Литвиненко Олексій Вікторович

(прізвище, ім'я, по батькові)

на тему «Система сервісної підтримки користувачів на базі модульного чат-бота в Microsoft Teams»

Актуальність.

Кваліфікаційна робота є актуальною та відповідає сучасним вимогам до розробки програмного забезпечення. Тема роботи обрана вдало, а запропоноване рішення має високу практичну цінність, що підтверджується можливістю його використання у корпоративних середовищах. Здобувач(ка) демонструє глибоке розуміння теми, належний рівень технічних знань та здатність до вирішення складних завдань. Робота заслуговує на позитивну оцінку.

Позитивні сторони.

1. Практична значущість: Розроблений чат-бот має високу практичну цінність.
2. Економічна ефективність: Рішення дозволяє знизити витрати.
3. Комплексний підхід до розробки: У роботі детально описані всі етапи — від аналізу вимог до впровадження в корпоративну інфраструктуру, що демонструє системний підхід здобувача до вирішення завдання.

Недоліки.

1. Недостатній аналіз альтернатив: У роботі не було проведено детального порівняння з іншими підходами або інструментами, які могли б виконувати подібні функції, що обмежує ширину дослідження.
2. Стислість теоретичної частини: Деякі аспекти, такі як опис сучасних технологій автоматизації або деталі архітектури чат-ботів, викладені недостатньо глибоко.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи магістерської

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "_____", а здобувач(ка) Олексій Литвиненко заслуговує присвоєння кваліфікації магістр з системного аналізу*

Рецензент:

науковий ступінь, вчене звання

підпис

Ім'я, ПРИЗВИЩЕ

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	10
ВСТУП.....	12
РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА	14
1.1. Вступ до сервісних процесів та підтримки користувачів	14
1.1.1 Визначення внутрішнього сервіса та його ролі для організацій.....	14
1.1.2 Актуальність технічної підтримки для сучасних компаній як приклад необхідності внутрішніх сервісів	20
1.2. Проблеми традиційних систем підтримки	23
1.2.1 Короткий огляд існуючих рішень (Jira, Freshdesk, Zendesk) та їх функціональні обмеження у порівнянні з потребами малого та середнього бізнесу	23
1.2.2 Виклики готових систем підтримки	28
РОЗДІЛ 2. KEYС: РОЗРОБКА МОУДЛЬНОГО СЕРВІСНОГО ЧАТ-БОТА ДЛЯ MICROSOFT TEAMS В КОМПАНІЇ INNOVECS.....	32
2.1. Початкові вимоги та постановка завдання.....	32
2.1.1 Jira Service Managment як приклад дорогого рішення для внутрішнього сервісу.	32
2.1.2 Потреба у створенні доступнішої та дешевшої, модульної системи підтримки користувачів.....	35
2.1.3 Визначення чат-ботів, їх типи і використання в корпоративних системах	36
2.1.4 Початкові вимоги: підтримка створення заявок, робота з вкладеннями, нотифікації для користувачів, робота з створеними заявами, підтримка додаткових функцій(паркування) та персоналізація потреб.....	42
2.2. Використані технології.....	44
2.2.1 Python	44
2.2.2 Microsoft Bot Framework SDK	49
2.2.3 Microsoft Teams	52

2.2.4 Jira	56
2.2.5 PostgreSQL	58
2.2.6 Azure	62
2.2.7 Zoho Survey	65
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА: РОЗРОБКА ТА ВПРОВАДЖЕННЯ МОДУЛЬНОГО ЧАТ-БОТА	68
3.1. Архітектура та функціональні можливості модульного чат-бота	68
3.1.1 Основні сценарії взаємодії	68
3.1.2 Опис кожної функції бота	71
3.1.3 Детальне описання архітектури бота	86
3.2. Впровадження бота в компанію	92
3.2.1 Інтеграція бота в корпоративне середовище Teams	92
3.2.2 Інтеграція з Jira	94
3.3. Тестування та результати впровадження	96
3.3.1 Порівняння за витратами та функціональністю	96
3.3.2 Практичні результати	98
3.3.3 Оцінка зручності для кінцевих користувачів	101
ВИСНОВКИ	102
ПЕРЕЛІК ПОСИЛАНЬ	103
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	106

ПЕРЕЛІК СКОРОЧЕНЬ

RDBMS – Relational Database Management System (Система управління реляційними базами даних).

SQL – Structured Query Language (Мова структурованих запитів).

ACID – Atomicity, Consistency, Isolation, Durability (Атомарність, Узгодженість, Ізольованість, Надійність).

AI – Artificial Intelligence (Штучний інтелект).

ERP – Enterprise Resource Planning (Планування ресурсів підприємства).

ABC – Abstract Base Classes (Абстрактні базові класи).

ML – Machine Learning (Машинне навчання).

TIOBE – Індекс популярності мов програмування.

SSAS – SQL Server Analysis Services (Служби аналізу даних SQL Server).

L1, L2, L3 – Рівні технічної підтримки: L1 (перший рівень), L2 (другий рівень), L3 (третій рівень).

OOTB – Out-of-the-box (Готове рішення, не вимагає налаштувань).

NLP – Natural Language Processing (Обробка природної мови).

NAT – Network Address Translation (Переклад мережевих адрес).

HIPAA – Health Insurance Portability and Accountability Act (Закон США про захист медичної інформації).

ISO – International Organization for Standardization (Міжнародна організація зі стандартизації).

SaaS – Software as a Service (Програмне забезпечення як послуга).

PaaS – Platform as a Service (Платформа як послуга).

IaaS – Infrastructure as a Service (Інфраструктура як послуга).

API – Application Programming Interface (Інтерфейс програмування застосунків).

MVCC – Multi-Version Concurrency Control (Багатоверсійне управління конкурентним доступом).

JSON – JavaScript Object Notation (Текстовий формат обміну даними).

RPA – Robotic Process Automation (Автоматизація роботизованих процесів).

FDW – Foreign Data Wrapper (Механізм взаємодії з зовнішніми джерелами даних у PostgreSQL).

VM – Virtual Machine (Віртуальна машина).

IVR – Interactive Voice Response (Інтерактивна голосова відповідь).

CRM – Customer Relationship Management (Управління відносинами з клієнтами).

HR – Human Resources (Кадровий відділ).

FWS – Facility Workplace Services (Послуги з управління робочим середовищем).

LUIS – Language Understanding Intelligent Service (Служба розуміння мови).

CIDR – Classless Inter-Domain Routing (Безкласова маршрутизація між доменами).

MAC – Media Access Control (Контроль доступу до середовища передачі).

DNS – Domain Name System (Система доменних імен).

CERN – European Organization for Nuclear Research (Європейська організація з ядерних досліджень).

NASA – National Aeronautics and Space Administration (Національне управління з аеронавтики і дослідження космосу).

ВСТУП

Актуальність теми. В сучасному бізнес-середовищі ефективність роботи підприємств значною мірою залежить від якості організації внутрішніх сервісів. Технічна підтримка, управління заявками, обробка запитів співробітників та інші сервіси є критично важливими для стабільного функціонування організацій. Без належної підтримки цих процесів функціонування підприємства стає не лише сумнівним, а й практично неможливим, особливо в умовах, коли всі основні бізнес-процеси покладаються на складні технології.

Для забезпечення ефективної роботи внутрішніх сервісів необхідні спеціальні інструменти. Головними серед них є засоби комунікації, прийняття заявок, їх обробки та відстеження прогресу. Такі інструменти не лише підвищують ефективність роботи технічних та адміністративних служб, але й забезпечують прозорість процесів для кінцевого користувача, надаючи йому змогу розуміти, на якому етапі знаходиться його запит і які дії виконуються. Створення власних автоматизованих рішень для підтримки цих процесів є актуальним завданням для сучасного бізнесу. Особливо в контексті економії ресурсів та можливості самим розвивати свої інструменти виходячи з персональних потреб, специфіки та завдань.

.

Теоретично-методичні аспекти. У контексті сучасного бізнесу питання організації внутрішніх сервісів часто розглядаються у блогах і статтях практиків, таких як Кріссі Кідд та Джо Гертвік, які акцентують увагу на ролі технічної підтримки в забезпеченні стабільності роботи компаній. Їхні матеріали зосереджуються на практичних аспектах впровадження трирівневої моделі підтримки та на перевагах автоматизації процесів.

Однак у сфері вивчення організації сервісів і технічної підтримки також є наукові роботи. Наприклад, Філіп Дж. Віндлі у своїй праці описує переваги багаторівневої підтримки, наголошуючи на важливості її адаптації

до масштабів і специфіки компанії. Нельсон К. Ю. Леунг та Сім Кім Лау досліджують класифікацію запитів, підкреслюючи, що стандартизація й структурованість процесів підтримки є ключовими факторами підвищення її ефективності.

Таким чином, питання організації внутрішніх сервісів і технічної підтримки знаходяться на перетині практичних підходів, що широко висвітлюються у блогах, і наукових досліджень, які закладають теоретичний фундамент для розробки ефективних моделей.

Метою дипломної роботи є розробка системи, якою став модульний чат-бот, для корпоративного середовища Microsoft Teams, який дозволить автоматизувати процеси обробки заявок, забезпечити інтеграцію з існуючими бізнес-системами та підвищити зручність користувачів у взаємодії з внутрішніми сервісами компанії. Робота також спрямована на створення економічно ефективної альтернативи готовим SaaS-рішенням, які часто не відповідають потребам малого та середнього бізнесу.

Апробація результатів дослідження. Результати дослідження були успішно впроваджені в компанії Innovescs, що дозволило зменшити витрати на підтримку сервісів. Результати дослідження апробовані на VII Всеукраїнська науково-технічна конференція «Комп'ютерні технології: інновації, проблеми, рішення» (02-03 грудня 2024 р.), Житомир, Державний Університет «Житомирська Політехніка».

РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА

1.1. Вступ до сервісних процесів та підтримки користувачів

1.1.1 Визначення внутрішнього сервісу та його ролі для організацій

Сучасний світ бізнесу та управління вимагає ефективних процесів для забезпечення безперебійної роботи як зовнішніх, так і внутрішніх операцій організацій. Для ефективної роботи організації, її головні активи повинні бути усім забезпечені. На мою думку головним активом кожної організації є люди які в ній працюють. Без них організації і не буде. Але у людей є проблеми та потреби, баз їх забезпечення і вирішення ефективність впаде. Для цього організації потребують сервіси.

Традиційно під "сервісом" розуміють послуги, що надаються клієнтам чи користувачам поза межами організації. Проте не менш важливу роль відіграє **внутрішній сервіс**, що спрямований на підтримку співробітників, оптимізацію робочих процесів та підвищення загальної ефективності компанії.

Внутрішні сервіси охоплюють широкий спектр задач, таких як технічна підтримка, обробка заявок, консультації та інші допоміжні функції, що забезпечують злагоджене функціонування внутрішньої інфраструктури. Правильно організовані внутрішні сервіси дозволяють компаніям швидко реагувати на виклики, зменшувати витрати часу та ресурсів, а також покращувати робочу атмосферу для співробітників.

Таким чином, **внутрішні сервіси** є не лише інструментом підтримки, але й стратегічним елементом управління, та розвитку. Але що таке сервіс в цілому?

Сервіс за МасКонелом – це дія або послуга, за яку споживач, компанія чи уряд готові заплатити.[1] Прикладами є робота, виконана перукарями, лікарями, юристами, механіками, банками, страховими компаніями тощо. Громадські послуги – це ті, за які платить суспільство (національна держава, фіскальний союз або регіон) у цілому. Використовуючи ресурси, навички, винахідливість і досвід, постачальник послуг приносить користь споживачам

послуг. Послуги можна визначити як нематеріальні дії або виконання, завдяки яким постачальник послуг створює цінність для клієнта.

Харісон визначає три ключові характеристики сервіса або послуги:[2]

Нематеріальність

Послуги за своєю природою є нематеріальними. Вони не виробляються, не транспортуються і не зберігаються на складі. Послуги неможливо зберегти для майбутнього використання. Вони створюються і споживаються одночасно.

Змінність

Кожна послуга є унікальною. Вона ніколи не може бути точно повторена, оскільки час, місце, обставини, умови, конфігурації або призначені ресурси будуть іншими для наступного надання, навіть якщо споживач замовляє ту саму послугу. Багато послуг вважаються гетерогенними і зазвичай адаптуються для кожного споживача або для кожного контексту послуги.

Приклад: Таксі, яке перевозить споживача послуги з дому на роботу, відрізняється від таксі, яке перевозить цього ж споживача з роботи додому – інший час, інший напрямок, можливо, інший маршрут, ймовірно, інший водій і автомобіль. Інший, більш поширений термін для цього – гетерогенність.

Скороминущість

Послуги мають дві риси скороминущості:

Ресурси, процеси та системи, необхідні для надання послуг, виділяються на певний період часу. Якщо споживач послуг не замовить і не споживе послугу протягом цього періоду, пов'язані ресурси можуть залишитися невикористаними. З точки зору постачальника послуг, це втрата можливості для бізнесу, якщо ресурси не можна використати інакше.

Приклад: Перукар обслуговує іншого клієнта. Порожнє місце в літаку не можна заповнити після вильоту.

Коли послуга повністю надана споживачеві, вона безповоротно зникає.

Приклад: пасажир доставлений до пункту призначення.

Постачальник послуги повинен надавати послугу в точний час її споживання. Послуга не має фізичного об'єкта, який був би незалежним від постачальника.

Споживач послуги також є невіддільним від процесу надання послуги.

Приклади: Споживач повинен сидіти в кріслі перукаря або в кріслі літака. Відповідно, перукар або пілот повинні бути у салоні чи літаку для надання послуги.

Усі вищезазначені характеристики та особливості сервісів повною мірою застосовуються й до **внутрішніх сервісів** організацій. Внутрішній сервіс є системою підтримки та обслуговування, що функціонує всередині компанії та спрямована на задоволення потреб її співробітників. Основна його роль полягає у забезпеченні ефективної роботи персоналу, оптимізації робочих процесів та підвищенні загальної продуктивності.

Що можуть включати у себе внутрішні сервіси?
Внутрішні сервіси охоплюють широкий спектр задач і можуть бути представлені у вигляді наступних категорій:

Таблиця 1.1.1.1

Приклади внутрішніх сервісів

<i>Категорія внутрішнього сервісу</i>	<i>Опис</i>
Технічна підтримка (IT Support)	Розв'язання проблем з обладнанням, ПЗ, підтримка корпоративних систем та обробка технічних заявок.
Фінансові послуги (Financial Services)	Консультації з фінансових питань, управління виплатами зарплат, премій, компенсацій та податків.
HR-сервіси (Human Resources)	Адміністрування страхування, навчання, розвиток персоналу, виплати бонусів, відпусток та кадрові питання.
Сервіси управління робочим середовищем (FWS)	Забезпечення комфортного робочого середовища, управління офісними просторами, обладнанням та інфраструктурою.

Інші внутрішні послуги	Юридична підтримка, корпоративна комунікація, психологічна підтримка, програми добробуту співробітників.
------------------------	--

Як наведено в Таблиці 1.1.1.1 сервісів може бути багато в середині організації і в залежності від розміру їх кількість і складність можуть змінюватися. Навіть, може змінюватися суть цих сервісів, але так чи інакше вони присутні в організаціях.

Роль внутрішніх сервісів у діяльності організації

Ефективно організовані внутрішні сервіси мають кілька ключових переваг для організацій:

- **Забезпечення ефективної роботи персоналу:** швидке вирішення проблем підвищує продуктивність співробітників і зменшує час можливого простою, а це в свою чергу зменшую збитки та збільшує надійність.
- **Оптимізація процесів:** автоматизація та стандартизація внутрішніх послуг дозволяє зменшити витрати часу та ресурсів. В свою чергу оптимізація процесів чи навіть їх перетворення, забезпечує розвиток та динамічну зміну. На мою думку це є чи не най головним фактором в усі часи. Спроможність на зміні, на структуровані та формалізовані зміни. Якщо подібній функції внутрішніх сервісів приділена увага в організації, то це підвищує час життя цієї організації у часи невизначеності, наприклад як у час війни.
- **Підвищення задоволеності співробітників:** якісний внутрішній сервіс створює комфортне середовище для роботи, що позитивно впливає на мотивацію.
- **Забезпечення прозорості та контролю:** внутрішні послуги полегшують управління та контроль над ресурсами компанії.

Прикладом роботи внутрішнього сервісу є технічна підтримка яка може називатися по різному Service Desk або Help Desk. Сервісна підтримка, зокрема технічна, є ключовим елементом забезпечення безперебійної роботи

інформаційних систем та задоволення потреб користувачів. Традиційні підходи до організації сервісної підтримки включають багаторівневу структуру, стандартизацію процесів та використання спеціалізованих інструментів для управління запитами. Що дозволяє оптимізувати роботу сервіса та розподілити запити та функції, в рамках відділу, даючи можливість не тільки займатися виключно підтримкою, але й розробкою, та навіть про активними діями наприклад по розробці нових процесів чи оптимізацією витрат.

Трирівнева модель технічної підтримки

Трирівнева модель підтримки є широко використовуваним методом організації технічної підтримки. Вона базується на розподілі запитів на три основні рівні, кожен із яких відповідає за вирішення певного класу проблем [3].

1. **Перший рівень (L1)** Основний контакт із користувачами. Запити, які надходять на цей рівень, зазвичай стосуються простих і часто повторюваних проблем, таких як скидання паролів, базова діагностика або відповіді на часті запитання. Співробітники L1 працюють за сценаріями та часто використовують базу знань для вирішення проблем [5]. "L1 підтримка забезпечує базове обслуговування клієнтів, швидко вирішуючи рутинні проблеми та передаючи складні випадки на вищі рівні" [5].

2. **Другий рівень (L2)** Запити, які не вдалося вирішити на першому рівні, передаються на L2. Цей рівень включає фахівців із глибокими знаннями про продукт або систему. Вони займаються діагностикою складніших проблем і часто вимагають доступу до більш детальної технічної інформації [3].

"Роль другого рівня підтримки полягає у глибокій технічній експертизі та вирішенні проблем, що потребують більше часу й ресурсів" [3].

3. **Третій рівень (L3)** Це найвищий рівень підтримки, який включає розробників продукту, інженерів або експертів із певної технології. L3 займається вирішенням складних проблем, які вимагають змін у коді,

інфраструктурі або процесах. Цей рівень також відповідає за створення нових оновлень або виправлення помилок. "L3 підтримка забезпечує остаточне вирішення складних проблем, що вимагають змін у системі" [5].

Класифікація запитів

Ефективна класифікація запитів є важливим елементом організації роботи технічної підтримки. Вона дозволяє швидше вирішувати проблеми, знижувати навантаження на працівників та покращувати якість обслуговування. Запити класифікуються за такими категоріями [4]:

1. **Рутинні запити:** Прості запити, які можна вирішити за допомогою бази знань або стандартних процедур. Наприклад, скидання паролів чи інформація про використання програмного забезпечення.
2. **Складні запити:** Проблеми, які потребують глибокої діагностики або аналізу, зазвичай вирішуються на рівні L2. Це можуть бути питання сумісності програмного забезпечення або налаштувань обладнання.
3. **Критичні інциденти:** Запити, які вимагають негайного вирішення через їхній вплив на бізнес. Такі випадки обробляються на рівні L3 або передаються спеціалізованим групам.

"Класифікація запитів дозволяє зосередити ресурси підтримки на найбільш критичних завданнях, оптимізуючи процеси" [4].

Таким чином може бути організовано ефективний сервіс в середині організації, чия роль та ефективність може бути вирішальною. Технічна підтримка є невід'ємною частиною внутрішнього сервісу організації, забезпечуючи ефективну роботу співробітників та стабільність бізнес-процесів. Роль внутрішніх сервісів стає ще більш важливою в умовах швидкого технологічного розвитку, жорсткої конкуренції та зростаючих вимог до якості внутрішніх послуг. Інвестуючи в розвиток внутрішніх сервісів, організації підвищують свою конкурентоспроможність, забезпечують задоволеність співробітників та створюють основу для сталого розвитку навіть в умовах невизначеності. Загалом, внутрішні сервіси наприклад сервіс навчання та розвитку, є важливими елементами,

що впливають на успіх та ефективність організації основою якої були і є люди які на неї працюють. Їх правильна організація та постійне вдосконалення сприяють розвитку та стійкості організацій.

1.1.2 Актуальність технічної підтримки для сучасних компаній як приклад необхідності внутрішніх сервісів

Незалежно від розміру, всі компанії потребують якісної та оперативної ІТ-підтримки, щоб працювати ефективно та мінімізувати час простою. Останніми роками бізнеси значно збільшили інвестиції в технології, що призвело до зростання складності та необхідності у практичній ІТ-підтримці.

Компанії, які мають власний ІТ-відділ, часто витрачають більшість часу на обробку запитів щодо підтримки, що може затримувати реалізацію масштабних ІТ-проектів. Такі затримки можуть призвести до втрати доходів та погіршення досвіду співробітників чи клієнтів. Для компаній, які не мають власного ІТ-відділу, відсутність якісної ІТ-підтримки може бути катастрофічною, адже технічні проблеми призводять до значного простою та втрати клієнтів.

Основні функції технічної підтримки

Технічна підтримка є основою сучасного бізнесу, забезпечуючи його ефективну роботу. За словами генерального директора іT2, якісна ІТ-підтримка охоплює кілька критично важливих функцій[6]:

Вирішення технічних проблем

Найпомітнішою частиною роботи технічної підтримки є усунення технічних неполадок. Коли комп'ютер перестає працювати, принтер відмовляє або програма зависає, команда підтримки знаходить причину й усуває її. Це потребує технічних знань, навичок вирішення проблем і терпіння.

Приклад: якщо ноутбук співробітника постійно відключається від Wi-Fi, ІТ-підтримка може перевірити налаштування мережі, оновити драйвери або замінити несправне обладнання.

Обслуговування систем та оновлення

Як і автомобілі, ІТ-системи потребують регулярного обслуговування для ефективної роботи[6]. ІТ-фахівці займаються встановленням оновлень, моніторингом стану системи та усуненням потенційних проблем до їх виникнення.

Управління мережею та безпекою

У сучасному взаємопов'язаному світі надійна та безпечна мережа є критично важливою для будь-якого бізнесу. ІТ-команди налаштовують мережеву інфраструктуру, впроваджують системи захисту та навчають співробітників розпізнавати загрози, як-от фішинг.

Резервне копіювання та відновлення даних

Втрату даних можна порівняти з катастрофою для бізнесу. ІТ-підтримка налаштовує автоматичне резервне копіювання в хмару або на зовнішні носії та розробляє плани відновлення у випадку втрати даних.

Допомога користувачам та навчання

Окрім усунення технічних проблем, ІТ-підтримка допомагає користувачам упевнено працювати з технологіями. Це включає створення інструкцій, проведення тренінгів або консультації за запитом.

Стратегічна важливість ІТ-підтримки

Підвищення ефективності операцій

Завдяки мінімізації простоїв ІТ-підтримка забезпечує безперебійну роботу співробітників, що максимізує продуктивність.

Забезпечення безперервності бізнесу

Під час непередбачених подій, таких як кібератаки чи природні катастрофи, ІТ-підтримка відіграє важливу роль у збереженні даних і забезпеченні безперервності роботи бізнесу.

Підтримка дистанційної роботи

ІТ-професіонали налаштовують віддалений доступ, вирішують проблеми з підключенням і забезпечують ефективну співпрацю співробітників з різних місць.

Сприяння цифровій трансформації

ІТ-підтримка допомагає бізнесу інтегрувати нові технології, такі як хмарні рішення, штучний інтелект або Інтернет речей, у вже існуючу інфраструктуру[6].

Економічний вплив ефективної ІТ-підтримки

Економія через профілактичне обслуговування

Своєчасне обслуговування обладнання та програмного забезпечення запобігає дорогим поломкам і подовжує термін служби активів.

Зменшення простоїв і пов'язаних з ними витрат

Кожна хвилина простою означає втрату продуктивності та доходу. ІТ-підтримка мінімізує ці простої, швидко вирішуючи проблеми.

Підвищення продуктивності співробітників

Коли технології працюють безперебійно, співробітники можуть зосередитися на своїх основних завданнях без затримок.

Забезпечення масштабованості та зростання

ІТ-підтримка допомагає бізнесу адаптуватися до змін і розширюватися, впроваджуючи нові рішення для підтримки бізнес-процесів.

Технічна підтримка – це не просто функція «за лаштунками»; це рушійна сила успіху бізнесу в цифрову епоху. Від вирішення технічних проблем до захисту даних і впровадження інновацій, ІТ-фахівці забезпечують безперебійну та ефективну роботу бізнесу.

Із появою нових технологій, таких як штучний інтелект та машинне навчання, ІТ-підтримка продовжить еволюціонувати. Однак одне залишається незмінним: потреба у кваліфікованих ІТ-професіоналах, які здатні об'єднати людей і технології, буде тільки зростати.

З наведеної інформації можна зробити висновок, що внутрішні сервіси компанії відіграють критично важливу роль не лише у підтримці працівників, але й у забезпеченні загальної ефективності роботи організації. Технічна підтримка як ключовий приклад внутрішнього сервісу демонструє, наскільки важливим є швидке вирішення проблем, обслуговування систем,

забезпечення безпеки даних та впровадження нових технологій для стабільного функціонування компанії.

Важливо підкреслити, що **кожен внутрішній сервіс** складається з двох основних компонентів: команди, яка надає послуги, та клієнтів, які їх отримують. Для досягнення максимальної ефективності внутрішніх сервісів необхідно впроваджувати сучасні інструменти взаємодії з клієнтами та оптимізувати процеси надання послуг.

1.2. Проблеми традиційних систем підтримки

1.2.1 Короткий огляд існуючих рішень (Jira, Freshdesk, Zendesk) та їх функціональні обмеження у порівнянні з потребами малого та середнього бізнесу

Існує багато готових рішень типу SaaS(Software as a service). Вони за часту розвинені, мають історію розробки та активно підтримуються та розвиваються. Наприклад для забезпечення роботи внутрішніх сервісів можна використати **Jira Service Management**. Непогане рішення, але яке коштує грошей, та має свої обмеження, якщо говорити в контексті малих та середніх бізнесів. Малий та середній бізнес (МСБ) потребує ефективних та доступних інструментів для управління внутрішніми сервісами. Популярні рішення, такі як **Jira Service Management, Freshdesk, Zendesk** та **Workativ**, пропонують широкий спектр можливостей. Проте, їх функціональність та вартість не завжди відповідають потребам і можливостям МСБ. Особливо, коли інфраструктура бізнесу складається з різних систем і їх поєднання або неможливе при використанні готових сервісів, або потребує додаткових щомісячних витрат на розширення та інтеграції. У цьому огляді детально розглянемо кожен з цих продуктів, їх позитивні та негативні сторони, а також їх відповідність вимогам МСБ.

1. Jira Service Management

Огляд продукту

Jira Service Management[7] від компанії Atlassian[7] — це потужна платформа, головним чином призначена для управління IT-послугами (ITSM). Вона пропонує високий рівень налаштування, дозволяючи бізнесам адаптувати робочі процеси, автоматизувати операції та інтегруватися з іншими інструментами екосистеми Atlassian, такими як Confluence та Bitbucket. Jira широко використовується для управління проектами розробки програмного забезпечення та відстеження проблем, надаючи розширені можливості для сервісних служб, управління інцидентами та вирішення проблем.

Позитивні сторони:

- **Висока гнучкість та налаштовуваність:** Можливість адаптувати систему під специфічні потреби бізнесу.
- **Інтеграція з екосистемою Atlassian[7]:** Тісна інтеграція з іншими продуктами, що покращує колаборацію та ефективність команд.
- **Потужні функції для IT-команд[7]:** Підтримка Agile та DevOps практик, що корисно для розробників та IT-фахівців.

Негативні сторони:

- **Складність використання:** Стрімка крива навчання може бути викликом для компаній без спеціалізованого IT-персоналу.
- **Висока вартість повного функціоналу:** Розширені можливості доступні лише в дорогих планах, що може бути недоступним для МСБ з обмеженими бюджетами.
- **Надмірність для малих команд:** Деякі функції можуть бути зайвими для невеликих організацій, що ускладнює використання.

2. Freshdesk

Огляд продукту

Freshdesk[8] — це програмне забезпечення для підтримки клієнтів, розроблене для оптимізації процесів служби підтримки та забезпечення ефективного обслуговування. Воно включає обробку заявок, багатоканальну

підтримку (електронна пошта, чат, телефон, соціальні мережі), інструменти автоматизації та базу знань. Відомий своїм інтуїтивним інтерфейсом та швидким впровадженням, Freshdesk популярний серед компаній, які шукають доступне та просте у використанні рішення.

Позитивні сторони:

- **Легкість використання:** Інтуїтивний інтерфейс, який не вимагає глибоких технічних знань.
- **Доступне ціноутворення:** Гнучкі плани, включаючи безкоштовний рівень для малих команд.
- **Багатоканальна підтримка[8]:** Підтримує різні канали комунікації з клієнтами.

Негативні сторони:

- **Обмежені можливості налаштування:** Може бути недостатньо гнучким для бізнесів зі складними або специфічними потребами.
- **Додаткові витрати на розширені функції:** Ключові можливості, такі як розширена аналітика та автоматизація, доступні лише в дорожчих планах.

3. Zendesk

Огляд продукту

Zendesk[9] — це широко використовувана платформа для підтримки клієнтів, яка пропонує систему обробки заявок, базу знань та портали самообслуговування. Вона підтримує омніканальну комунікацію та відома своїми потужними інтеграціями з сторонніми інструментами, такими як Slack, Salesforce та HubSpot. Zendesk також надає аналітичні можливості та функції автоматизації на основі штучного інтелекту, що робить його популярним серед великих підприємств.

Позитивні сторони:

- **Широкий набір функцій[9]:** Відповідає потребам компаній різного масштабу.
- **Потужні інтеграції[9]:** Легко інтегрується з багатьма популярними інструментами та платформами.

- **Аналітика та автоматизація[9]:** Допомагає покращити ефективність та якість підтримки.

Негативні сторони:

- **Висока вартість за повний функціонал:** Розширені можливості доступні лише на преміум-планах, що може бути недоступним для МСБ.
- **Складність налаштування:** Може вимагати значних зусиль для конфігурації та оптимізації під специфічні потреби.
- **Потреба в спеціалізованому персоналі:** Для максимального використання можливостей може знадобитися наявність ІТ-фахівців.

4. Workativ

Огляд продукту

Workativ[10] — це платформа, яка пропонує інтелектуальних чат-ботів та автоматизацію для підтримки співробітників. Вона спрямована на покращення внутрішньої підтримки через самообслуговування та автоматизацію рутинних завдань. Workativ інтегрується з корпоративними месенджерами, такими як Slack та Microsoft Teams, забезпечуючи зручність для співробітників.

Позитивні сторони:

- **Автоматизація внутрішніх процесів[10]:** Знижує навантаження на службу підтримки, автоматизуючи рутинні завдання.
- **Використання штучного інтелекту[10]:** Покращує взаємодію та швидкість реагування на запити.
- **Інтеграція з месенджерами:** Підвищує зручність та доступність підтримки для співробітників.

Негативні сторони:

- **Вузька спеціалізація:** Фокус на внутрішній підтримці може бути недостатнім для зовнішньої підтримки клієнтів.
- **Обмежений функціонал:** Менше можливостей порівняно з повноцінними сервісними платформами.

- **Витрати на впровадження:** Може вимагати додаткових ресурсів для інтеграції та налаштування.
Загальні обмеження для МСБ
- **Ефективність з точки зору витрат:** Хоча багато платформ пропонують безкоштовні або базові плани, для доступу до ключових функцій часто потрібно оновитися до дорожчих планів, що може бути фінансово обтяжливим для МСБ.
- **Складність та надмірність функцій:** Багато рішень розроблені з урахуванням потреб великих підприємств, що призводить до надлишку функцій та складності для малих бізнесів.
- **Потреба в спеціалізованому персоналі:** Ефективне використання та налаштування цих платформ може вимагати наявності ІТ-фахівців, яких може не бути в МСБ.
- **Відповідність специфічним потребам:** МСБ часто потребують простих та інтуїтивних інструментів, які можна швидко впровадити та адаптувати до їх унікальних бізнес-моделей.

Висновок

Таблиця 1.2.1.1

Приблизна розцінка сервісних планів

Сервіс	Плани та вартість
Jira Service Management	- Безкоштовний план (до 3 агентів); - Standard: \$20 за агента/місяць; - Premium: \$40 за агента/місяць; - Enterprise: індивідуальне ціноутворення.
Freshdesk	- Free: безкоштовно для необмеженої кількості агентів; - Growth: \$15 за агента/місяць; - Pro: \$49 за агента/місяць; - Enterprise: \$79 за агента/місяць.

Zendesk	- Support Team: \$19 за агента/місяць; - Support Professional: \$55 за агента/місяць; - Support Enterprise: \$115 за агента/місяць; - Suite Team: \$55 за агента/місяць; - Suite Growth: \$89 за агента/місяць; - Suite Professional: \$115 за агента/місяць.
Workativ	- Starter: \$99 на місяць (до 50 співробітників); - Professional: \$199 на місяць (до 100 співробітників); - Enterprise: індивідуальне ціноутворення.

На основі проведеного аналізу можна зробити висновок, що сучасні готові рішення для технічної підтримки, такі як **Jira Service Management**, **Freshdesk**, **Zendesk** та **Workativ**, є високоефективними та функціональними інструментами, які успішно вирішують завдання середніх і великих компаній. Потужний функціонал для управління заявками, автоматизації процесів і аналітики.

Однак, незважаючи на всі позитивні сторони, ці рішення мають ряд функціональних обмежень у контексті малого та середнього бізнесу (МСБ): це висока вартість як наведено в таблиці 1.2.1 ціни утворюються за користувача, складність налаштування та використання, надмірність функцій, проблеми інтеграції.

Багато МСБ використовують різноманітні системи, які не завжди можна легко поєднати за допомогою готових рішень. Інтеграція часто потребує додаткових витрат часу та ресурсів іноді виглядає як «костиль» і працює відповідно. Тому можливим рішенням, може стати інвестиція у розробку своєї невеликої системи комунікації, яку буде створено на базі існуючих систем силами наявної команди підтримки, наприклад команди третього рівня підтримки.

1.2.2 Виклики готових систем підтримки

В попередньому розділі були проаналізовані сервісні системи. Але необхідно детальніше дослідити, основні проблеми та виклики пов'язані з подібними системами та їх використанням в бізнесі. Системи підтримки стикаються з кількома викликами, що можуть заважати їхній ефективності та сталості в сучасному малому та середньому бізнесі. Дві найактуальніші проблеми — це високі витрати на ліцензування та обмежена функціональність, що часто не дозволяє організаціям максимально використовувати свій операційний потенціал і приймати більш гнучкі, масштабовані рішення.

1. Високі витрати на ліцензування

Однією з найважливіших проблем готових систем підтримки чи SaaS є їхня залежність від дорогих ліцензійних моделей. Ці системи зазвичай працюють на основі ліцензування "за користувача" або "за пристрій", що може стати надзвичайно дорогим у міру зростання організації. Для багатьох компаній, особливо малого та середнього бізнесу (МСБ), фінансовий тягар від таких регулярних витрат може напружувати бюджети та заважати масштабуванню. Згідно з дослідженнями, витрати на підтримку традиційних систем включають не тільки початкову покупку, але й постійні витрати на оновлення, підтримку та технічне обслуговування програмного забезпечення[14].

Крім того, багато традиційних систем пропонують обмежену гнучкість у своїх ліцензійних моделях, змушуючи компанії купувати ліцензії навіть на ті функції чи послуги, які вони можуть не повністю використовувати. Це призводить до значних втрат ресурсів, оскільки компанії змушені переплачувати за функціональність, якою вони не користуються[15]. Дослідження показують, що високі витрати на ліцензування програмного забезпечення є однією з основних причин, чому бізнес розглядає можливість переходу на більш сучасні, хмарні рішення[14].

2. Обмежена функціональність та налаштованість(Customization)

Ще однією ключовою проблемою традиційних систем підтримки є їхня обмежена функціональність, особливо у порівнянні з сучасними хмарними альтернативами. Традиційні системи часто розробляються для конкретних

випадків використання та не мають гнучкості для адаптації до динамічних потреб сучасних компаній. У міру того, як компанії зростають і їхні операційні вимоги змінюються, вони часто виявляють, що їхні застарілі системи не можуть задовольнити нові потреби без дорогих оновлень або налаштувань[11].

Одним із найпоширеніших обмежень традиційних систем є нездатність безперешкодно інтегруватися з іншими бізнес-інструментами та платформами. Сучасні компанії зазвичай використовують широкий набір програмних рішень для управління взаємовідносинами з клієнтами (CRM), планування ресурсів підприємства (ERP) і управління проектами. Однак традиційні системи підтримки часто не можуть інтегруватися з цими платформами, що призводить до інформаційних бар'єрів та неефективності у робочих процесах[12]. Така відсутність інтеграції створює значні перешкоди для підвищення операційної ефективності та швидкості реагування.

Крім того, традиційні системи часто не підтримують багатоканальну комунікацію, яка є критично важливою для сучасного обслуговування клієнтів. Клієнти очікують можливості взаємодії з компаніями через електронну пошту, чат, соціальні мережі та інші цифрові платформи. Однак багато застарілих рішень здатні працювати лише з обмеженою кількістю каналів, що ускладнює запровадження омніканальних стратегій обслуговування клієнтів[11].

З іншого боку, хмарні рішення програмного забезпечення як послуги (SaaS) пропонують більш сучасний підхід, але і вони мають свої особливості у питанні налаштування. Хмарне SaaS-рішення зазвичай не можна глибоко кастомізувати під унікальні потреби бізнесу, оскільки постачальники обмежують модифікації, прагнучи зберегти стандартизований, простий у використанні продукт. Зазвичай хмарне SaaS-рішення є готовим (Out-of-the-box, OOTB)[13] зі встановленим набором функцій і можливостей.

Водночас локально розгорнуті (on-premises) SaaS-рішення[13] дозволяють ширші можливості кастомізації порівняно з їхніми хмарними

аналогами. Хоча персоналізація та незначні конфігурації все ж можливі у хмарних рішеннях, глибші модифікації можуть потребувати додаткових витрат як на саме вдосконалення, так і на подальше обслуговування цих змін. Таким чином, перед впровадженням будь-якого OOTB SaaS-рішення[13] компанії повинні ретельно оцінити, наскільки таке рішення відповідає їхнім поточним вимогам та потенційному майбутньому зростанню.

3. Повільні інновації та оновлення

Третьою проблемою, що ускладнює високі витрати та обмежену функціональність, є повільний темп інновацій у традиційних системах підтримки. На відміну від сучасних хмарних платформ, які можна регулярно та автоматично оновлювати, традиційні системи часто потребують ручного оновлення або навіть загальних оновлень системи для впровадження нових функцій або виправлення безпеки. Це може призвести до періодів простою, порушуючи бізнес-процеси та збільшуючи загальну вартість володіння[14].

Крім того, повільний темп інновацій у традиційних системах означає, що компанії часто відстають у впровадженні нових технологій, які могли б покращити їхню підтримку. Наприклад, досягнення в галузі штучного інтелекту (AI) та машинного навчання значно змінили спосіб надання послуг підтримки, відкриваючи можливості для таких функцій, як прогностична аналітика та чат-боти на базі AI. Однак багато традиційних систем не обладнані для інтеграції цих технологій, змушуючи компанії або значно інвестувати в кастомізовані рішення, або зовсім відмовлятися від переваг[15].

Так одним з простих і відносно не дорогих рішень може бути розробка власного чат-бота.

РОЗДІЛ 2. КЕЙС: РОЗРОБКА МОУДЛЬНОГО СЕРВІСНОГО ЧАТ-БОТА ДЛЯ MICROSOFT TEAMS В КОМПАНІЇ INNOVECS

2.1. Початкові вимоги та постановка завдання

2.1.1 Jira Service Managment як приклад дорогого рішення для внутрішнього сервісу.

В даному розділі будуть продемонстровані проблеми та причини змін в компанії де я проходив практику. Буде відображено конкретні та унікальні проблеми які стали поштовхом для розробки бота, який детально буде продемонстрований в наступних розділах. Jira Service Management Help Desk від Atlassian є популярним інструментом для організації технічної підтримки в багатьох компаніях. Незважаючи на його потужні можливості та інтеграцію з іншими продуктами Jira, для деяких організацій може стати джерелом проблем через високі витрати на ліцензування.

Основою для роботи внутрішніх сервісів, в компанії був сервіс **Jira Service Managment**. Цей сервіс використовувався, для двох завдань. Щоб створювати запити, а сервісні команди могли відстежувати свої завдання і виконувати їх. Jira Service Managment це майже те ж саме що і Jira з невеликою різницею, створювати запити(тікети) можливо ззовні, без необхідності мати ліцензію Jira. Таким чином функціонал придбаних ліцензій дублювався що дуже сильно збільшувало витрати на внутрішні сервіси. Тобто компанії необхідно сплачувати заробітну платню кожному працівнику сервісу, покупати йому дві дублюючі ліцензії і це за винятком усіх інших витрат.

Причини змін та проблеми використання Jira Service Management Help Desk

1. Високі витрати на ліцензування

Однією з головних проблем використання Jira Service Management Help Desk є значні витрати на ліцензії для виконавців:

- **Індивідуальні ліцензії для кожного виконавця:** Кожен співробітник, який бере участь у обробці заявок, потребує окремої ліцензії. Це суттєво збільшує

бюджет організації, особливо якщо врахувати, що багато з них вже мають ліцензії на інші продукти Jira для своїх робочих потреб.

Таблиця 2.1.1.1

Продукти та додатки	Тип ліцензування	Наступна оцінка вартості	Платіжний цикл	Розробник
<i>Jira Service Management</i>	За користувача	USD 47	Monthly	Atlassian
<i>Jira</i>	За користувача	USD 15	Monthly	Atlassian
<i>Jira Misc Workflow Extension</i>	За користувача	USD 1.5	Monthly	Atlassian

- **Необхідність ліцензій для різних відділів:** Технічна підтримка часто вимагає залучення співробітників з різних відділів — HR, IT, фінансів тощо. Забезпечити всіх необхідними ліцензіями стає фінансово обтяжливим. Кожен з робітників перерахованих відділів вже має ліцензії, які необхідні для виконання своїх обов’язків. Більшість працівників, цих відділів, особливо IT відділу, виконують унікальні ролі, і володіють унікальним набором здібностей, розрахованих на специфічні завдання.

Таблиця 2.1.1.2

Приклад складу команди за посадами

Посада	Роль у команді
Інженер даних	Відповідає за обробку, аналіз та управління великими обсягами даних.
Продукт-менеджер	Координує розробку продукту, визначає пріоритети та забезпечує виконання бізнес-цілей.
Розробник Java	Розробляє програмне забезпечення на Java, забезпечуючи функціональність систем.

Сервіс-спеціаліст	Конфігурує системи, зокрема HR-системи, для оптимізації внутрішніх процесів.
Менеджер з розробки ПЗ	Керує командами розробників, визначає технічну стратегію та пріоритети.
Інженер-програміст (Back-End фінансові системи)	Розробляє серверну частину фінансових систем, забезпечуючи їх стабільність та інтеграцію.

В таблиці 2.1.2 відображено гіпотетичний склад однієї команди. Таким чином щоб ефективно виконувати запити клієнтів кожен з спеціалістів повинен отримати ліцензію. А це приклад тільки однієї з внутрішніх команд.

- **Обмеженість бюджету:** Особливо для малих та середніх бізнесів, витрати на ліцензування можуть перевищувати доступні ресурси. Таким чином витрати тільки на ліцензії Jira для продемонстрованого складу команди складатимуть USD 381 на місяць.

Таблиця 2.1.1.3

Витрати команди

Команда	Витрати в місяць	Витрати в рік
Команда розробки Таблиця 2.1.2	USD 381	USD 4,572

2. Обмежений доступ до заявок для виконавців без ліцензій

Якщо виконавець не має ліцензії Jira Service Management:

- **Виконавець не може бачити та обробляти заявки:** Відсутність доступу до системи унеможливорює ефективну роботу.
- **Необхідність конвертації заявок:** Заявки з Help Desk потрібно перетворювати на стандартні задачі Jira. Після цього клієнт втрачає доступ до своєї заявки та не може відстежувати її статус.

- **Втрата прозорості для клієнта:** Клієнт змушений звертатися до підтримки якщо його заяву було конвертовано, щоб дізнатися про прогрес, що створює додаткове навантаження на команду підтримки. При конвертації, заява клієнта зникає з сервісу Jira Service Management.

3. Проблеми з інформуванням клієнтів

- **Недостатня ефективність електронної пошти:** Оновлення статусу заявки надсилаються на електронну пошту клієнта, але багато користувачів можуть не відкривати ці листи через великий потік повідомлень або просто ігнорувати їх.

- **Необхідність особистого спілкування:** Виконавці змушені писати клієнтам безпосередньо (наприклад, у месенджерах), щоб інформувати про прогрес, що відволікає від основної роботи.

- **Додаткові часові витрати:** Постійне інформування клієнтів забирає час, який міг би бути витрачений на вирішення інших задач.

4. Обмежений доступ до порталу підтримки

- **Розміщення посилання на підтримку в HR-сервісі:** Оскільки не всі співробітники мають доступ до HR-сервісу або регулярно ним користуються, багато хто просто не знає, де можна подати заявку на підтримку.

Сукупність цих факторів, створила необхідність знайти альтернативу, для підтримання внутрішніх сервісів, з можливістю контролю над продуктом, та зменшення витрат.

2.1.2 Потреба у створенні доступнішої та дешевшої, модульної системи підтримки користувачів

Альтернативною системою став чат-бот. Він є потужним інструментом для автоматизації взаємодії з користувачами, надаючи численні переваги, такі як цілодобова доступність, зниження витрат та масштабованість. Чато-боти в корпоративних системах можуть відігравати важливу роль в оптимізації робочих процесів, покращенні взаємодії з користувачами та зниженні операційних витрат

за рахунок автоматизації рутинних завдань. Але основні причин вибору чат-боту як основи нової сервісної систему були наступні:

1. **Дешева і легка розробка:** Чат-бот створюється на основі вже готових інструментів, та рішень. За звичай бот розповсюджується в месенджерах, що дозволяє не витрачати час та ресурси на дистрибуцію, створення інтерфейсу і довге навчання, бо документація в відкритому доступі, а прикладів реалізації також достатньо.
2. **Легка інтеграція з різними системами:** Чат-бот може бути легко інтегрований з існуючими системами та платформами компанії, такими як CRM, ERP, бази даних тощо.
3. **Повний контроль та доступність:** Розробка власного чат-бота забезпечує повний контроль над його функціональністю та безпекою даних.
4. **Використання існуючої інфраструктури:** Чат-бот використовує вже наявну інфраструктуру компанії (сервери, бази даних, мережі), що підвищує її ефективність та коефіцієнт корисної дії (ККД), оскільки раніше недовантажені ресурси тепер використовуються. Також знижує витрати, так як додаткових коштів для розробки не потребує, легко імплементуючись в наявну інфраструктуру.
5. **Можливість інтеграції інших сервісів:** Окрім функцій підтримки, в чат-бот можна інтегрувати додаткові сервіси, такі як автоматизація процесів, інформування співробітників, бронювання ресурсів тощо.

2.1.3 Визначення чат-ботів, їх типи і використання в корпоративних системах

Чат-бот[14] — це розмовний інструмент, призначений для розуміння запитів клієнтів і автоматичної відповіді, імітуючи письмові або усні розмови людини. Як ви побачите нижче, деякі чат-боти є елементарними, пропонуючи користувачам прості меню з варіантами для вибору. Однак більш просунуті чат-боти можуть використовувати штучний інтелект (ШІ) і обробку

природної мови (NLP), щоб краще розуміти введення користувачем і легко вести складні розмови.[14]

Які існують типи чат-ботів?

1. Чат-боти на основі меню або кнопок

Чат-боти на основі меню або кнопок є найпростішим типом чат-ботів, де користувачі можуть взаємодіяти, натискаючи на варіанти меню, що найкраще відображають їхні потреби. Залежно від того, що вибирає користувач, бот може запропонувати інший набір варіантів для вибору, поки не буде досягнуто найкращого варіанта.[14] Такі чат-боти працюють за принципом дерева рішень.

Хоча ці чат-боти пропонують просту функціональність і можуть бути корисними для відповідей на повторювані запитання користувачів, вони стикаються з труднощами при більш складних запитах, оскільки обмежені заздалегідь визначеними відповідями. По-перше, вони можуть потребувати більше часу для розуміння потреб клієнта, якщо користувачеві доводиться проходити кілька ітерацій кнопок меню. По-друге, якщо запит користувача не входить до варіантів меню, чат-бот стає непотрібним, оскільки не пропонує поля для вільного введення тексту. Якраз цей тип боту був обраний для реалізації. Сервісний бот, необхідний для невеликого набору завдань, які звісно можуть розширитися з часом, але усі вони є формалізованими, чітко встановленими процесами.

2. Чат-боти на основі правил

Розвиваючи просту функціональність дерева рішень, чат-боти на основі правил використовують умовну логіку «якщо/тоді» для побудови автоматизованих розмов. Такі боти діють як інтерактивні FAQ, де конструктор розмов програмує заздалегідь визначені комбінації запитань і відповідей, щоб бот міг розуміти введення користувачем і реагувати точно.

Чат-боти цього типу працюють на основі виявлення ключових слів, їх легко навчити, і вони добре працюють із передбачуваними запитаннями. Однак вони стикаються з труднощами при складних запитах, які не передбачені конструктором розмов, оскільки їхні відповіді залежать від заздалегідь підготовленого контенту. Через обмеження цих ботів вони можуть не зрозуміти запит користувача,

пропустити важливі деталі або попросити повторити інформацію. Це може спричинити негативний досвід користувача, а іноді — навіть неможливість перенаправлення до живого агента, що ще більше погіршує ситуацію.

3. Чат-боти на основі ШІ

На відміну від чат-ботів на основі правил, які працюють лише з передбачуваними запитаннями та відповідями, чат-боти на основі ШІ можуть розуміти запитання незалежно від того, як вони сформульовані. Завдяки можливостям штучного інтелекту та розуміння природної мови (NLU) такі боти можуть швидко визначати всю релевантну інформацію, що дозволяє розмові проходити більш природно. Якщо ШІ-бот не розуміє запит, він може уточнити або запропонувати кілька можливих дій для вибору користувачем.

Алгоритми машинного навчання, які використовуються в ШІ-ботах, дозволяють їм навчатися на основі взаємодії з користувачами та вдосконалювати базу знань. З часом такі боти краще розуміють наміри користувача і можуть надавати більш точні відповіді. Крім того, ці боти можуть запам'ятовувати розмови і використовувати цей контекст у майбутніх взаємодіях. Завдяки інтеграції з такими інструментами, як RPA, користувачі можуть виконувати завдання через чат-бот, наприклад, замовляти піцу за допомогою збережених даних про попередні замовлення.

4. Голосові чат-боти

Голосовий чат-бот — це інструмент, який дозволяє користувачам взаємодіяти з ботом через голосові команди, а не через текст. Деякі голосові чат-боти можуть бути базовими, що викликає розчарування у користувачів через обмежені можливості таких систем, як IVR. Однак, завдяки штучному інтелекту, голосові чат-боти можуть мати ті ж просунуті функції, що й текстові ШІ-боти, використовуючи технології перетворення голосу в текст і навпаки. Вони можуть аналізувати запити користувачів і надавати відповіді у розмовному стилі.

Голосові чат-боти забезпечують швидший спосіб комунікації, оскільки користувач може отримати відповідь у режимі реального часу без необхідності вводити текст або обирати варіанти в меню.

5. Чат-боти на основі генеративного ШІ

Наступне покоління чат-ботів із генеративними можливостями ШІ пропонує ще більший функціонал, розуміючи природну мову та адаптуючись до стилю спілкування користувача. Вони можуть не тільки відповідати на запитання, але й створювати новий контент, такий як тексти, зображення або звуки на основі моделей великих мов. Генеративні ШІ-боти можуть розпізнавати, підсумовувати, перекладати та створювати контент у відповідь на запит користувача без необхідності втручання людини.

Використання в корпоративних системах: У корпоративних середовищах чат-боти виконують кілька важливих функцій:

- **Підтримка клієнтів:** Вони надають миттєві відповіді на запити клієнтів, автоматизують рутинні завдання та передають складніші проблеми людським агентам.
- **Автоматизація внутрішніх процесів:** Чат-боти допомагають співробітникам із повторюваними завданнями, такими як пошук даних, усунення неполадок ІТ-систем і керування внутрішніми системами, звільняючи людські ресурси для більш стратегічної роботи.
- **Продажі та маркетинг:** У маркетингу чат-боти залучають потенційних клієнтів, відповідаючи на запитання щодо продуктів, супроводжують користувачів під час процесу покупки та надають персоналізовані рекомендації на основі поведінки користувачів.

Створена розробка виконє виключно функцію автоматизації процесів.

Переваги використання чат-бота

- **Зниження витрат:** Відсутність необхідності в придбанні дорогих ліцензій для кожного користувача чи виконавця дозволяє суттєво знизити витрати.
- **Підвищення ефективності:** Автоматизація рутинних задач та швидке реагування на запити користувачів покращують продуктивність команди.

- **Покращення комунікації:** Чат-бот може надавати актуальну інформацію та оновлення безпосередньо користувачам, зменшуючи навантаження на службу підтримки.
- **Масштабованість та модульність:** Можливість додавати нові функції та сервіси в чат-бот без значних витрат часу та ресурсів.

Використання існуючої інфраструктури

Впровадження чат-бота дозволяє ефективніше використовувати вже наявні ресурси компанії:

- **Серверні потужності:** Використання існуючих серверів знижує додаткові витрати на обладнання.
- **Мережеві ресурси:** Оптимальне використання мережевої інфраструктури підвищує її ККД.
- **Програмні рішення:** Інтеграція з поточними програмними продуктами дозволяє уникнути дублювання функціональності та оптимізувати робочі процеси.

Розширення можливостей через інтеграцію

Чат-бот може стати центральним елементом для різних сервісів:

- **Автоматизація:** Впровадження автоматизованих відповідей та процесів знижує людський фактор та пришвидшує виконання задач.
- **Інформування:** Швидке розповсюдження важливих повідомлень, новин чи оновлень для співробітників або клієнтів.
- **Аналітика:** Збір даних про взаємодію користувачів з ботом для подальшого аналізу та покращення сервісів.

Недоліки чат-ботів

- **Обмежена функціональність:** Основним недоліком чат-ботів є їхня обмежена здатність обробляти складні або нестандартні запити. У більшості випадків чат-боти добре справляються з передбачуваними і простими завданнями, але при виникненні складніших питань вони можуть бути неефективними і змушені перенаправляти запит до живого оператора.[15]

- **Відсутність емоційного інтелекту:** Чат-боти не можуть розпізнавати та адекватно реагувати на емоційний стан користувача. Наприклад, якщо клієнт розчарований або засмучений, бот не зможе запропонувати співчутливе або персоналізоване вирішення проблеми, що може призвести до незадоволення користувача і негативного досвіду взаємодії з компанією.
- **Необхідність у налаштуванні та обслуговуванні:** Налаштування чат-ботів може вимагати значного часу та ресурсів, особливо якщо компанія використовує складні рішення, інтегровані з іншими системами. Після впровадження чат-боти потребують регулярних оновлень і оптимізації, щоб залишатися актуальними та ефективними.
- **Залежність від технологій:** Чат-боти повністю залежать від стабільного доступу до інтернету та інших технологій. Проблеми з підключенням можуть зробити їх недоступними або непрацездатними. Крім того, чат-боти можуть бути вразливими до кібератак або витоків даних, що потребує додаткових заходів для забезпечення безпеки.[15]

Створення доступнішої та дешевшої модульної системи підтримки користувачів на основі чат-бота є стратегічно вигідним рішенням. Воно не лише знижує витрати та підвищує ефективність існуючої інфраструктури, але й відкриває нові можливості для автоматизації та покращення комунікації. Завдяки повному контролю над розвитком та інтеграцією додаткових сервісів, чат-бот стає гнучким інструментом, який може адаптуватися до потреб бізнесу та сприяти його зростанню. Ще однією важливою перевагою чат бота, є той факт, що для них створено не мало фреймворків, які дуже прискорюють розробку ботів. Бот може створюватися і розповсюджуватися через месенджери для яких були створені відповідні інструменти. Таким чином не має потреби виділяти ресурси на створення нових, складних рішень. Тому бот може стати доволі дешевим в контексті інвестицій для створення власного інструменту, для взаємодії в рамках внутрішнього сервісу між виконавцями та клієнтами.

2.1.4 Початкові вимоги: підтримка створення заявок, робота з вкладеннями, нотифікації для користувачів, робота з створеними заявами, підтримка додаткових функцій(паркування) та персоналізація потреб

Для забезпечення ефективної та зручної підтримки користувачів було вирішено розробити чат-бота, який би відповідав специфічним потребам організації та був би інтегрований в систему. Початкові вимоги до цього бота включають ряд ключових функцій, спрямованих на покращення процесу обробки заявок, комунікації з користувачами та інтеграції додаткових сервісів. Нижче детально розглянуті основні вимоги до чат-бота.

1. Створення заявок з розподілом за відділами

1.1. Гнучкий вибір адресата заявки

- **Користувач самостійно обирає відділ:** При створенні заявки користувач повинен мати можливість вибрати, до якого відділу вона буде направлена:
 - **IT:** Технічна підтримка, вирішення проблем з обладнанням чи програмним забезпеченням. Чи запит на розробку, та внесення змін в існуючі системи.
 - **People:** Включає підрозділи **HR**, **Finance** та **FWS** (Facility Workplace Services).
- **Різні форми для різних відділів:** Кожен відділ має специфічні вимоги до інформації у заявці, тому форми повинні бути адаптовані відповідно до потреб:

2. Нотифікації для користувачів

2.1. Проактивні сповіщення

- **Зміна статусу заявки:** Користувач отримує повідомлення, коли його заявка переходить на новий етап обробки (прийнята, в роботі, вирішена тощо).
- **Коментарі від виконавців:** При додаванні коментарів до заявки виконавцем користувач негайно сповіщається про це.
- **Вбудовані повідомлення в чаті:** Сповіщення надходять безпосередньо через чат-бота, що забезпечує швидкий доступ до інформації.

3. Робота з вкладеннями

3.1. Підтримка вкладень при створенні заявок

- **Додавання файлів:** Користувач може прикріплювати файли (зображення, документи, скріншоти) при створенні заявки, що допомагає точніше описати проблему або запит.
- **Додавання додаткових файлів:** Можливість додавати вкладення після створення заявки.

4. Маніпуляції зі створеними заявками

4.1. Перегляд активних та закритих заявок

- **Історія заявок:** Користувач має доступ до списку всіх своїх заявок з можливістю фільтрації за статусом.
- **Деталі заявки:** Перегляд повної інформації про заявку, її статус, опис та назва разом з кодом. В майбутньому важливо додати останні декілька коментарів та список вкладень.

4.2. Оновлення та доповнення заявок

- **Додавання коментарів:** Користувач може залишати додаткові коментарі або уточнення до своєї заявки.
- **Оновлення опису:** Можливість редагувати опис заявки у випадку необхідності.
- **Додавання вкладень:** Прикріплення нових файлів після створення заявки.

5. Підтримка додаткових функцій

5.1. Сервіс паркування

- **Інтеграція з системою бронювання паркувальних місць:** Чат-бот надає можливість бронювати місця для паркування, що було одним із першочергових сервісів для запуску. Ця функція, була в іншого бота, написаного не на Python, і був в зовнішньому месенджері. Тому створення механізму паркування було одно з початкових завдань для нового бота.

5.2. Доступ до корисних посилань

- **Список корисних ресурсів:** Надання швидкого доступу до важливих внутрішніх посилань, таких як корпоративний портал, політики компанії, контакти відділів тощо.

- **Персоналізація:** Бажана можливість налаштувати список посилань відповідно до потреб користувача або його відділу.

6. Персоналізація та уніфікація платформ

6.1. Усунення розпорошеності ботів

- **Єдина платформа для взаємодії:** В процесі розробки було вирішено об'єднати функціональність різних ботів, які раніше працювали на різних платформах (наприклад, Telegram), в одному чат-боті. Бот повинен бути створений таким чином, щоб легко підтримувати нові функції.

Розробка чат-бота з такими початковими вимогами спрямована на покращення якості підтримки користувачів, підвищення ефективності внутрішніх процесів та створення єдиного центру для взаємодії співробітників з різними сервісами компанії. Забезпечення можливості створення розподілених заявок, підтримка роботи з вкладеннями, проактивні нотифікації, управління заявками та інтеграція додаткових функцій, таких як паркування та доступ до важливих посилань, створює зручний та ефективний інструмент для співробітників. Уніфікація платформ та персоналізація сервісів підкреслюють прагнення компанії до інновацій та орієнтованості на потреби своїх працівників.

2.2. Використані технології

2.2.1 Python

Як основа для написання бота була використана мова програмування Python, бо вона має багато переваг перед іншими мовами програмування в аспекті роботи з даними. А також активно підтримується як сама мова так і інструменти для неї зроблені.

Python є інтерпретованою, об'єктно-орієнтованою мовою програмування високого рівня з динамічною семантикою. Її вбудовані структури даних високого рівня, поєднані з динамічним набором типів та динамічним зв'язуванням, роблять її дуже привабливою для швидкої розробки програм, а також для використання як мови скриптів або мови

з'єднання, щоб з'єднати існуючі компоненти разом. Простий та легкий у вивченні синтаксис Python підкреслює зрозумілість коду та зменшує вартість підтримки програм. Python підтримує модулі та пакети, що сприяє модульності програм та повторному використанню коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідному або бінарному вигляді безкоштовно для всіх основних платформ та можуть бути вільно поширені.

Гвідо ван Россум почав працювати над мовою програмування Python наприкінці 1980-х років як наступник мови програмування ABC і випустив першу версію Python 0.9.0[16] в 1991 році. Python 2.0 був випущений у 2000 році. Python 3.0, який був випущений у 2008 році, був значним оновленням, яке не було повністю сумісним з попередніми версіями. Python 2.7.18, випущений у 2020 році, був останнім релізом Python 2[17].

Python - це багато парадигмова мова програмування. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато з їх функцій підтримують функціональне програмування та аспектно-орієнтоване програмування (включаючи мета програмування[18] та мета об'єкти). Багато інших парадигм підтримуються за допомогою розширень, включаючи дизайн за контрактом та логічне програмування[19].

Python використовує динамічний типізм та поєднання рахунку посилань та автоматичного збирача сміття з циклом виявлення для управління пам'яттю. Для прив'язки імен використовується динамічне розв'язування (пізніше зв'язування), яке пов'язує імена методів та змінних під час виконання програми.

В дизайні мови передбачено деяку підтримку функціонального програмування в традиції Lisp. Мова має функції `filter`, `map` і `reduce`; `comprehensions` для списків, словників, множин і генераторів виразів. В стандартній бібліотеці є два модулі (`itertools` та `functools`), які реалізують функціональні інструменти, запозичені з Haskell та Standard ML.

Основна філософія мови Python узагальнена в документі "The Zen of Python" (PEP 20) і включає афоризми, такі як[20]:

Красиве краще за потворне.

Явне краще за неявне.

Просте краще за складне.

Складне краще за заплутане.

Читабельність має значення..

Замість включення всієї функціональності безпосередньо до ядра мови, Python був розроблений як високоякісно розширювана завдяки модулям. Ця компактна модульність зробила його особливо популярним як засіб додавання програмованих інтерфейсів до існуючих додатків. Бачення Ван Россума про невелику мову ядра з великою стандартною бібліотекою та легко розширюваним інтерпретатором виникла з його роздратування на ABC, який прихильно сприймав протилежний підхід[21].

Python намагається мати простіший, менш заплутаний синтаксис та граматику, дозволяючи розробникам вибирати свою методологію програмування. На відміну від гасла Perl "є більше одного способу це зробити", Python прихильний до філософії "повинен бути один - і краще всього тільки один - очевидний спосіб це зробити". Алекс Мартеллі, член Фонду програмного забезпечення Python та автор книг про Python, написав: "Описуючи щось як 'хитре', в культурі Python це не вважається компліментом"[22].

Розробники Python прагнуть уникнути передчасної оптимізації та відкидають заплатки до не критичних частин CPython еталонної реалізації, які можуть забезпечити незначні підвищення швидкості за рахунок збереження чіткості. Якщо швидкість є важливою, програміст Python може перемістити функції, що вимагають часу, до модулів розширення, написаних на мовах, таких як C; або використовувати PyPy, компілятор з миттєвою компіляцією. Також доступний Cython, який перекладає скрипт Python у C і виконує прямі виклики API на рівні C до інтерпретатора Python.

Розробники Python прагнуть уникнути передчасної оптимізації та відкидають заплатки до не критичних частин CPython еталонної реалізації, які можуть забезпечити незначні підвищення швидкості за рахунок

збереження чіткості. Якщо швидкість є важливою, програміст Python може перемістити функції, що вимагають часу, до модулів розширення, написаних на мовах, таких як C; або використовувати PyPy, компілятор з миттєвою компіляцією. Також доступний Cython, який перекладає скрипт Python у C і виконує прямі виклики API на рівні C до інтерпретатора Python.

У спільноті Python поширене новотвірне слово "pythonic", яке має широкий спектр значень, пов'язаних зі стилем програмування. "Pythonic" код може добре використовувати ідіоми Python, бути природним або відображати вміння використовувати мову програмування, або відповідати мінімалістичній філософії Python і акцентувати на читабельності. Код, який складно зрозуміти або виглядає як грубий переклад з іншої мови програмування, називається "unpythonic".

З 2003 року Python стабільно потрапляє до десятки найпопулярніших мов програмування за версією індексу спільноти програмістів TIOBE, де на грудень 2022 року став найпопулярнішою мовою програмування (перед C, C++ та Java). У 2007, 2010, 2018 та 2020 роках вона була обрана мовою програмування року за "найбільше зростання рейтингу за рік" (єдина мова, яка це зробила чотири рази на 2020 рік).

Одне емпіричне дослідження встановило, що скриптові мови, такі як Python, є більш продуктивними, ніж традиційні мови, такі як C та Java, для задач програмування, пов'язаних з маніпулюванням рядків та пошуком у словнику, і визначили, що використання пам'яті часто "краще, ніж у Java, і не набагато гірше, ніж у C або C++".

Великі організації, що використовують Python, включають Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify, а також деякі менші суб'єкти, такі як ILM і ІТА. Соціальний новинний сайт Reddit був написаний переважно на Python.

Обґрунтування вибору Python

Python був обраний як основа написання боту. Він, як мова програмування, став ядром створеного сервісу. Вся логіка і взаємодія різних компонентів, повністю

написана цією мовою програмування. Наведу кілька причин, чому був використаний саме Python.

1. Універсальність та гнучкість

Однією з головних причин вибору **Python** стала його універсальність. Python є мовою загального призначення, що дозволяє вирішувати широкий спектр завдань — робота з ботом, створення ендпоінтів, переадресація. Завдяки великій кількості бібліотек та фреймворків, таких як **requests**, **aiohttp**, **pandas** та інші, можна швидко реалізовувати взаємодію з API, обробку даних та інші необхідні функції. Це забезпечує єдність технологічного стеку та спрощує процеси розробки та підтримки.

2. Повторне використання існуючих напрацювань

Під час розробки чат-бота були використані конструкції та модулі, розроблені інженерами компанії для інших задач, пов'язаних з інтеграцією та обробкою даних. Це дозволило уникнути "винайдення велосипеда" і значно скоротити час та ресурси на розробку.

3. Підтримка Microsoft Bot Framework SDK

Останнім часом **Microsoft Bot Framework SDK** припинив активну підтримку своїх бібліотек на деяких мовах програмування, зосередившись на Python та ще кількох мовах. Це стало важливим фактором при виборі мови, оскільки використання Python гарантує сумісність та підтримку з боку Microsoft, а також доступ до актуальної документації та оновлень.

4. Компетенція команди

Вся команда розробників має досвід роботи з Python, що сприяло ефективній співпраці та розподілу завдань. Це дозволило:

- **Швидко розпочати розробку** без необхідності додаткового навчання чи перекваліфікації.
- **Забезпечити якісне тестування та рев'ю коду**, оскільки інші спеціалісти могли легко перевіряти та вдосконалювати код.

- **Знизити залежність від окремих розробників:** у випадку потреби в удосконаленнях або виправленні помилок інші члени команди можуть безперешкодно підтримувати та розвивати проект.

Вибір Python для розробки чат-бота був обґрунтованим та стратегічно вигідним рішенням. Його універсальність, наявність потужних бібліотек, сумісність з Microsoft Bot Framework SDK та компетенція команди в цій мові дозволили ефективно реалізувати поставлені завдання, забезпечити якісну підтримку проекту та створити основу для подальшого розвитку та масштабування системи.

2.2.2 Microsoft Bot Framework SDK

Microsoft Bot Framework SDK — це набір інструментів та бібліотек, розроблених компанією Microsoft, які призначені для створення інтерактивних та інтелектуальних чат-ботів. SDK надає розробникам можливість створювати ботів, які можуть взаємодіяти з користувачами через різні канали комунікації, такі як Microsoft Teams, Skype, Slack, Facebook Messenger, Telegram та інші [23].

Microsoft Bot Framework SDK з'явився як частина ширшої ініціативи Microsoft, спрямованої на створення екосистеми інтелектуальних помічників та розмовних інтерфейсів. Історія його виникнення та розвитку сягає середини 2010-х років, коли спостерігався стрімкий ріст інтересу до технологій штучного інтелекту, машинного навчання та обробки природної мови. Компанія Microsoft, маючи великий досвід у цих сферах (через свої когнітивні сервіси та інші продукти), побачила можливість створити платформу, яка б полегшила розробникам процес створення, тестування та підтримки складних чат-ботів.

На початку Microsoft Bot Framework був представлений у 2016 році як набір інструментів, що включали **Bot Builder SDK**[23], Bot Connector[23] та Bot Directory[23]. Ці компоненти забезпечували:

1. Зручну розробку бота:

Bot Builder SDK надавав бібліотеки для .NET та Node.js, полегшуючи створення логіки бота, обробку діалогів, керування станом і інтеграцію зі

службами розпізнавання мови (LUIS). Пізніше з'явилася підтримка й інших мов програмування та платформ.

2. Підключення до різноманітних каналів:

Bot Connector дозволяв одним інтерфейсом забезпечити взаємодію з багатьма каналами (Skype, Slack, Microsoft Teams, Facebook Messenger, Telegram, Email та інші). Це означало, що розробник не повинен був писати окремий код для кожної платформи, а міг через спільний інтерфейс під'єднати свого бота до різних каналів комунікації.

3. Управління та аналітика:

Bot Framework забезпечував інструменти для адміністрування, ведення журналів і відстеження продуктивності. Згодом з'явилися й додаткові компоненти, такі як Bot Framework Emulator для локального тестування бота та Application Insights для збору аналітики.

З часом Microsoft продовжувала розвивати Bot Framework, додаючи нові можливості й інструменти. Оновлені версії SDK перейшли до більш сучасної архітектури, отримали підхід "component-based" та підтримку Bot Framework Composer – інструменту для візуального проектування діалогів і інтерактивних сценаріїв. Цей розвиток відображає загальну стратегію Microsoft щодо створення максимально гнучкої та масштабованої платформи.

Основні особливості Microsoft Bot Framework SDK:

1. Мультиплатформність:

Завдяки гнучкому SDK можна розробляти ботів на різних мовах програмування і розгортати їх на різних платформах (Azure, локальні сервери чи хмарні сервіси інших провайдерів).

2. Легкість інтеграції з когнітивними сервісами:

Bot Framework SDK тісно інтегрується з Microsoft Cognitive Services, зокрема з LUIS (Language Understanding) та QnA Maker. Це дозволяє ботам розуміти наміри користувача, аналізувати натуральну мову та динамічно генерувати відповіді.

3. Модель діалогів та станів:

Однією з ключових переваг є готова структура для побудови діалогових взаємодій, керування контекстом розмови та збереження стану між повідомленнями. Це полегшує створення складних сценаріїв взаємодії.

4. Інструменти для розробників:

Bot Framework Emulator, Bot Framework Composer та інші утиліти суттєво полегшують розробку, налагодження та тестування. Розробник може зручно перевірити роботу бота ще до того, як він вийде на публіку.

MS Bot SDK: Причини вибору цього SDK для інтеграції з Microsoft Teams

Microsoft Bot Framework SDK був обраний через низку важливих причин, що цілісно вписуються в інфраструктурне середовище компанії та відповідають її практикам. Оскільки основним інструментом внутрішньої комунікації є Microsoft Teams, інтеграція чат-бота безпосередньо в цей месенджер стала логічним кроком, а активне використання сервісів Microsoft Azure суттєво полегшило сумісність і зручність роботи з іншим корпоративним програмним забезпеченням. Відмінною рисою MS Bot SDK є те, що його було спеціально розроблено для інтеграції з Microsoft Teams, завдяки чому розробники отримують гарантовану сумісність, підтримку усіх оновлень та можливість працювати з унікальними функціями платформи. Так, бот може використовувати спеціальні команди, кнопки дій, адаптивні картки та рекомендовані відповіді, що робить взаємодію користувачів із ним більш інтуїтивною та ефективною. Глибока інтеграція з Azure спрощує збереження стану розмови, дозволяє використовувати хмарні сервіси для аналізу мови й даних, а також забезпечує масштабованість системи при зростанні навантажень. Крім того, MS Bot SDK не обмежується лише Microsoft Teams: за потреби можна додати підтримку Slack, Facebook Messenger, Telegram та інших платформ, що відкриває перспективи розширення в майбутньому. Рішення від Microsoft вирізняється надійною підтримкою, детальною документацією та доступом до прикладів коду, що спрощує розробку та обслуговування. Важливим чинником лишається дотримання корпоративних стандартів безпеки та

використання єдиної екосистеми, яка покликана захистити дані й гарантувати безпечну автентифікацію користувачів. Усе це робить Microsoft Bot Framework SDK оптимальним вибором для розробки та впровадження чат-бота.

2.2.3 Microsoft Teams

Microsoft Teams — це додаток для командної співпраці, розроблений Microsoft як частина сімейства продуктів Microsoft 365. Він надає можливості для командного чату та відеоконференцій, зберігання файлів, а також інтеграції власних і сторонніх додатків та сервісів.[24]

Teams замінив інші платформи для бізнес-комунікацій та співпраці від Microsoft, зокрема Skype for Business та Microsoft Classroom.

Під час пандемії COVID-19 Teams, а також програмне забезпечення, таке як Zoom, Slack, Google Meet та інші, здобули значну популярність у зв'язку з переходом багатьох зустрічей у віртуальне середовище.[25]

Станом на січень 2023 року платформа мала близько 280 мільйонів активних користувачів щомісяця.[26]

Ідея створення Teams виникла в контексті тенденції до переходу на дистанційні та гібридні формати роботи, а також попиту на інструмент, що об'єднував би в собі текстові чати, відеоконференції, спільний доступ до файлів і гнучкі інструменти для планування. Microsoft приділила особливу увагу інтеграції з власною екосистемою, зокрема з Office 365[24] (нині — Microsoft 365), Outlook та SharePoint, створивши єдиний робочий простір для щоденних бізнес-процесів.

Історично Teams став продовженням і логічним наступником Skype for Business, поступово витіснивши його та перетворившись на головний хаб для командної роботи в межах хмарної екосистеми Microsoft. З моменту запуску він динамічно розвивається, отримуючи нові функції, покращення в безпеці й продуктивності, а також розширену інтеграцію з іншими сервісами. Протягом останніх років Teams зарекомендував себе як незамінний інструмент для віддаленої та розподіленої роботи, ставши ключовим

елементом корпоративної інфраструктури для безлічі компаній у всьому світі.

Особливості та можливості Teams:

Чати

Teams дозволяє користувачам спілкуватися у двосторонніх чатах з одним або кількома учасниками. Учасники можуть надсилати текстові повідомлення, емодзі, стікери, GIF-файли, а також ділитися посиланнями та файлами.

Команди

Функція "Команди" дозволяє спільнотам, групам або командам працювати разом у спільному просторі, де обговорюються повідомлення та цифровий контент на конкретні теми. Члени команди можуть приєднатися через запрошення адміністратора чи шляхом спільного доступу до URL-посилання.

Канали

Канали дозволяють членам команди спілкуватися без використання електронної пошти чи SMS. Користувачі можуть відповідати на пости текстом, зображеннями, GIF-файлами та мемами. Прямі повідомлення дозволяють надсилати приватні повідомлення певним користувачам. Канали підтримують конектори для інтеграції зі сторонніми сервісами, такими як Mailchimp, Facebook Pages, Twitter, Power BI та Bing News.

Групові розмови

Ad-hoc групи можна створювати для миттєвого обміну повідомленнями, голосовими (VoIP) і відеодзвінками в межах клієнтського програмного забезпечення.

Зустрічі

Зустрічі можна запланувати з кількома учасниками, які можуть ділитися аудіо, відео, повідомленнями та контентом. Можливо підключення через посилання на зустріч. Функції запису та транскрипції дозволяють автоматично створювати протоколи. Teams має плагін для Microsoft Outlook, щоб планувати зустрічі Teams безпосередньо через Outlook.

Кімнати для обговорень

Кімнати для обговорень (Breakout Rooms) дозволяють розділити зустріч на невеликі групи. Це корисно для тренінгів або середовищ, де всі учасники не можуть говорити одночасно. Час для кімнат може бути встановлений ведучим, після закінчення якого всі учасники автоматично повертаються до основної зустрічі.

Front Row

Front Row змінює макет екрана користувача, розміщуючи основного спікера чи контент у центрі галереї, а відеопотоки інших учасників – під ним.

Освіта

Microsoft Teams для освіти дозволяє вчителям розповсюджувати, коментувати та оцінювати завдання учнів через вкладку Assignments для користувачів Office 365 Education. Також доступна інтеграція із формами для тестів через Office Forms.

Протоколи

Microsoft Teams працює на основі низки власних протоколів Microsoft. Відеоконференції використовують протокол MNP24, відомий зі Skype. Для підключення SIP та H.323-клієнтів потрібні спеціальні шлюзи.[27] Завдяки технології Interactive Connectivity Establishment (ICE) клієнти можуть підключатися через NAT та брандмауери, якщо пряме з'єднання неможливе.

Інтеграції

Teams підтримує інтеграції через Microsoft AppSource, маркетплейс для додатків. У 2020 році Microsoft об'єднався з KUDO для забезпечення мовного перекладу під час зустрічей.[28] У червні 2022 року було додано поліпшення аудіо через AI, яке усуває фонові шуми та непотрібний зворотний зв'язок.[29]

Microsoft Teams: Чому саме Teams було вибрано як платформу для бота

В основі вибору Microsoft Teams як платформи для бота лежала необхідність інтегрувати його в уже існуюче робоче середовище компанії. Майже всі співробітники активно використовували Teams, оскільки цей інструмент входить до складу пакету Microsoft 365, що включає Excel, Word

та інші необхідні застосунки. Це означало, що кожен фахівець мав гарантований доступ до Teams і, відповідно, до всіх внутрішніх сервісів, пов'язаних із ботом. Завдяки знайомому інтерфейсу Teams, час на навчання суттєво скоротився, а ефективність взаємодії з новим сервісом зросла.

Ще одним ключовим фактором стала надійність та безпека цієї платформи, що повністю відповідала корпоративним стандартам. Використання Teams забезпечувало централізоване управління доступом, а також дотримання вимог безпеки при роботі з чутливою інформацією. Важливою перевагою було й те, що в системі Teams уже містились унікальні ідентифікатори користувачів, такі як корпоративні електронні адреси. Це спростило персоналізацію послуг бота, автоматичне заповнення необхідних даних та прискорило процеси, пов'язані зі зверненнями, наприклад, при створенні тікетів.

Завдяки інтеграції бота у Teams внутрішні сервіси компанії стали більш доступними та зручними для співробітників. Замість додаткових платформ чи месенджерів, які вимагали б нових впроваджень і витрат, було вирішено максимально використовувати інструменти, що вже існували. Це не лише зменшило витрати та оптимізувало робочий процес, а й забезпечило чітку послідовність дій у рамках заданих процедур, покращило точність введення даних та спростило контроль за станом тікетів і виконанням завдань.

Оскільки Microsoft Teams є одним із найпоширеніших корпоративних рішень, його підтримка з боку Microsoft є постійною й динамічною. Це дозволяє платформі регулярно оновлюватися, реагувати на нові виклики, а компанії – отримувати актуальні інструменти для вдосконалення внутрішніх процесів. Крім того, Teams органічно інтегрується з іншими сервісами Microsoft, такими як Outlook, SharePoint чи OneDrive, створюючи єдину екосистему. Завдяки цьому всі процеси, від доступу до файлів і спільної роботи з документами до відстеження тікетів, відбуваються в єдиному середовищі з ясними правилами й зрозумілою навігацією.

Таким чином, використання Microsoft Teams для впровадження бота оптимально відповідало потребам компанії. Воно гарантувало загальну

доступність сервісів для всіх співробітників, безпечне та централізоване управління, а також легку інтеграцію з іншими сервісами Microsoft. Завдяки знайомому інтерфейсу Teams і можливості автоматизації рядових операцій процеси стали ефективнішими, прозорішими та менш затратними. У результаті компанія отримала надійний, зручний і безпечний інструмент взаємодії зі своїми внутрішніми сервісами, що відповідає потребам і бізнесу, і його працівників.

2.2.4 Jira

Jira — це пропрієтарний продукт, розроблений компанією Atlassian, що дозволяє здійснювати відстеження помилок, завдань та гнучке управління проектами. Jira використовується багатьма клієнтами та користувачами по всьому світу для управління проектами, часом, вимогами, завданнями, помилками, змінами, кодом, тестами, релізами та спринтами.

Назва

Назва продукту походить від другої та третьої складів японського слова "Gojira", що в перекладі означає "Годзилла".[30] Назва виникла як прізвисько, яке розробники Atlassian використовували для Bugzilla — попередньої системи відстеження помилок, що застосовувалася внутрішньо.[30]

Опис

За даними Atlassian, Jira використовується для **відстеження завдань та управління проектами**. [31] Організації, що застосовували Jira для управління помилками та проектами, включають Fedora Commons, Hibernate, а також Apache Software Foundation, яка використовує як Jira, так і Bugzilla. Jira надає інструменти для міграції з конкурентної системи Bugzilla.

Jira пропонується у трьох пакетах:[32]

1. **Jira Software** – основний продукт з функціями гнучкого управління проектами (колишній окремий продукт: Jira Agile).
2. **Jira Service Management** – для ІТ-операцій або бізнес-сервісних служб.

3. **Jira Align** – для стратегічного управління продуктами та портфелями.

Jira написана мовою **Java** та використовує технологічний стек Pico Inversion of Control, Apache OFBiz Entity Engine та WebWork 1. Для **віддалених викликів процедур (RPC)** Jira підтримує REST, SOAP та XML-RPC інтерфейси.[33]

Інтеграція:

Jira інтегрується з системами контролю версій, такими як Clearcase, CVS, Git, Mercurial, Perforce, Subversion і Team Foundation Server. Додаток підтримує кілька мов, зокрема англійську, французьку, німецьку, японську та іспанську. Jira реалізує **Networked Help Desk API** для спільного використання тікетів підтримки між різними системами відстеження завдань.

Ліцензування

Jira є комерційним програмним продуктом, що може працювати **локально** (on-premises) або як **хмарний сервіс**. [34] Спочатку Jira була інструментом з відкритим кодом, доступним для завантаження. Згодом Atlassian перевела продукт у закритий формат і побудувала бізнес навколо нього.

Безкоштовні ліцензії: Atlassian надає Jira безкоштовно для проєктів з відкритим вихідним кодом, що відповідають певним критеріям, а також для некомерційних, неурядових, неполітичних, неакадемічних та неприбуткових організацій. Повний вихідний код доступний для користувачів під **ліцензією розробника** для внесення змін.[34]

Чому Jira залишається основною системою для управління заявками, та як API допомагає інтегрувати Jira з ботом

Jira залишається основною системою для управління заявками завдяки її широкому використанню в компанії, зручності інтерфейсу та вже налаштованим процесам. Співробітники добре знайомі з принципами роботи Jira, що робить її природним вибором для подальшого розвитку внутрішніх сервісів. Однією з ключових переваг є високий рівень налаштовуваності: система дозволяє гнучко адаптувати робочі процеси, створювати власні поля, статуси, типи задач, а також застосовувати інші параметри, оптимізуючи роботу під специфічні потреби компанії. Додатково, Jira пропонує потужний API, що відкриває можливість

інтегрувати її з ботами та сторонніми сервісами, отримувати й змінювати дані, а також налаштовувати майбутнє розширення функціональності без значних зусиль. Важливою рисою є функція автоматизації, яка дає змогу створювати сценарії автоматичного надсилання інформації до бота за певних умов, гнучко налаштовувати зміст і напрямок передачі даних, а також стежити за коректністю процесів завдяки журналам подій. Попри певні недоліки, наприклад проблеми з кодуванням символів, які не належать до латиниці, це обмеження потенційно може бути усунуте в майбутніх оновленнях, роблячи систему ще більш зручною. Інтеграція Jira з ботом завдяки API надає змогу створювати та оновлювати задачі безпосередньо через бота, додавати коментарі чи вкладення, отримувати інформацію про статус, повідомляти користувачів про зміни та навіть персоналізувати їхній досвід взаємодії.

2.2.5 PostgreSQL

В боті, окрім коду, є ще одна не менш важлива річ, це база даних, на данному етапі вона виконує функцію персоналізації, а також поєднувача між різними системами. Такою базою є PostgreSQL.

PostgreSQL є безкоштовною та відкритою реляційною системою управління базами даних (RDBMS), яка наголошує на розширюваності та відповідності SQL. Вона була спочатку названа POSTGRES, що походить від її походження як наступник бази даних Ingres, розробленої в Університеті Каліфорнії, Берклі[35][36]. У 1996 році проект був перейменований на PostgreSQL, щоб відображати його підтримку SQL. Після перегляду в 2007 році, розробницька команда вирішила залишити назву PostgreSQL та псевдонім Postgres.

PostgreSQL має можливість виконання транзакцій з властивостями атомарності, послідовності, ізоляції та стійкості (ACID), автоматично оновлюваними видами, матеріалізованими видами, тригерами, зовнішніми ключами та збереженими процедурами[37]. Вона розроблена для роботи з

різними завантаженнями, від одиночних машин до складів даних або веб-сервісів з багатьма паралельними користувачами. PostgreSQL була базою даних за замовчуванням для macOS Server та також доступна для Linux, FreeBSD, OpenBSD та Windows.

PostgreSQL керує конкурентністю за допомогою багатOVERСІЙНОГО управління конкурентністю (MVCC), що дає кожній транзакції "снэпшот" бази даних, дозволяючи внесення змін без впливу на інші транзакції. Це в значній мірі усуває необхідність у читаючих блокуваннях та забезпечує дотримання принципів ACID. PostgreSQL пропонує три рівні ізоляції транзакцій: Читання комітів, Повторювані читання та Серіалізовані. Оскільки PostgreSQL є імунним до "dirty reads", запит на рівень ізоляції транзакції Read Uncommitted замість цього надає рівень read committed. PostgreSQL підтримує повну серіалізованість за допомогою методу серіалізованого знімку (SSI).

У PostgreSQL схема містить усі об'єкти, крім ролей та просторів таблиць. Схеми діють як простори імен, дозволяючи об'єктам з однаковим іменем існувати в одній базі даних. За замовчуванням відповідно до відновлення бази даних є схема з іменем public, але можна додавати будь-які додаткові схеми, і схема public не є обов'язковою.

Параметр `search_path` визначає порядок, в якому PostgreSQL перевіряє схеми для об'єктів без вказаної схеми. За замовчуванням він встановлений на `$user, public` (`$user` відноситься до поточного користувача бази даних). Цей параметр можна встановити на рівні бази даних або ролі, але оскільки він є параметром сесії, його можна вільно змінювати (навіть декілька разів) під час сесії клієнта, що впливає лише на цю сесію.

Неіснуючі схеми, перераховані в `search_path`, при пошуку об'єктів, безпомилково пропускаються.

Нові об'єкти створюються в першій дійсній схемі (одній, яка існує в даний момент), яка з'являється в `search_path`..

Типи даних

Підтримуються широкі можливості нативних типів даних, включаючи:

Boolean
Arbitrary-precision numerics
Character (text, varchar, char)
Binary
Date/time (timestamp/time with/without time zone, date, interval)
Money
Enum
Bit strings
Text search type
Composite
HStore, an extension enabled key-value store within PostgreSQL
Arrays (variable-length and can be of any data type, including text and composite types) up to 1 GB in total storage size
Geometric primitives
IPv4 and IPv6 addresses
Classless Inter-Domain Routing (CIDR) blocks and MAC addresses
XML supporting XPath queries
Universally unique identifier (UUID)
JavaScript Object Notation (JSON), and a faster binary JSONB (not the same as BSON)

Крім того, користувачі можуть створювати власні типи даних, які, як правило, можуть бути повністю індексовані за допомогою інфраструктури індексування PostgreSQL - GiST, GIN, SP-GiST. Приклади таких типів даних включають географічну інформаційну систему (GIS) від проекту PostGIS для PostgreSQL.

Зовнішні обгортки даних

PostgreSQL може посилатися на інші системи для отримання даних через зовнішні обгортки даних (FDW). Це можуть бути будь-які джерела даних, такі як файлова система, інша система управління реляційними базами даних (RDBMS) або веб-сервіс. Це означає, що звичайні запити до бази даних

можуть використовувати ці джерела даних як звичайні таблиці, навіть об'єднувати декілька джерел даних разом.

Інтерфейси

Для підключення до додатків PostgreSQL містить вбудовані інтерфейси libpq (офіційний інтерфейс додатків на C) та ECPG (вбудована система на мові C). Сторонні бібліотеки для підключення до PostgreSQL доступні для багатьох мов програмування, включаючи C++, Java, Julia, Python, Node.js, Go та Rust..

База даних PostgreSQL: Обґрунтування вибору PostgreSQL як бази даних

PostgreSQL використовується як центральна база даних для зберігання та управління важливими корпоративними даними, відіграючи ключову роль у функціонуванні організації. Завдяки цій СУБД бот отримує доступ до актуальної інформації про користувачів, включно з унікальними ідентифікаторами різних систем, посадами та відділами, що дозволяє індивідуально налаштовувати функціонал сервісу залежно від ролей та прав доступу. Це забезпечує релевантність відображуваних даних. Та можливість вільної інтеграції функцій і даних між різними системами

Надійність та масштабованість PostgreSQL роблять її ідеальним вибором для робочого середовища, де обсяги інформації та кількість запитів постійно зростають. Вона без проблем витримує підвищені навантаження, зберігаючи стабільність і продуктивність. Окрім цього, глибока інтеграція з мовою програмування Python спрощує процес розробки та підтримки чат-бота, дозволяючи ефективно взаємодіяти з базою даних. Потужні можливості SQL-запитів, які пропонує PostgreSQL, сприяють реалізації складних вибірок, динамічному наданню інформації та іншим сценаріям, необхідним для якісної персоналізації.

Таким чином, вибір PostgreSQL зумовлений трьома основними чинниками: можливістю інтегруватися з існуючими корпоративними даними для взаємодії між різними системами, здатністю масштабуватися відповідно до потреб бізнесу та

легкістю взаємодії з ботом завдяки сумісності з Python і підтримці складних запитів.

2.2.6 Azure

Microsoft Azure, або просто **Azure** — це платформа хмарних обчислень, розроблена компанією Microsoft. Вона надає можливості для управління, доступу та розробки застосунків і сервісів для фізичних осіб, компаній та урядів через свою глобальну інфраструктуру.

Azure також забезпечує функціональність, яка зазвичай не включена до інших хмарних платформ, включаючи програмне забезпечення як сервіс (SaaS), платформу як сервіс (PaaS) та інфраструктуру як сервіс (IaaS). Microsoft Azure підтримує багато мов програмування, інструментів та фреймворків, зокрема як специфічні для Microsoft, так і сторонні програмні рішення та системи.

Azure було вперше представлено на Professional Developers Conference (PDC) у жовтні 2008 року під кодовою назвою "Project Red Dog".[38] Офіційний запуск відбувся у лютому 2010 року під назвою Windows Azure, а 25 березня 2014 року платформа отримала нову назву — Microsoft Azure.[39][40]

Основні послуги та можливості

Microsoft Azure пропонує понад 600 різних сервісів, серед яких:

1. Обчислювальні послуги

Azure дозволяє запускати віртуальні машини з операційними системами Windows та Linux, а також використовувати платформу як сервіс (PaaS) для розробки й розгортання застосунків.

2. Зберігання даних

Сервіси включають:

- Blob-хмарне сховище для неструктурованих даних,
- Файлове сховище,
- Черги повідомлень для обміну даними,

- Таблиці для зберігання структурованих даних.

3. Мережеві послуги

Azure надає:

- Віртуальні мережі,
- Балансування навантаження,
- DNS-сервіси, забезпечуючи безпечне й надійне підключення.

4. Бази даних

Сервіси баз даних включають:

- Керовані реляційні бази даних, такі як Azure SQL Database,
- NoSQL-рішення, наприклад, Azure Cosmos DB.

5. Аналітика та штучний інтелект

Azure надає можливості:

- Машинного навчання,
- Аналізу великих даних,
- Когнітивні сервіси для обробки природної мови та розпізнавання зображень.

6. Інтернет речей (IoT)

Azure підтримує підключення, відстеження та управління IoT-пристроями, а також обробку даних, що надходять від них.

Глобальна інфраструктура

Azure має широку географічну присутність завдяки центрам обробки даних по всьому світу, що забезпечує високу доступність та відмовостійкість сервісів. Станом на 2018 рік платформа була доступна в 54 регіонах. Microsoft також стала першим великим хмарним провайдером, який відкрив центри обробки даних у Африці, зокрема у двох регіонах Південної Африки.

Безпека та відповідність стандартам

Microsoft Azure приділяє особливу увагу захисту даних та відповідності міжнародним стандартам. Платформа сертифікована за багатьма програмами відповідності, серед яких:

- ISO 27001,
- HIPAA.

Azure пропонує інструменти для забезпечення конфіденційності та захисту даних клієнтів.

Інновації та майбутній розвиток

Microsoft активно розвиває Azure, інтегруючи передові технології, такі як квантові обчислення. У 2021 році була представлена служба Azure Quantum, що надає доступ до квантових апаратних і програмних засобів.

Таким чином, Microsoft Azure є однією з провідних хмарних платформ, що пропонує широкий спектр послуг для вирішення завдань у галузі інформаційних технологій, розробки застосунків та управління даними.

Причини використання Azure для розміщення бота

Microsoft Azure відіграє ключову роль у забезпеченні роботи чат-бота в середовищі Microsoft Teams. Перш за все, саме завдяки Azure можливо зареєструвати бота та налагодити його зв'язок із командним простором. Реєстрація в Azure слугує основою для ідентифікації бота, а також дає змогу Teams розпізнавати його й налаштовувати коректну взаємодію. Це означає, що будь-який запит від користувачів у межах Teams автоматично спрямовується до серверів, де розгорнуто бот, забезпечуючи безперервну й оперативну комунікацію.

Окрім цього, Azure надає інструменти для безпечної автентифікації та авторизації бота, що гарантує захищений доступ до внутрішніх функцій і даних. Завдяки вбудованим можливостям контролю доступу, розробники отримують змогу керувати ключами, токенами й аналізувати роботу бота, відстежуючи журнали подій і збираючи аналітичні дані для подальшого вдосконалення його функціональності.

Додатковою перевагою використання Azure є можливість інтегрувати сервіси зберігання даних, наприклад Azure Blob Storage. Це відкриває шлях до надійного збереження файлів, стану розмов та іншої інформації, необхідної для більш персоналізованого досвіду користувачів. У результаті, розгортання бота на базі Azure спрощує процеси масштабування та

підтримки, забезпечуючи стабільну роботу й гнучкість у майбутньому розвитку сервісу.

Таким чином, інтеграція з Azure є вирішальним чинником для ефективного функціонування чат-бота в Microsoft Teams, дозволяючи оптимізувати керування доступом, зберігати дані в безпечному хмарному середовищі та надавати користувачам якісний, безперебійний досвід взаємодії з ботом.

2.2.7 Zoho Survey

Zoho Survey – це інструмент для створення онлайн-опитувань та збору даних, розроблений компанією Zoho, яка відома широким спектром хмарних бізнес-рішень. Продукт призначений для організацій, дослідників, маркетологів та будь-кого, хто прагне швидко отримати зворотний зв'язок, проаналізувати потреби клієнтів чи оцінити думку внутрішніх команд. Завдяки Zoho Survey можна створювати персоналізовані анкети, налаштовувати логіку переходів між питаннями, застосовувати різноманітні типи питань і вбудовувати форми на веб-сайти або ділитися ними через соціальні мережі та електронну пошту.

Історія Zoho Survey пов'язана із загальною еволюцією екосистеми Zoho. Компанія Zoho, заснована в 1996 році в Індії, з часом перетворилася на глобального гравця у сфері хмарних програмних рішень. Вона здобула популярність завдяки своїм веб-додаткам для продуктивності, CRM-системам, інструментам для бухгалтерського обліку, управління персоналом та багатьом іншим рішенням. Кожен продукт Zoho інтегрований у спільну інфраструктуру, створюючи єдину екосистему з єдиним обліковим записом користувача. Zoho Survey з'явився як частина цієї великої хмарної платформи, надаючи інструментарій для збору аналітики й формування інформованих бізнес-рішень.

Особливістю інфраструктури Zoho є її цілісність та взаємопов'язаність. Користувачі можуть легко поєднувати Zoho Survey з іншими продуктами Zoho – наприклад, із Zoho CRM чи Zoho Campaigns – щоб перенаправляти результати опитувань безпосередньо у воронку продажів або надсилати персоналізовані кампанії залежно від отриманих відповідей. Такий підхід забезпечує безперебійний

інформаційний потік усередині компанії та дозволяє якісніше керувати клієнтськими взаємодіями, маркетингом і внутрішніми процесами.

Компанія Zoho відома тим, що орієнтується на потреби малого та середнього бізнесу, надаючи доступні та зручні рішення з високим рівнем безпеки та контролю даних. Вона є приватною компанією, що активно інвестує у власну інфраструктуру та науково-дослідні розробки. Філософія Zoho полягає в тому, щоб створювати інструменти, які допомагають бізнесам зростати, приймати рішення на основі даних та забезпечувати кращу взаємодію з клієнтами, а Zoho Survey – один із ключових компонентів цього стратегічного підходу.

Zoho Survey: форма для створення тікетів, та контролю за тікетами

Zoho Survey став ключовим елементом у процесі створення та обробки тікетів у компанії, оскільки цей інструмент був уже частиною наявних програмних рішень і не вимагав додаткових витрат на впровадження чи інтеграцію. Його інтерфейс виявився набагато зручнішим для створення тікетів порівняно з використанням бота в Microsoft Teams, де багатокрокові сценарії взаємодії могли дратувати користувачів та збільшували ризик помилок. У той час як розробка складних послідовностей у SDK для бота могла бути незручною, Zoho Survey пропонував інтуїтивно зрозумілий механізм заповнення форм, дозволяючи користувачам швидко та безпомилково надавати необхідну інформацію.

Гнучкість Zoho Survey проявляється у можливості адаптувати форми під конкретні потреби, створюючи різноманітні поля, додаткові інструкції чи підказки для користувачів. Такий підхід сприяв підвищенню точності введених даних та розширенню можливостей взаємодії. Окрім цього, використання Zoho Survey забезпечило надійність та безперервність процесу подання заявок. Навіть за умов тимчасової недоступності бота система продовжувала збирати тікети, які зберігалися у Zoho Survey, а у випадку технічних несправностей оператори могли вручну перенести їх у Jira, гарантуючи збереження та опрацювання всіх звернень.

Інтеграція з Jira здійснювалася завдяки спеціально розробленому бекенду, що автоматично передавав усі отримані з форм Zoho Survey дані безпосередньо до системи управління задачами. Такий підхід мінімізував навантаження на персонал та прискорив обробку запитів. Водночас було вирішено проблему з обробкою вкладень: оскільки передати файли безпосередньо з Zoho Survey у Jira виявилося неможливо, їх почали надсилати боту, який уже інтегрував вкладення в створені задачі. Хоча це й додавало ще один крок у процесі, підхід загалом дозволив зберегти всю функціональність і повноту інформації.

Здатність Zoho Survey реалізовувати складніші логічні схеми та процеси порівняно з ботом у Microsoft Teams також покращила користувацький досвід. Користувачі могли надавати детальнішу інформацію, а адміністратори – налаштовувати форми відповідно до специфічних потреб команди чи проєкту. У підсумку використання Zoho Survey стало стратегічно обґрунтованим рішенням, яке спростило процеси підтримки, зробило їх більш надійними, масштабованими та приємними для користувачів.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА: РОЗРОБКА ТА ВПРОВАДЖЕННЯ МОДУЛЬНОГО ЧАТ-БОТА

3.1. Архітектура та функціональні можливості модульного чат-бота

Чат-бот був розроблений як модульна система, що складається з кількох основних функціональних блоків. Кожен модуль відповідає за конкретний набір задач та інтеграцій, що дозволяє легко розширювати та підтримувати систему в майбутньому.

Основні функції чат-бота:

Ключовий елемент архітектури чат-бота — це дерево діалогів, яке визначає можливі сценарії взаємодії з користувачем залежно від його запиту. Чат-бот побудований на основі логічно організованої структури, що дозволяє користувачам швидко знайти потрібну функцію чи інформацію.

3.1.1 Основні сценарії взаємодії

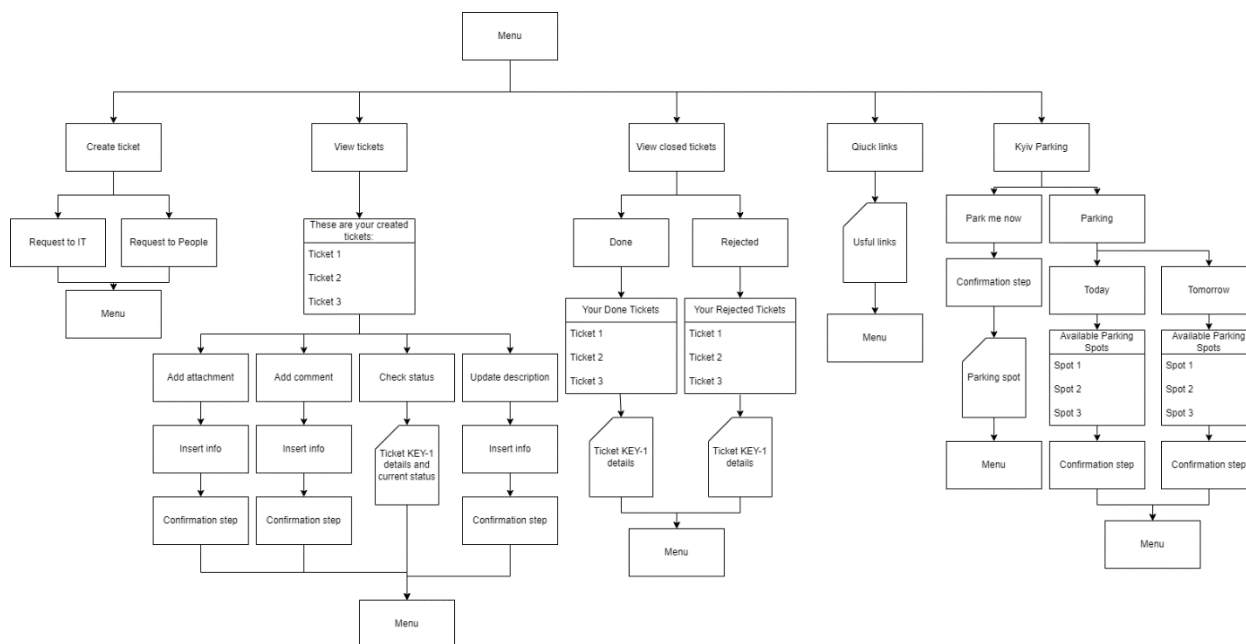


Рис. 3.1.1 Схема дерева діалогів

1. Головне меню

На початковому етапі взаємодії з користувачем надається головне меню, яке містить основні категорії. Серед них можливість створити заявку, яка

поділяється на типи "Request to IT" та "Request to People", можливість переглянути вже створені заявки, включно із завершеними та відхиленими, розділ швидких посилань для оперативного доступу до корисних ресурсів, а також функція "Київ Паркінг" для пошуку доступних паркувальних місць.

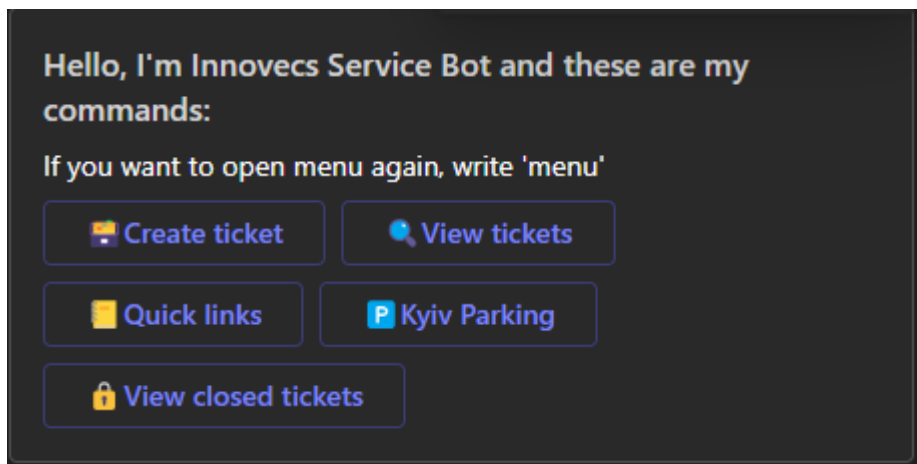


Рис. 3.1.2 Приклад картки головного меню

2. Створення заявки

Під час створення заявки користувач обирає між варіантами "Request to IT" та "Request to People". Після цього він може доповнити заявку, додаючи вкладення, детальний опис або стислий короткий опис. Такий підхід дозволяє більш точно сформулювати суть звернення та надати необхідну інформацію в зручному форматі.

3. Перегляд заявок

У розділі перегляду заявок користувач отримує доступ до власних створених запитів, де може дізнатися про їхній статус, а також вносити оновлення та доповнення. Це може включати додавання вкладень, коментарів та іншу додаткову інформацію, необхідну для подальшого опрацювання звернення. Окремо надається можливість переглядати завершені (Done) та відхилені (Rejected) заявки, де доступні деталі кожної з них, що допомагає користувачеві тримати руку на пульсі всіх процесів, пов'язаних із його зверненнями.

4. Швидкі посилання

Розділ швидких посилань покликаний забезпечити оперативний доступ до корисних ресурсів, таких як інструкції з використання системи, контактні дані важливих відділів та інші матеріали, що можуть знадобитися користувачеві під час роботи з сервісом.

5. Київ Паркінг

Функція "Київ Паркінг" призначена для пошуку доступних паркувальних місць. Користувач має можливість обрати опцію "Park me now" для миттєвого бронювання вільного місця або ж скористатися опцією "Parking" для отримання інформації про доступні місця сьогодні або наступного дня. На основі вибору користувачеві буде надано список доступних паркувальних локацій, що дозволяє легко спланувати поїздку та заощадити час на пошук вільного паркомісця.

Логіка побудови дерева

Кожна гілка дерева діалогів відповідає конкретному запиту користувача, а кожен вузол — це крок у взаємодії.

1. Підтвердження дій

На етапі підтвердження дій користувачеві пропонується вибір:

- **"Так"** — операція завершується, і користувач отримує підтвердження успішного виконання.
- **"Ні"** — бот повертає користувача до головного меню для подальших дій.

2. Команда скасування

На будь-якому етапі діалогу користувач може скористатися кнопкою **"Cancel"** або просто написати команду **"Cancel"** у чаті. У цьому випадку діалог переривається, і бот автоматично повертає користувача до головного меню.

Далі детально розглянемо кожен з цих функцій, їх внутрішні діалоги та особливості реалізації.

3.1.2 Опис кожної функції бота

1. *Create Ticket* (Створити заявку)

Опис функцій:

Функція **Create Ticket** дозволяє користувачам створювати нові заявки до різних відділів компанії через інтерактивний діалог з ботом. Цей модуль інтегрований з системою **Zoho Survey** для зручного збору інформації та подальшої обробки заявок.

Процес створення заявки:

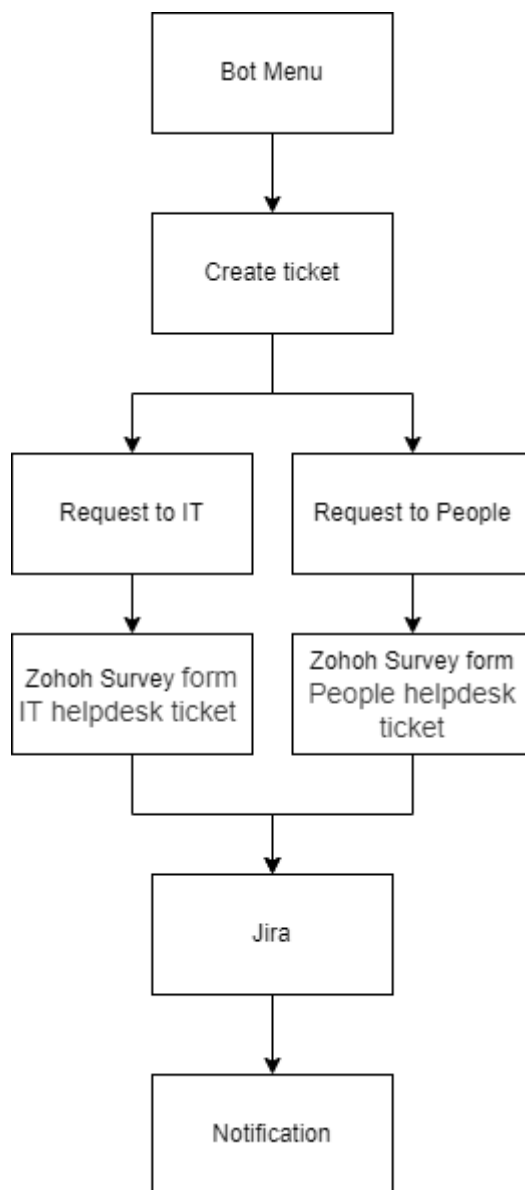


Рис. 3.1.3 Схема процесу створення заяви

1. Вибір відділу:

На початку роботи користувач надсилає будь-яке повідомлення, отримуючи доступ до головного меню, де з-поміж інших функцій доступна можливість створити тикет.

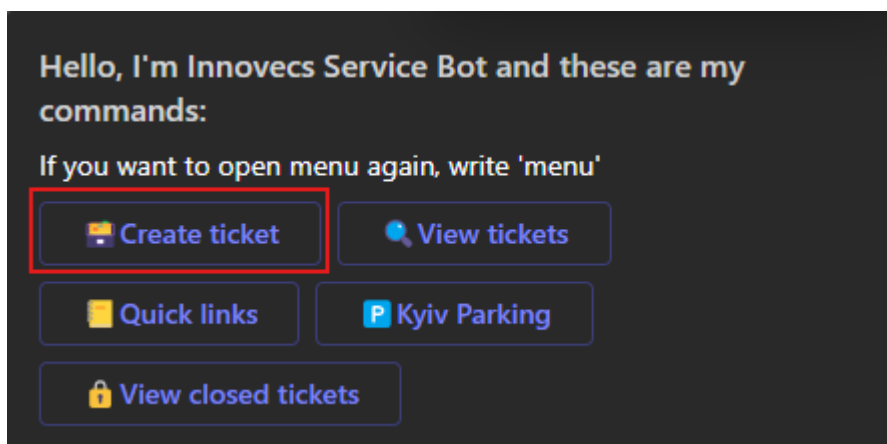


Рис. 3.1.4 Відповідна кнопка в головному меню для створення тикета

Після активації цієї опції система пропонує користувачеві обрати відділ, до якого буде спрямовано заявку. Доступні дві ключові категорії: IT та People. Залежно від обраного варіанту користувач зможе перейти до спеціалізованої форми, адаптованої під потреби конкретного відділу.

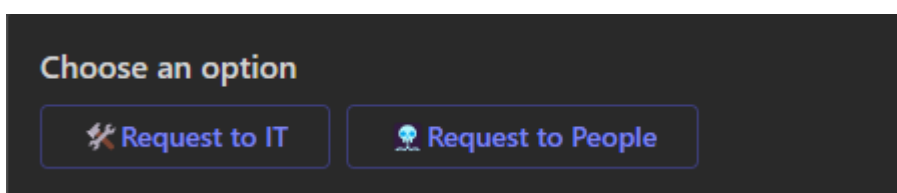


Рис. 3.1.5 Демонстрація картки після вибору функції створення тикета

2. Перенаправлення до відповідної форми:

Після вибору відділу бот перенаправляє користувача на форму, створену в Zoho Survey, яка налаштована саме під потреби цього підрозділу. Відкриття форми відбувається у браузері Microsoft Edge. При цьому посилання генерується динамічно та шифрується так, щоб

користувач не міг змінити свою електронну адресу, з якої було ініційовано запит. Будь-яка спроба модифікувати email у посиланні призведе до втрати доступу до форми. Таке рішення забезпечує безпеку та унеможливорює створення тікетів від імені інших осіб.

Приклад посилання URL:

<https://survey.zohopublic.eu/zs/z0BUAe?reporter=example@mail.com&hash=a5d9c7d7ebc9ccrreter8h4j6l3>

3. Заповнення форми:

Частина форми, яка відповідає за ідентифікацію користувача, заповнюється автоматично, та ховається. З міркувань безпеки, та заборони створити тікет від імені іншої людини, що може бути використано в зловмисних намірах. Ідентифікація проходить за допомогою корпоративної пошти з якої надійшов запит. Бот отримує цю інформацію з Microsoft Teams за допомогою інтегрованих функцій в Microsoft Bot SDK

Користувач заповнює необхідні поля, надаючи детальну інформацію про свою проблему або запит. Форма динамічна і легко може бути змінена під потреби відповідного відділу чи бізнес потреби. Зараз, для створення тікету в Джиру надсилається email автора, короткий опис завдання/назва, детальний опис.

Create IT ticket

Greetings,

Here you can create the request to IT department

* Components

--- Select ---

* Summary

* Details

0 / 20000

Attachment

Choose File

Submit

Рис. 3.1.6 Демонстрація форми створення запиту

В формі надано можливість при бажанні додавати вкладення (файли, зображення) для більш точного опису.

4. Підтвердження та відправка:

Після заповнення обов'язкових полів форми користувач підтверджує відправку заявки. Дані за допомогою мови програмування Zoho продуктів Deludje автоматично передаються до системи **Jira**, де створюється відповідна задача для обробки відділом і розподілення. Якщо в формі були наданні вкладення, то алгоритм трохи замінюється. Спочатку створюється тікет в

джира і Deludje отримує відповідь про створення тікета та його ключ. Ключ разом з вкладенням надсилаються на сервер компанії до коду бота де створено endpoint, бот отримує код та архівоване вкладення і надсилає по коду тікета в джиру. Коли команда побачить тікет, зразу прийде нотифікація користувачу. А чи створився тікет можна подивитися в функції view tickets.

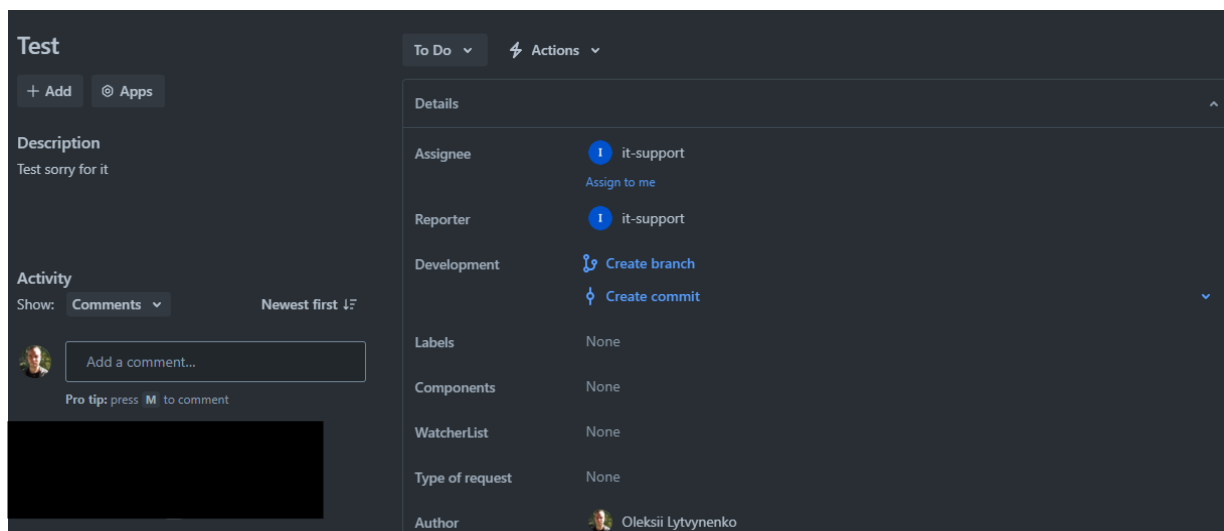


Рис. 3.1.7 Демонстрація створеного тікета в системі Jira

2. View Tickets (Переглянути заявки)

Опис функції:

Функція View Tickets надає користувачу можливість швидко отримати доступ до списку активних заявок, що перебувають у роботі. Це дозволяє відстежувати їх поточний статус, здійснювати взаємодію з існуючими задачами та вносити необхідні зміни.

Дана функція пропонує користувачеві переглядати перелік усіх активних заявок, які він створив. Окрім загального списку, можна вибрати окрему заявку, щоб отримати детальну інформацію, включно з описом, поточним статусом та, в перспективі, іменем виконавця. Користувач має змогу додавати коментарі, прикріплювати додаткові файли або розширювати опис, якщо виникла потреба у уточненнях з боку виконавця чи під час розвитку ситуації.

Внутрішні механізми цієї функції забезпечують зручний діалог: користувач оперативно дізнається про будь-які оновлення у своїх заявках. Якщо статус змінено або додано коментар, користувач отримує відповідне сповіщення. Крім того, при закритті тікета система надсилає посилання на форму зворотного зв'язку, де всі потрібні поля вже заповнені автоматично. Це спрощує процес оцінки якості обслуговування та допомагає системі постійно вдосконалюватися.

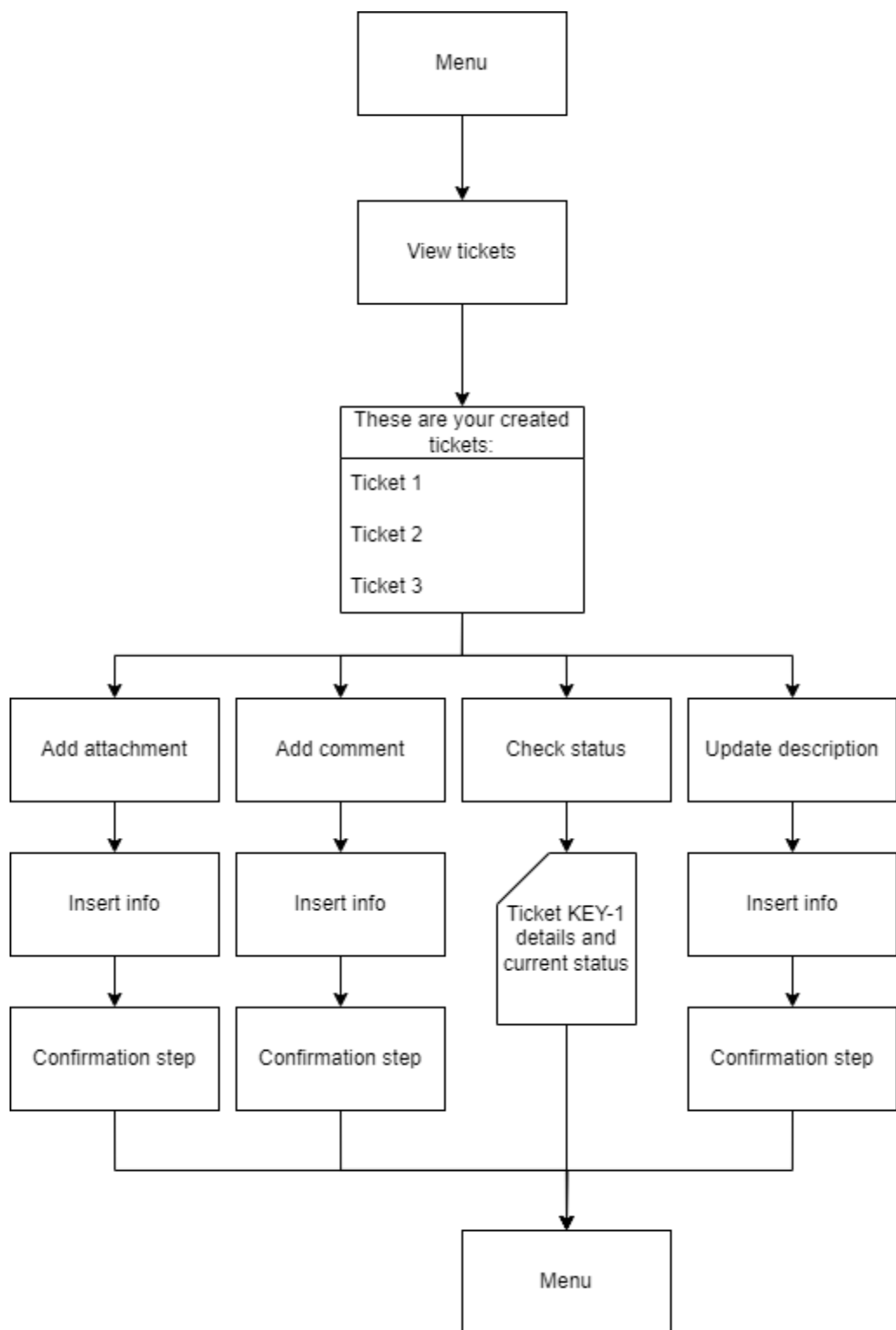


Рис. 3.1.8 Схема роботи функції перегляду запитів

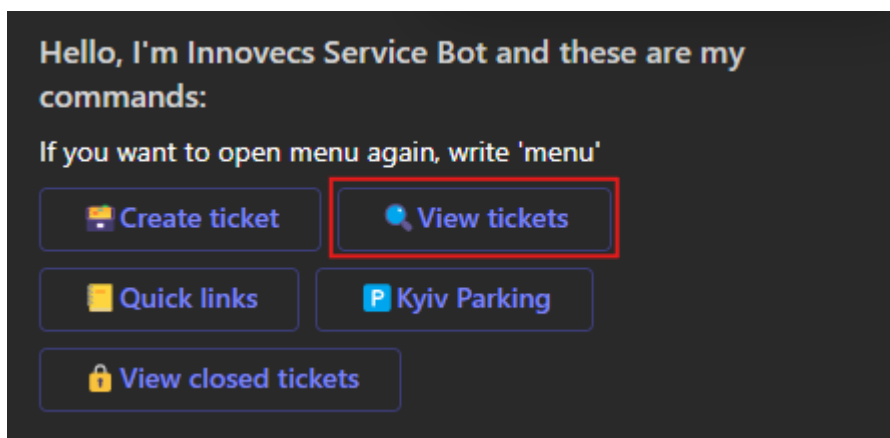


Рис. 3.1.9 Відповідна до функції перегляду запитів кнопка в головному меню

Можливості:

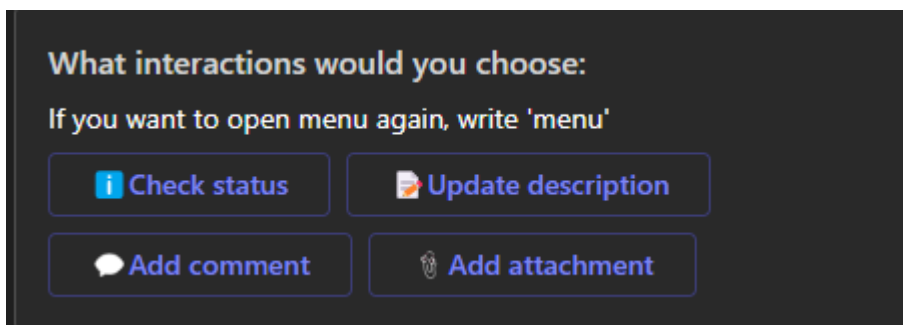


Рис. 3.1.10 Демонстрація картки можливостей після вибору функції перегляду запитів

- **Перегляд списку активних заявок:**

Бот надає список всіх заявок користувача, які знаходяться в роботі.

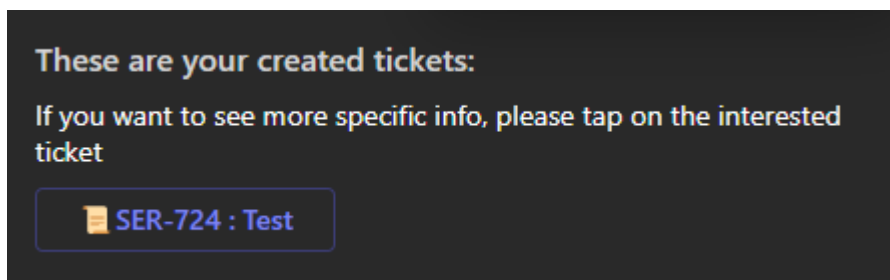


Рис. 3.1.11 Список моїх активних запитів

- **Деталі заявки:**

Можливість отримати детальну інформацію про конкретну заявку, включаючи опис, поточний статус, а в майбутньому ще відповідального виконавця.

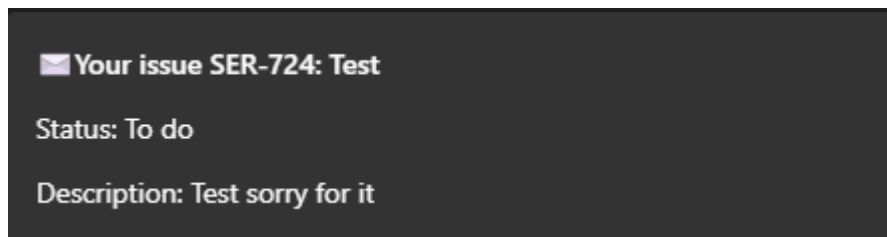


Рис. 3.1.12 Приклад інформації про статус запита

- **Додавання коментарів:**

Користувач може додавати коментарі або додаткову інформацію до своїх заявок.

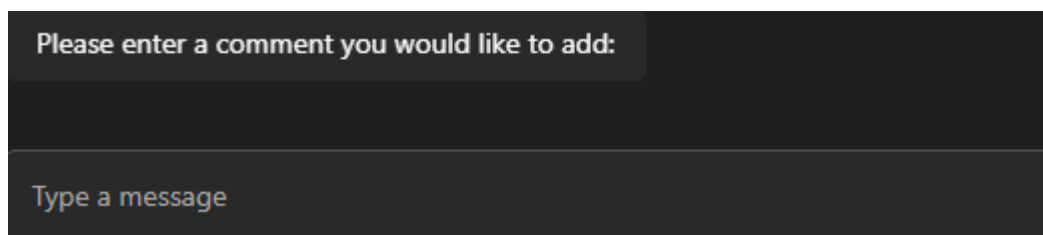


Рис. 3.1.13 Запит на введення коментаря

- **Додавання вкладень:**

Можливість прикріплювати додаткові файли до існуючих заявок.

- **Розширення опису:**

Користувач може доповнити опис своєї заяви, якщо він щось забув або за проханням виконавця .

Внутрішні діалоги та особливості:

- **Сповіщення про оновлення:**

Користувач отримує повідомлення, якщо в його заявці відбулися зміни (наприклад, змінено статус або додано коментар).

- **Можливість надання зворотного зв'язку:**

Користувач інформується про виконання запиту. Разом з повідомленням надходить посилання на форму зворотного зв'язку. Де автоматично заповнена вся необхідна інформація. Наприклад який відділ виконав завдання, код завдання і автора зворотного зв'язку.

3. View Closed (Переглянути закриті заявки)

Опис функції:

Функція View Closed призначена для ознайомлення з історією звернень користувача, які вже завершені. Вона дає можливість переглянути всі закриті заявки, включно з тими, що були успішно вирішені, а також тими, які з певних причин відхилено.

Використання цієї функції допомагає користувачу швидко отримати доступ до архіву своїх звернень, відновити контекст вирішеної проблеми чи проаналізувати, чому певні запити відхилили. У майбутньому планується розширити інформативність даного розділу, надавши доступ до коментарів та імен осіб, які виконували завдання. Навігація по списку закритих заявок організована зручно та зрозуміло, а можливість залишити зворотний зв'язок про якість обслуговування робить систему більш прозорою та орієнтованою на постійне вдосконалення.

.

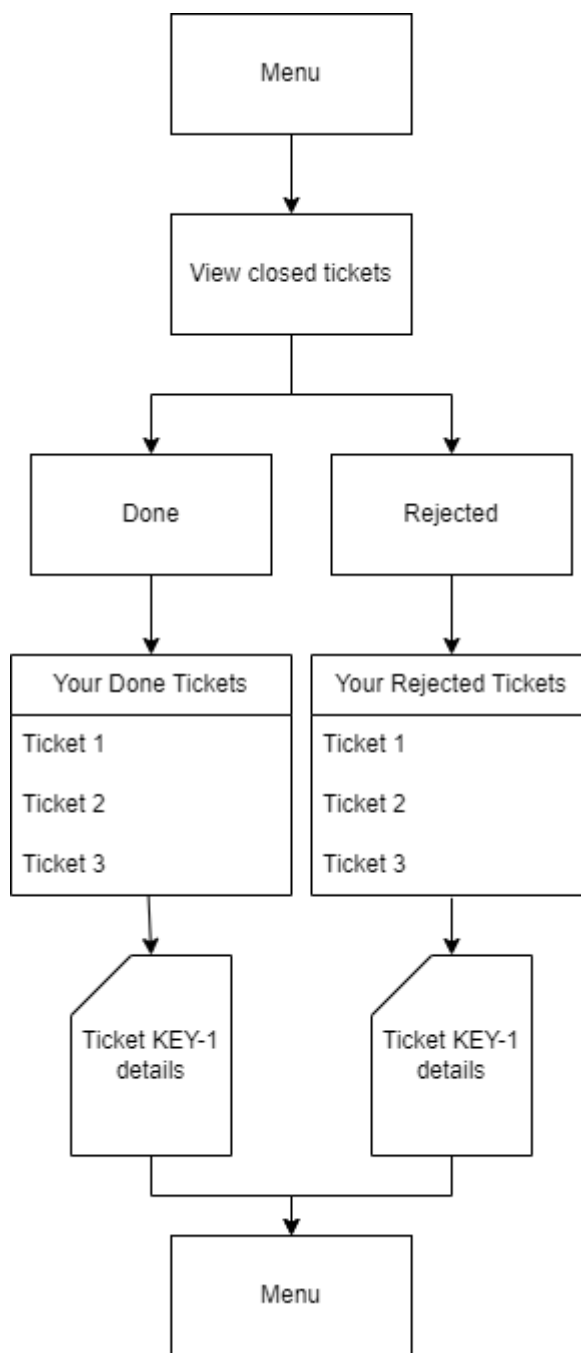


Рис. 3.1.14 Схема роботи функції перегляду закритих запитів

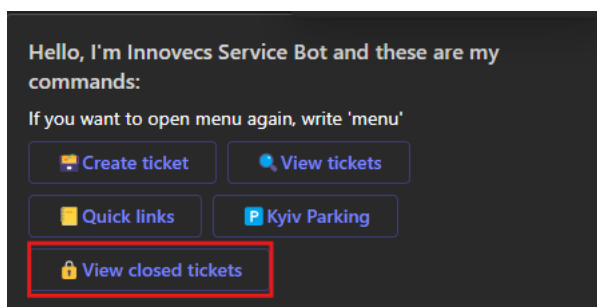


Рис. 3.1.15 Кнопка в меню виклика функції перегляду закритих запитів

Можливості:

- **Перегляд списку виконаних заявок:**

Бот надає список всіх заявок користувача, які були закриті. Та розділяє заяви які були виконані або відхилені. Наприклад за неможливістю їх виконання.

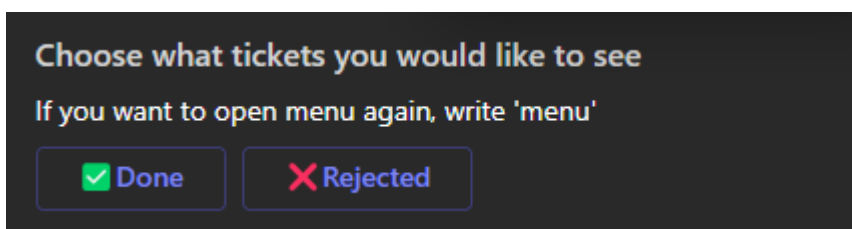


Рис. 3.1.16 Картка після виклику функції перегляду закритих запитів

- **Деталі закритої заявки:**

Можливість отримати повну інформацію про закриту заявку. Планується розширити список того що може бачити користувач, включаючи список коментарів, так того хто виконував завдання.

Внутрішні діалоги та особливості:

- **Навігація по закритих заявках:**

Подібно до функції перегляду активних заявок, бот надає зручні інструменти для навігації.

- **Зворотний зв'язок:**

Користувач може залишити відгук про якість обслуговування або оцінити роботу виконавця.

4. *Kyiv Parking* (Паркування в Києві)

Опис функції:

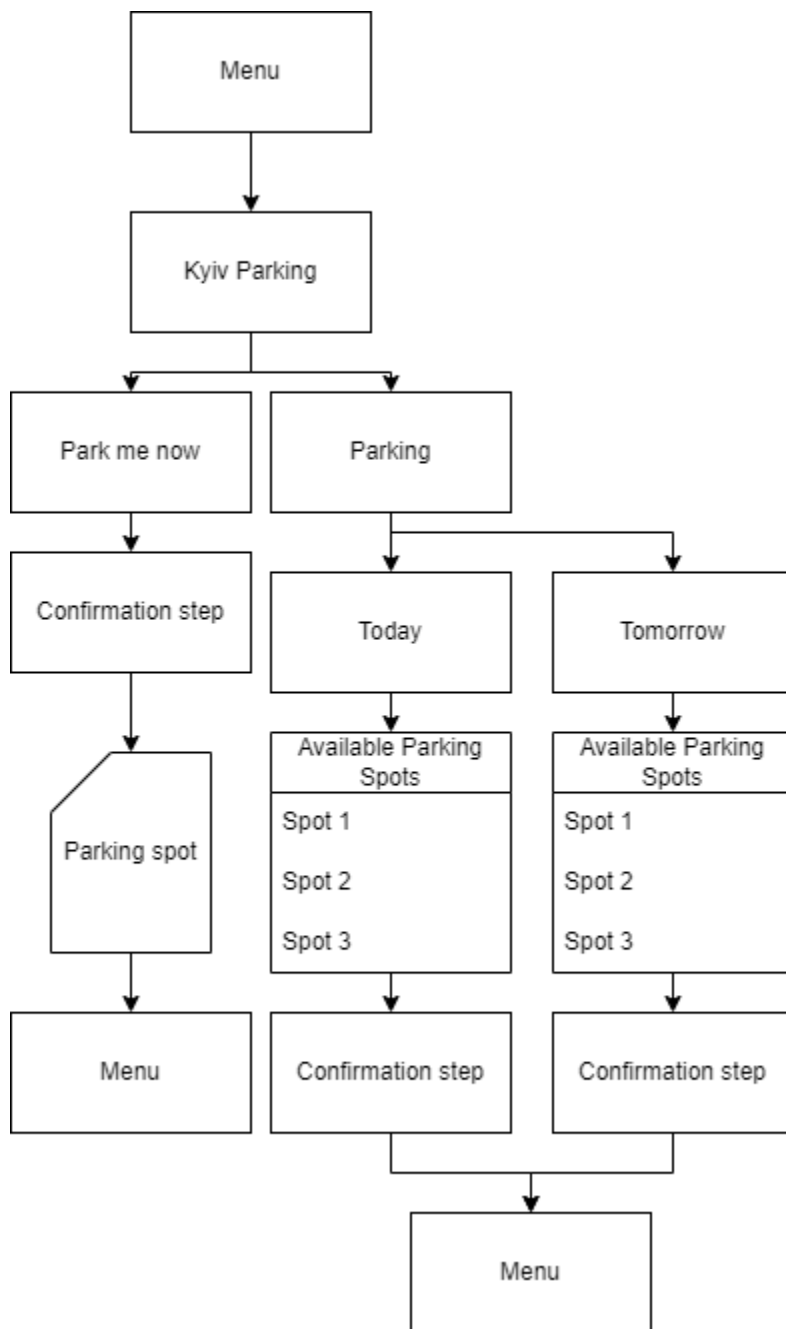


Рис. 3.1.17 Схема роботи функції паркування

Функція Kyiv Parking інтегрує сервіс бронювання паркувальних місць для офісу компанії в Києві, надаючи користувачам інструментарій для простого та оперативного пошуку вільних місць. Завдяки цій опції співробітники можуть дізнатися про наявність паркувальних локацій у визначений час, а також забронювати їх заздалегідь.

Бот пропонує різні сценарії використання. Користувач може виконати швидке паркування, просто обравши відповідну опцію, аби за кілька кліків зайняти вільне місце з пріоритетом на перший поверх. Також доступна можливість перевірити наявність місць на певну дату чи час. Процес бронювання відбувається покроково: бот запитує потрібну інформацію, користувач вносить необхідні дані, а після підтвердження отримує всі деталі свого бронювання. Особливістю є те, що функція призначена для працівників, розташованих в Україні, забезпечуючи персоналізований досвід та полегшуючи логістику в офісі.

Можливості:

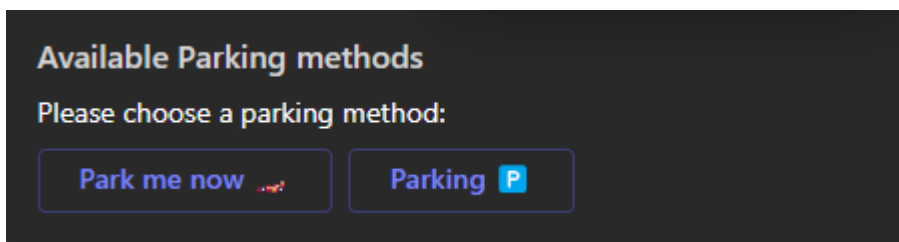


Рис. 3.1.18 Картка вибору методу паркування

- **Перевірка доступності паркувальних місць:**

Користувач може дізнатися про наявність вільних місць на конкретну дату та час.

- **Бронювання місця:**

Можливість забронювати паркувальне місце, вказавши необхідні параметри (дата, час прибуття та від'їзду).

- **Швидке паркування:**

Бот надає можливість в три кліки запакуватися на яке завгодно вільне місце з пріоритетом на перший поверх.

Внутрішні діалоги та особливості:

- **Інтерактивний процес бронювання:**

Бот проводить користувача через послідовність кроків для збору необхідної інформації для бронювання.

- **Підтвердження бронювання:**

Після успішного бронювання користувач отримує підтвердження з деталями.

- **Персоналізована функція:**

Паркування для Київського офісу, є прикладом персоналізованих функцій. Яка доступна тільки для працівників компанії які знаходяться в Україні.

5. *Quick Links (Швидкі посилання)*

Опис функції:

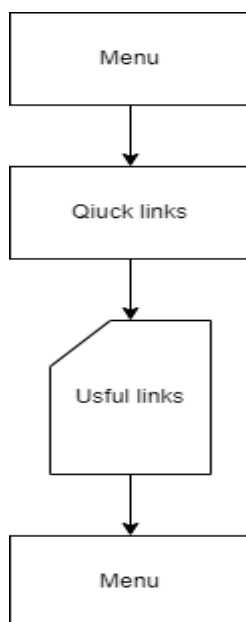


Рис. 3.1.19 Схема роботи функції швидких посилань

Функція Quick Links забезпечує швидкий та безпосередній доступ до внутрішніх ресурсів компанії. Замість витрачання часу на пошук потрібної інформації у різних джерелах, користувачі можуть миттєво перейти до корпоративного порталу, ознайомитися з політиками та процедурами компанії, отримати контактну інформацію відділів чи подати запит на відпустку. Крім того, у цьому розділі можуть бути розміщені й інші корисні ресурси, які регулярно оновлюються та доповнюються.

Завдяки внутрішнім налаштуванням список посилань можна адаптувати залежно від ролі користувача чи конкретного відділу, що робить функцію більш персоналізованою. Адміністратори мають змогу оперативно оновлювати актуальність посилань через конфігураційні файли, забезпечуючи доступ до найсвіжішої інформації та сприяючи підвищенню ефективності внутрішніх комунікацій.

Доступні посилання:

- Корпоративний портал
- Політики та процедури компанії
- Контакти відділів
- Запит на відпустку
- Інші корисні ресурси

3.1.3 Детальне описання архітектури бота

Загальна схема сервісів:

Архітектура чат-бота побудована на інтеграції різних сервісів, які взаємодіють між собою для забезпечення персоналізованого досвіду користувача, управління заявками та автоматизації бізнес-процесів. Нижче описані ключові компоненти та їх роль у системі:

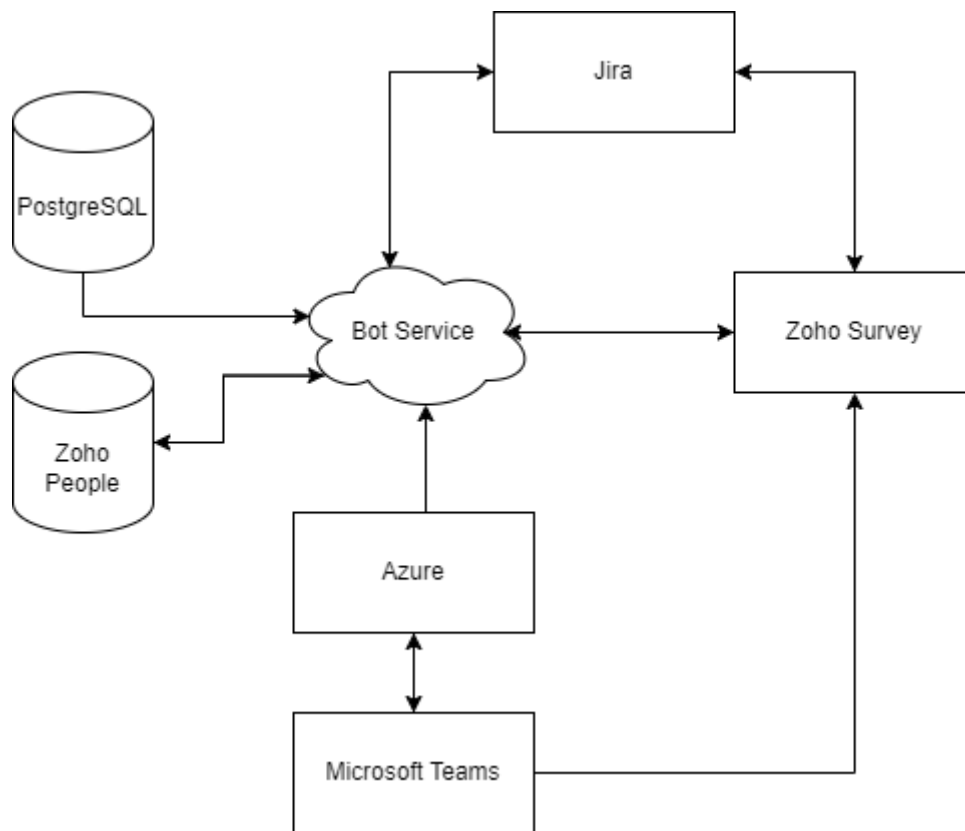


Рис. 3.1.20 Схема взаємодії систем з ботом

Компоненти системи:

1. Microsoft Teams

Microsoft Teams є основним середовищем, у якому користувачі взаємодіють із ботом. Саме тут вони вводять запити, отримують відповіді та повідомлення від бота, а також користуються функціоналом, який надається іншими сервісами, інтегрованими в систему. Teams слугує інтуїтивною точкою доступу до всього функціоналу, пов'язаного з обробкою заявок, отриманням інформації та взаємодією з внутрішньою інфраструктурою.

2. Azure

Microsoft Azure виступає як хмарна платформа для розгортання бота. Вона забезпечує надійне та безпечне середовище для роботи системи, а також спрощує масштабування та управління додатковими ресурсами. Крім того, Azure надає

необхідні інструменти для захисту даних, безперебійної роботи сервісу та інтеграції з іншими компонентами системи.

3. Bot Service

Bot Service є ключовим елементом логіки бота. Він відповідає за обробку вхідних запитів, взаємодію зі сторонніми сервісами та формування відповідей, які надсилаються користувачам. Завдяки цьому компоненту бот може аналізувати введені дані, визначати необхідні дії, а потім звертатися до відповідних сервісів для отримання, обробки або передачі інформації, перетворюючи складні процеси на прості та зручні сценарії для користувачів.

4. PostgreSQL

PostgreSQL у цій системі виконує роль центральної бази даних, яка допомагає реалізувати персоналізацію та інтеграцію з іншими бізнес-системами.

Зберігаючи унікальні ідентифікатори користувачів та інші важливі дані, база даних дає змогу швидко і точно визначати особу користувача, звертатися до його профілю та адаптувати відповіді бота під конкретні потреби. Також це спрощує інтеграцію з Zoho People, оскільки вся необхідна інформація про користувачів доступна з централізованого джерела.

5. Jira

Jira є основною системою для роботи із заявками, створеними за допомогою бота. Вона відповідає за відстеження життєвого циклу звернень – від моменту створення до закриття. Завдяки інтеграції з ботом Jira може надсилати нотифікації про зміну статусу заявок, приймати оновлення, такі як коментарі чи вкладення, а також зберігати історію взаємодії з користувачем. Такий підхід гарантує прозорість у роботі з завданнями та підвищує ефективність внутрішніх процесів компанії.

6. Zoho People

Zoho People інтегровано до бота з метою керування паркувальними місцями.

Адміністратори можуть використовувати цей інструмент, щоб визначати кількість доступних місць та фіксувати інформацію про їх зайнятість, включно з даними про користувача, час та конкретне місце. Дані, отримані з Zoho People, дозволяють боту надавати точну та актуальну інформацію щодо паркування, що спрощує планування та логістику для співробітників.

7. Zoho Survey

Zoho Survey використовується для створення зручних користувацьких форм, через які можна ініціювати різноманітні процеси, зокрема створення заявок у Jira. Після того як користувач заповнює форму, Zoho Survey автоматично передає зібрані дані до Jira, де створюється відповідне завдання. Крім того, ці форми можна ефективно застосовувати для збору зворотного зв'язку та інших видів інформації, динамічно адаптуючи контент форм під вимоги бізнесу та користувачів.

Таким чином, кожен компонент архітектури бота тісно пов'язаний з іншими, створюючи єдину екосистему, де Microsoft Teams слугує інтерфейсом для взаємодії, Azure забезпечує надійний хмарний фундамент, Bot Service керує логікою, PostgreSQL відповідає за персоналізацію, Jira веде управління заявками, Zoho People стежить за доступністю паркувальних місць, а Zoho Survey пропонує гнучкі інструменти для збору та передачі інформації.

Логіка взаємодії між компонентами

1. PostgreSQL як основа персоналізації

На стартових етапах взаємодії з ботом, коли користувач отримує доступ до меню, визначення відображуваних елементів меню ґрунтується на даних із PostgreSQL. Ця база даних зберігає інформацію про користувачів, включно з унікальними ідентифікаторами, які можуть використовуватися для інтеграції з іншими системами. Наприклад, звертаючись до PostgreSQL, бот отримує потрібний ID користувача для подальшого звернення до Zoho People, де буде створено запис про паркування для конкретної особи.

2. Jira як основа для роботи із заявками

Система Jira виступає центральним сховищем і основним інструментом для роботи із заявками. Усі звернення користувачів реєструються в Jira та асоціюються з їхніми корпоративними електронними адресами й кодами завдань. Якщо користувач у боті запитує інформацію про статус заявки, додає коментар чи отримує повідомлення про зміни, усе це відбувається за допомогою інтеграції з Jira. Таким чином, Jira відповідає за відстеження життєвого циклу заявок, інформування про їх статуси, а також оновлення й доповнення завдань на вимогу користувачів через бот.

3. Zoho People для управління паркуванням

Адміністративні процеси, пов'язані з паркувальними місцями, налаштовані в Zoho People. Цей сервіс зберігає конфігурації доступних паркувальних місць та інформацію про вже заброньовані локації. Коли користувач у боті звертається за даними про вільне місце для паркування, бот отримує список доступних позицій і дізнається, які з них зайняті, саме через Zoho People. Це дозволяє актуально та оперативно надавати інформацію про наявні місця.

4. Zoho Survey як інтерфейс для заявок

Zoho Survey забезпечує інтуїтивний і зручний інтерфейс для збору інформації від користувачів шляхом заповнення форм. Після того як користувач заповнює форму й надсилає її, бекенд Zoho Survey автоматично ініціює створення заявки в Jira. Якщо у формі присутні вкладення, логіка обробки даних ускладнюється: спершу створюється заявка, потім дані про вкладення надсилаються для подальшої обробки через додаткові точки взаємодії, щоб у кінцевому результаті всі матеріали опинилися в Jira, асоційовані з конкретним тікетом.

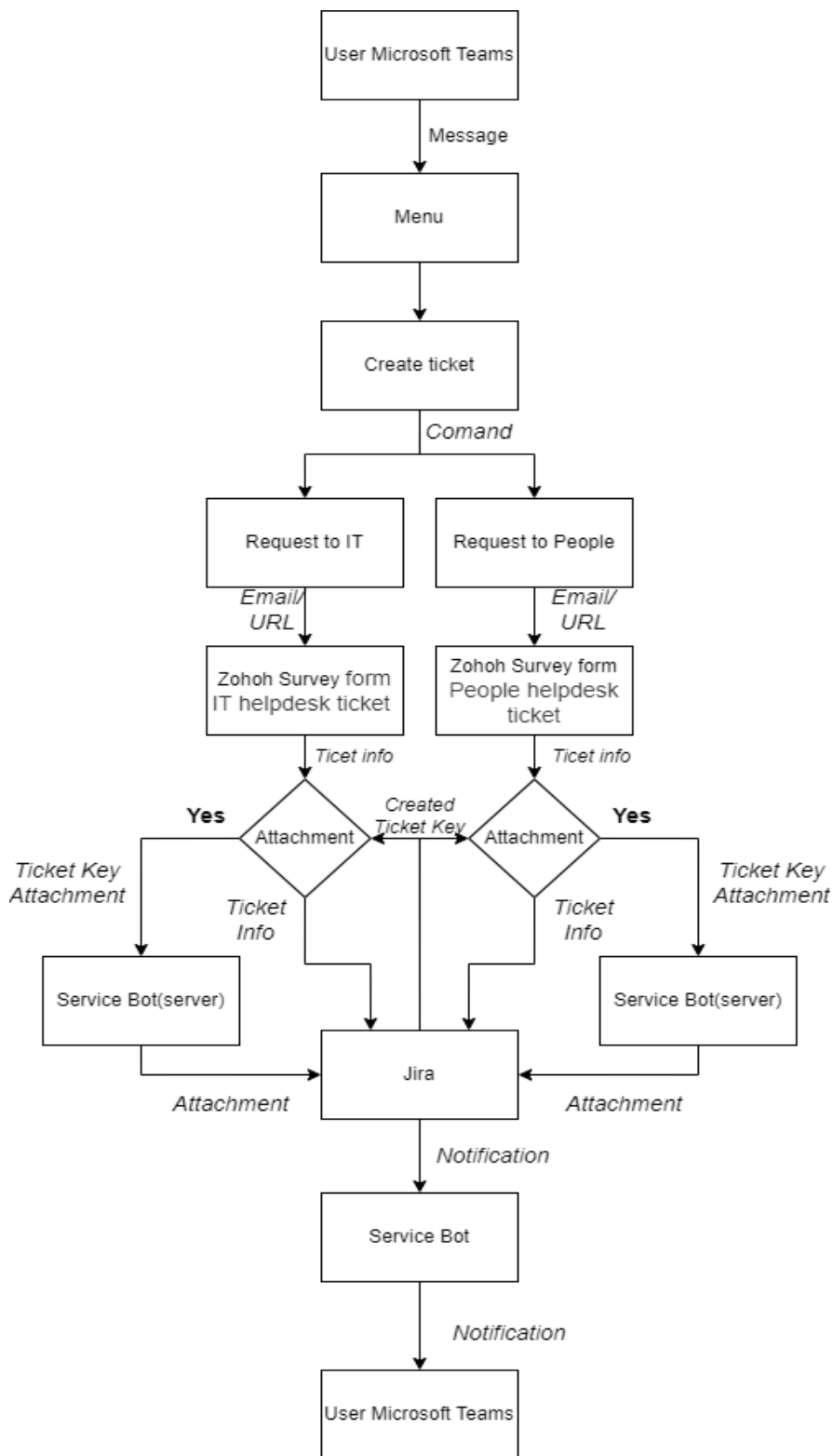


Рис. 3.1.21 Схема – приклад потоку даних при створенні запиту

5. Взаємодія через Bot Service

Bot Service є центральною ланкою, що координує роботу всіх компонентів. Саме він

забезпечує зв'язок між Microsoft Teams, Azure, PostgreSQL, Jira, Zoho People та Zoho Survey. Коли користувач взаємодіє з ботом у Teams, запити передаються до Bot Service, який вже визначає, до якого сервісу звернутися далі, які дані отримати та як відформатувати відповідь для користувача.

6. Потік даних

Основні сценарії взаємодії мають приблизно такі послідовності:

- Персоналізація даних користувачів: Microsoft Teams надсилає запит до Bot Service, який звертається до PostgreSQL за потрібною інформацією, потім повертається до Bot Service й надсилає користувачеві відповіді через Microsoft Teams.
- Управління заявками: Microsoft Teams комунікує з Bot Service, той звертається до Jira для отримання або оновлення статусу заявки, а потім дані повертаються тим же шляхом до користувача.
- Управління паркуванням: Microsoft Teams через Bot Service взаємодіє із Zoho People для отримання актуальної інформації про доступні паркувальні місця.
- Збір інформації через форми: Microsoft Teams передає запит Bot Service, який прямує до Zoho Survey. Після заповнення форми Zoho Survey взаємодіє з Jira для створення відповідної заявки.

3.2. Впровадження бота в компанію

Успішне впровадження чат-бота в корпоративне середовище Microsoft Teams вимагає ретельного планування та дотримання визначених кроків налаштування. Нижче наведено детальний план, що ілюструє, як бот інтегрується в організацію, використовує Azure для розміщення та реєстрації, а також як налаштувати доступ та видимість бота для співробітників.

3.2.1 Інтеграція бота в корпоративне середовище Teams

- **Доступність в межах організації:**

Впроваджений бот доступний для всіх співробітників компанії, які використовують Microsoft Teams у межах корпоративного середовища. Завдяки політиці доступу, що налаштована адміном Teams, бот автоматично додається до списку застосунків (Apps) усіх користувачів організації. Це означає, що працівники не потребують додаткових дій для встановлення або пошуку бота — він вже доступний у їхньому середовищі Teams.

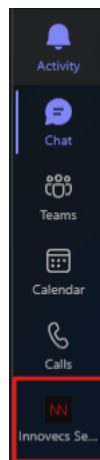


Рис. 3.2.1 Демонстрація способу розповсюдження бота

- **Обмеження зовнішнього доступу:**

Оскільки бот зареєстрований та опублікований у внутрішньому каталозі застосунків організації, зовнішні користувачі (ті, хто не належить до домену або організації) не зможуть його побачити, а тим більше взаємодіяти з ним. Це гарантує, що корпоративні дані та сервісні функції бота захищені від несанкціонованого зовнішнього доступу.

- **Зручний доступ користувачів:**

Після впровадження бота, він може відображатися в панелі навігації Teams, а також у списку чатів. Користувачі можуть безпосередньо звертатися до бота, відкривши його в чаті.

- **Доступ за посиланням:**

Існує можливість надати користувачам пряме посилання на застосунок бота (наприклад, у внутрішньому порталі або розсилці). Однак навіть за наявності посилання лише користувачі з відповідними правами (співробітники компанії) зможуть відкрити та використовувати бота. Або за потреби, знайти його в пошуку застосунків Teams.

3.2.2 Інтеграція з Jira

Інтеграція чат-бота з Jira забезпечує автоматизовану обробку заявок та інформування користувачів про прогрес вирішення їхніх запитів. Це дозволяє оптимізувати внутрішні процеси, підвищити прозорість та оперативність реагування на звернення співробітників.

Детальний опис процесу роботи Jira

1. Створення заявки:

Після того як користувач у чат-боті підтвердить створення заявки (тікета), вона автоматично надходить до відповідної дошки Jira, прив'язаної до відділу, до якого звернувся співробітник. Наприклад, якщо звернення адресоване до ІТ-відділу, тикет з'являється на дошці цього відділу.

2. Обробка тікета оператором:

Оператор, відповідальний за цю дошку (наприклад, у ІТ-відділі), розглядає нову заявку та змінює її статус згідно з внутрішньою процедурою. У випадку ІТ процес найбільш складний. Оператор:

- Перевіряє зміст заявки.
- Визначає рівень підтримки (L1, L2 або L3), необхідний для її вирішення.

При кожній зміні статусу або переході до іншого рівня підтримки тикет може переміщуватися з однієї дошки або колонки Jira до іншої, залежно від внутрішньої структури робочого процесу.

3. Автоматизація та інформування користувача:

За допомогою автоматизованих правил у Jira при кожному переході тікета між статусами користувач отримує нотифікацію (через Teams), що дозволяє йому відстежувати хід вирішення проблеми.

Окрім того, у Jira налаштовано автоматичні коментарі від імені системи або керівників відділів, які пояснюють, що означає певний статус, і які кроки очікуються надалі. Це робить процес прозорим для користувачів: вони розуміють, чому тікет змінив статус, на якому етапі вирішення він зараз та хто відповідає за наступний крок.

4. Рівні підтримки та перерозподіл завдань:

Якщо заявка потребує ескалації на вищий рівень підтримки (наприклад, з L1 до L2 або L3), тікет автоматично переноситься на дошку команди, яка відповідає за цей рівень. При цьому користувач також отримує нотифікацію про зміну статусу, а відповідний персонал — про нове завдання. Таким чином забезпечується чітка вертикаль передачі задач між різними компетентними групами всередині компанії.

Налаштування webhook для нотифікацій в Teams

В процесі інтеграції Jira з Microsoft Teams було впроваджено механізм відправлення нотифікацій за допомогою webhook. Це дало змогу автоматично сповіщати користувачів про зміни статусів тікетів, додавання коментарів або інші суттєві події безпосередньо в Teams.

Основні аспекти налаштування:

1. Визначення кінцевої точки (endpoint):

У Jira було обрано та налаштовано спеціальний endpoint для відправлення HTTP-запитів (POST) з даними про тікети. Ця кінцева точка відповідає надісланому формату повідомлення та передає інформацію до Teams.

2. Динамічний контент нотифікацій:

Вміст повідомлень, що надходять у Teams, налаштований динамічно. Це

означає, що залежно від статусу тікета, його типу, опису та інших атрибутів можна змінювати формат і зміст нотифікацій. Наприклад, при переході тікета на інший рівень підтримки або під час внесення коментаря нотифікація може містити конкретні дані та посилання.

3. Гнучка зміна умов спрацьовування:

За допомогою автоматизації в Jira можливо визначити точні умови, за яких відправлятимуться нотифікації. Це можуть бути:

- Перехід тікета в новий статус.
- Зміна виконавця.
- Додавання коментаря або вкладень.

Дана гнучкість дозволяє адаптувати процес нотифікації до бізнес-процесів організації.

4. Обмеження кодування символів:

У ході експлуатації було виявлено, що Jira надсилає нотифікації у форматі, який коректно відображає лише латинські символи. Використання кирилиці або специфічних символів (наприклад, німецьких "ß", "ä", польських "ł" тощо) призводить до появи знаків питання замість цих символів. Така поведінка ускладнює читання нотифікацій користувачами, що потребує пошуку додаткових рішень щодо налаштувань кодування чи формату передачі даних.

3.3. Тестування та результати впровадження

3.3.1 Порівняння за витратами та функціональністю

Витрати на Jira Help Desk: Модель "As Is"

Система Jira Help Desk є потужним інструментом для автоматизації управління заявками, проте її вартість для середнього підприємства є значною. У таблиці 1 наведено розрахунки витрат на використання Jira Service Management та пов'язаних продуктів у поточному стані.

Таблиця 3.3.1.

Витрати компанії на продукти Atlassian As Is

Продукти та додатки	Кількість користувачів	Наступна оцінка вартості	Платіжний цикл	Розробник
<i>Jira Service Management</i>	26	USD 512.80	Monthly	Atlassian
<i>Jira</i>	159	USD 1,225.05	Monthly	Atlassian
<i>Jira Misc Workflow Extension</i>	159	USD 239.93	Monthly	Atlassian

Загальні витрати у цій моделі складають **1,977.78** доларів США на місяць, або **23,733.36** доларів США на рік.

Витрати на Jira Help Desk: Модель "To Be"

При масштабуванні використання Jira для більшої кількості користувачів витрати значно зростають. У таблиці 2 наведено прогнозовані витрати у цій моделі.

Таблиця 3.3.2.

Витрати компанії на продукти Atlassian To Be

Продукти та додатки	Кількість користувачів	Наступна оцінка вартості	Платіжний цикл	Розробник
<i>Jira Service Management</i>	159	USD 3,135.9	Monthly	Atlassian
<i>Jira</i>	159	USD 1,225.05	Monthly	Atlassian
<i>Jira Misc Workflow Extension</i>	159	USD 239.93	Monthly	Atlassian

Загальні витрати у цій моделі складають **4,600.88** доларів США на місяць, або **55,210.56** доларів США на рік. Це майже у 2.5 рази більше, ніж у моделі "As Is".

Витрати на розробку сервісного бота

Для порівняння, розробка внутрішнього сервісного бота коштувала компанії лише 4,200 доларів США, що включало оплату праці молодших

спеціалістів протягом трьох місяців, якщо їх заробітна плата складає 700 доларів США. Після впровадження бота витрати на підтримку мінімальні, оскільки він працює на існуючій корпоративній інфраструктурі. І може бути легко зміненим відповідно потребам та обставинам бізнесу.

Порівняння витрат

Для кращого розуміння економії витрат наведемо порівняльну таблицю:

Таблиця 3.3.3.

Порівняння річних витрат

Рішення	Кількість користувачів	Річні витрати (USD)
<i>As Is</i>	26	USD 23,733.36
<i>To Be</i>	159	USD 55,210.56
<i>Chat-bot</i>	159	USD 17,579.76

Економія у порівнянні з моделлю "To Be" склала **37,630.80 доларів США на рік**, що дозволяє компанії спрямовувати ці кошти на інші стратегічні завдання.

Використання сервісного бота замість Jira Help Desk забезпечило значну економію коштів. Річні витрати на підтримку системи знизилися майже утричі, а гнучка модульна архітектура бота відкрила можливості для швидкого розширення функціоналу без додаткових ліцензійних витрат. Це рішення є ефективною альтернативою для середніх підприємств, що прагнуть знизити витрати і підвищити ефективність внутрішніх процесів.

3.3.2 Практичні результати

Після впровадження корпоративного чат-бота в Microsoft Teams були досягнуті значні покращення в роботі внутрішніх сервісів компанії.

1. Покращення комунікації та обробки запитів

Як видно з наданих даних, тепер доступна аналітика щодо кількості запитів до HR і IT служб (наприклад, 129 і 253 відповідей відповідно). Це дозволяє краще контролювати завантаженість команд і оптимізувати їхню роботу. Раніше звітність була відсутня, але тепер кожен запит реєструється і відображається в системі, що створює прозорий процес управління.



Рис.3.3.1 Статистикою запитів до People(HR, Finance..) та IT

2. Проактивне сповіщення та статуси

Завдяки проактивним повідомленням у Microsoft Teams співробітники оперативно отримують інформацію про зміни статусу заявок або нові коментарі. Це знижує потребу в додаткових уточненнях і зменшує час на комунікацію. Функція зворотного зв'язку допомагає поліпшити якість сервісів.

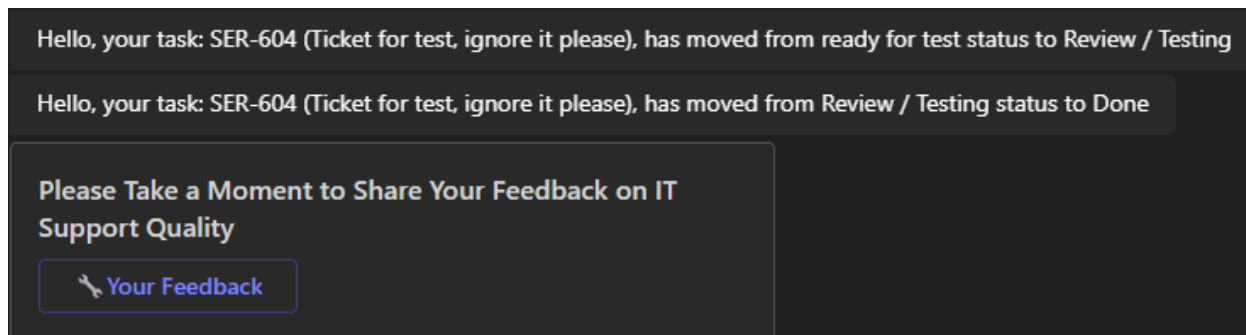


Рис.3.3.2. Приклад про активних повідомлень про зміну статусу тикета

3. Автоматизація управління паркуванням

Раніше процес бронювання паркомісць здійснювався вручну, що створювало незручності. Тепер завдяки інтеграції з Microsoft Teams співробітники можуть бронювати паркомісця одним кліком, як видно з таблиці з даними про бронювання.

Це значно прискорило процес і зменшило ймовірність помилок, які виникали через людський фактор.

Added time ▾	Booking start ▾	Booking end ▾	Modified By ▾	Modified time ▾	is_deleted ▾	Parking place ▾
27-Nov-2024 10:42:50	28-Nov-2024	28-Nov-2024	People	27-Nov-2024 10:42:50	No	112
27-Nov-2024 12:04:11	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 12:04:11	No	109
27-Nov-2024 09:52:30	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 09:52:30	No	112
27-Nov-2024 09:24:55	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 09:24:55	No	110
26-Nov-2024 19:48:52	27-Nov-2024	27-Nov-2024	People	26-Nov-2024 19:48:52	No	68
27-Nov-2024 08:29:51	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 08:29:51	No	111
26-Nov-2024 23:42:08	27-Nov-2024	27-Nov-2024	People	26-Nov-2024 23:42:08	No	67
27-Nov-2024 09:04:33	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 09:04:33	No	108
26-Nov-2024 13:24:43	27-Nov-2024	27-Nov-2024	People	26-Nov-2024 13:24:43	No	66

Рис.3.3.3. Кількість запитів на паркування за один день

4. Єдиний інтерфейс

Інтеграція всіх функцій у Microsoft Teams дозволила створити єдиний інтерфейс для співробітників, де вони можуть виконувати завдання, не перемикаючись між різними платформами. Інтеграція забезпечила єдиний інтерфейс для взаємодії, який доступний усім співробітникам. І використовується більшістю з них.

5. Можливість розширення функцій

Чат-бот побудований на модульній архітектурі, що дозволяє легко додавати нові функції, такі як автоматизація фінансових запитів, підтримка декількох мов або інтеграція з іншими корпоративними інструментами.

3.3.3 Оцінка зручності для кінцевих користувачів

Виклики та вдосконалення

На початковому етапі впровадження сервісного бота виникли кілька проблем:

1. **Спротив до змін.** Користувачі, звиклі до Jira, неохоче переходили на нову систему. Це проявлялося особливо в ситуаціях, коли виникали технічні проблеми, і бот сприймався як менш надійний інструмент. Мінімалістичний інтерфейс та майже його відсутність в боті викликають супротив.
2. **Проблеми з формами опитувань.**

Користувачі зберігали посилання на опитувальники і використовували їх напямую, минаючи бота. Це було вирішено шляхом впровадження шифрування, яке дозволяє відкрити форми лише через бота. Користувачі вводили некоректні email-адреси, що призводило до помилок. Було впроваджено автоматичне заповнення email із профілю Microsoft Teams.

3. **Проблеми з комунікацією.**

Деякі співробітники ігнорували текстові повідомлення. Для вирішення цього бот був вдосконалений: додано емодзі, кольорові тригери та інтерактивні картки для кращого сприйняття. Замість стандартних рекомендованих відповідей у Teams, які не завжди помічали, впроваджено інтерактивні картки для підтвердження дій.

4. **Технічні нюанси.**

У Jira виникали проблеми із відсутністю акаунтів для деяких співробітників, через що заявки не могли бути прив'язані до конкретного автора. Ця проблема вирішувалася за допомогою збереження даних у формі опитування, що дозволяло ідентифікувати автора.

ВИСНОВКИ

У ході дослідження та вирішення головного завдання, а саме оптимізації витрат і консолідації функціональності внутрішніх сервісів, було прийнято рішення розробити модульний чат-бот. Розробка чат-бота виявилася економічно вигіднішою порівняно з використанням комерційних SaaS-рішень, оскільки вимагала лише одноразової інвестиції і часу вже наявних співробітників.

Чат-бот був спроектований відповідно до потреб компанії Innovecs і успішно інтегрований у її існуючу інфраструктуру. Його впровадження дозволило автоматизувати значну частину процесів обробки заявок, знизити витрати на підтримку сервісів і забезпечити можливість подальшого розширення функціоналу.

Попри технічну успішність проекту, чат-бот не був повністю позитивно сприйнятий користувачами через фактори, детально описані у дипломній роботі. Це є важливим аспектом, який потребує вдосконалення у майбутньому.

Подальший розвиток системи передбачає створення додаткового функціоналу, оптимізацію існуючих процесів і покращення користувацького досвіду для забезпечення більшої ефективності та задоволеності співробітників. Бот є перспективним інструментом, який здатен відповідати зростаючим вимогам бізнесу та сприяти підвищенню ефективності внутрішніх сервісів. На мою думку, можна зробити ще один висновок. Організаціям, вигідно використати власні розробки. Бо в результаті їх імплементація, та підтримання буде дешевше ніж, навіть замовляти імплементацію ззовні.

ПЕРЕЛІК ПОСИЛАНЬ

1. McConnell, Campbell R.; et al. (2009). Economics. Principles, Problems and Policies (18th ed.). New York: McGraw-Hill. ISBN 978-0-07-337569-4. Archived from the original (PDF contains full textbook) on 6 October 2016., Glossary, p. G-25.
2. Harrison, Tina; Estelami, Hooman (5 December 2014). *The Routledge Companion to Financial Services Marketing*. Routledge. ISBN 9781134095629.
3. Phillip J. Windley (2002). "Delivering High Availability Services Using a Multi-Tiered Support Model"
4. Nelson K. Y. Leung, Sim Kim Lau (2007). "Information Technology Help Desk Survey: To Identify the Classification of Simple and Routine Enquiries" Journal of Computer Information Systems
5. Chrissy Kidd, Joe Hertvik (2024). "'IT Support Levels Clearly Explained: L1, L2, L3, and More'" Service Management Blog
6. Avery Devereaux (June 6, 2024) "The Crucial Role Of Technical IT Support In Modern Businesses" The Havok Journal
7. "A brief overview of Jira Service Management" (2024) Source visited 10/11/2024 URL: <https://www.atlassian.com/software/jira/service-management/product-guide/overview#jira-service-management>
8. "What is Freshdesk?" (2024) Source visited 10/11/2024 URL: <https://www.freshworks.com/products/what-is-freshdesk>
9. "Service that's up close and personal?" (2024) Source visited 10/11/2024 URL: <https://www.zendesk.com/service>
10. "Workativ Assistant: A Conversational AI & Automation Platform for innovative workplace solutions?" (2024) Source visited 10/11/2024 URL: <https://workativ.com/conversational-ai-platform/what-is-workativ-assistant>
11. Sharma, A. (2020). "The organization of customer support services," European Journal of Marketing, Emerald.

12. Okręglińska, M., Mittal, P., & Navickas, V. (2023). "Exploring the Mechanisms Linking Perceived Organizational Support, Autonomy, Risk Taking, Competitive Aggressiveness and Corporate Sustainability: The Mediating Role of Innovativeness," Sustainability, MDPI.
13. Don Hall (August, 01, 2024). "Disadvantages of Software-as-a-Service (SaaS)," Technologic Advice Journal.
14. Bella Church (5 September 2023) "5 types of chatbot and how to choose the right one for your business." IBM watsonx, IBM Blog
15. Alex Ivankov (February 23, 2023) Journal PROFOLUS "Advantages and Disadvantages of Chatbots"
16. Rossum, Guido Van (20 January 2009). *"The History of Python: A Brief Timeline of Python"*
17. Peterson, Benjamin (20 April 2020). *"Python Insider: Python 2.7.18, the last release of Python 2"*.
18. The Cain Gang Ltd. *"Python Metaclasses: Who? Why? When?"*
19. *"PyDatalog"*. Archived from the original on 13 June 2020. Retrieved 22 July 2012.
20. Peters, Tim (19 August 2004). *"PEP 20 – The Zen of Python"*. Python Enhancement Proposals. Python Software Foundation.
21. Venners, Bill (13 January 2003). *"The Making of Python"*. Artima Developer. Artima.
22. Martelli, Alex; Ravenscroft, Anna; Ascher, David (2005). *Python Cookbook, 2nd Edition*. O'Reilly Media. p. 230.
23. Eric Urban, Jonathan Fingold, Cyril Anderson "What is the Bot Framework SDK?" Learn.Microsoft Article 11/30/2022
24. Warren, Tom (November 2, 2016). *"Microsoft Teams launches to take on Slack in the workplace"*. The Verge. Archived from the original on February 20, 2020.
25. *"COVID impact on meeting apps: Google Meet, Zoom, Microsoft Teams never had it better"*. May 31, 2021.

26. Novet, Jordan (March 27, 2023). "Microsoft says its new version of Teams is twice as fast". *CNBC*.
27. "RealConnect Service Network Communications Explained : Jeff Schertz's Blog". *blog.schertz.name*. March 7, 2019.
28. Rubicomm (March 2, 2022). "Real-time language interpretation add-on now available for Microsoft Teams". *ITWeb*.
29. Hachman, Mark. "New Microsoft Teams AI feature could kill the need for a headset". *PCWorld*.
30. "What does JIRA mean?". *Atlassian.com official website*. Retrieved 5 November 2019.
31. "Customers". *Atlassian.com official website*. Retrieved 2 December 2020.
32. "Jira Overview". *Atlassian.com official website*. Retrieved 20 April 2020.
33. "JIRA RPC Services". *Atlassian.com official website*. 21 October 2011.
34. "Licensing and Pricing". *Atlassian.com official website*. Archived from the original on 14 July 2011.
35. Stonebraker M.; Rowe, L. A. (May 1986). *The design of POSTGRES*. *Proc. 1986 ACM SIGMOD Conference on Management of Data*. Washington, DC. Retrieved December 17, 2011.
36. "PostgreSQL: History". *PostgreSQL Global Development Group*.
37. "What is PostgreSQL?". *PostgreSQL 9.3.0 Documentation*. *PostgreSQL Global Development Group*
38. Abandy, Roosevelt (August 24, 2022). "The History of Microsoft Azure". *Microsoft Tech Community*.
39. Tharakan, Anya George and Dastin, Jeffery (October 20, 2016). "Microsoft shares hit high as cloud business flies above estimates". *Reuters*. Thomson Reuters.
40. "Upcoming Name Change for Windows Azure". *Microsoft Azure*. March 24, 2014.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Державний університет Інформаційно - Комунікаційних технологій

НАВЧАЛЬНО -НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Кваліфікаційна робота на тему:

**«Система сервісної підтримки користувачів на базі модульного чат -
бота в Microsoft Teams»**

Виконав: Литвиненко Олексій

Група: СДАМ-61

Керівник: Нафеев Ровіл к.ф.-м.н., доцент
кафедри інформаційних систем та
технологій

Актуальність теми

- В сучасному бізнес-середовищі ефективність роботи підприємств значною мірою залежить від якості організації внутрішніх сервісів. Технічна підтримка, управління заявками, обробка запитів співробітників та інші сервіси є критично важливими для стабільного функціонування організацій. Без належної підтримки цих процесів функціонування підприємства стає не лише сумнівним, а й практично неможливим, особливо в умовах, коли всі основні бізнес-процеси покладаються на складні технології.
- Для забезпечення ефективної роботи внутрішніх сервісів необхідні спеціальні інструменти. Головними серед них є засоби комунікації, прийняття заявок, їх обробки та відстеження прогресу. Такі інструменти не лише підвищують ефективність роботи технічних та адміністративних служб, але й забезпечують прозорість процесів для кінцевого користувача, надаючи йому змогу розуміти, на якому етапі знаходиться його запит і які дії виконуються. Створення власних автоматизованих рішень для підтримки цих процесів є актуальним завданням для сучасного бізнесу. Особливо в контексті економії ресурсів та можливості самим розвивати свої інструменти виходячи з персональних потреб, специфіки та завдань.

Мета роботи

- **Мета роботи** створення системи - модульного чат-бота для інтеграції в корпоративне середовище Microsoft Teams що дозволяє автоматизувати обробку заявок, підвищити ефективність внутрішніх сервісів, зменшити витрати на підтримку систем і забезпечити зручність користувачів.
- **Об'єкт дослідження** внутрішні сервіси та інструменти для їх підтримки в корпоративних середовищах.
- **Предмет дослідження** процес автоматизації обробки заявок і підтримки внутрішніх сервісів за допомогою чат-бота.

3

Внутрішні сервіси та їх роль

- **Сервіс** за МасКонелом – це дія або послуга, за яку споживач, компанія чи уряд готові заплатити. Прикладами є робота, виконана перукарями, лікарями, юристами, механіками, банками, страховими компаніями тощо. Громадські послуги – це ті, за які платить суспільство (національна держава, фінансовий союз або регіон) у цілому. Використовуючи ресурси, навички, винахідливість і досвід, постачальник послуг приносить користь споживачам послуг. Послуги можна визначити як нематеріальні дії або виконання, завдяки яким постачальник послуг створює цінність для клієнта.

<i>Категорія внутрішнього сервісу</i>	<i>Опис</i>
Технічна підтримка (IT Support)	Розв'язання проблем з обладнанням, ПЗ, підтримка корпоративних систем та обробка технічних заяв.
Фінансові послуги (Financial Services)	Консультації з фінансових питань, управління виплатами зарплат, премій, компенсацій та податків.
HR-сервіси (Human Resources)	Адміністрування страхування, навчання, розвиток персоналу, виплати бонусів, відпусток та кадрові питання
Сервіси управління робочим середовищем (FWS)	Забезпечення комфортного робочого середовища, управління офісними просторами, обладнанням та інфраструктурою
Інші внутрішні послуги	Юридична підтримка, корпоративна комунікація, психологічна підтримка, програми добробуту співробітників

4

Короткий огляд SaaS рішень та їх основні проблеми

Приклад SaaS рішень

Назва сервісу	Короткий опис
Jira Service Management	Це потужна платформа, головним чином призначена для управління IT-послугами та керуванням процесом виконання заяв.
Freshdesk	Це програмне забезпечення для підтримки клієнтів, розроблене для оптимізації процесів служби підтримки та забезпечення ефективного обслуговування
Zendesk	Це широко використовувана платформа для підтримки клієнтів, яка пропонує систему обробки заявок, базу знань та портали самообслуговування
Workativ	Це платформа, яка пропонує інтелектуальних чат-ботів та автоматизацію для підтримки співробітників через самообслуговування

Основні проблеми SaaS рішень

- 1. Високі витрати на ліцензування
- 2. Обмежена функціональність та налаштованість (Customization)
- 3. Повільні інновації та оновлення

5

Проблеми SaaS рішень на прикладі Jira та конкретної команди

Посада	Роль у команді
Інженер даних	Відповідає за обробку, аналіз та управління великими обсягами даних.
Продукт-менеджер	Координує розробку продукту, визначає пріоритети та забезпечує виконання бізнес-цілей.
Розробник Java	Розробляє програмне забезпечення на Java, забезпечуючи функціональність систем
Сервіс-спеціаліст	Конфігурує системи, зокрема HR-системи, для оптимізації внутрішніх процесів.
Менеджер з розробки ПЗ	Керує командами розробників, визначає технічну стратегію та пріоритети
Інженер-програміст (Back-End фінансові системи)	Розробляє серверну частину фінансових систем, забезпечуючи їх стабільність та інтеграцію

Продукти та додатки	Тип ліцензування	Наступна оцінка вартості	Платіжний цикл
Jira Service Management	За користувача	USD 47	Monthly
Jira	За користувача	USD 15	Monthly
Jira Misc Workflow Extension	За користувача	USD 1.5	Monthly

Команда	Витрати в місяць	Витрати в рік
Команда розробки	USD 381	USD 4,572

6

Ідентифікація проблем та формування вимог на прикладі компанії Innovects

1. Високі витрати на ліцензування

- Індивідуальні ліцензії для кожного виконавця
- Необхідність ліцензій для різних відділів
- Обмеженість бюджету команд

2. Обмежений доступ до заявок для виконавців без ліцензій

- Виконавець не може бачити та обробляти заявки
- Необхідність конвертації заявок:
- Втрата прозорості для клієнта

3. Проблеми з інформуванням клієнтів

- Недостатня ефективність електронної пошти
- Необхідність постійного особистого спілкування
- Додаткові часові витрати

1. Створення заявок з розподілом за відділами

- Вибір відділу користувачем(ІТ, People).

2. Нотифікації для користувачів

- Повідомлення про зміну статусу заявки.
- Сповіщення про коментарі виконавців

3. Робота з вкладеннями

- Можливість додавання файлів
- Додавання файлів після створення заявки.

4. Маніпуляції зі створеними заявками

- Додавання коментарів
- Оновлення опису
- Детальний перегляд інформації про заявку

5. Підтримка додаткових функцій

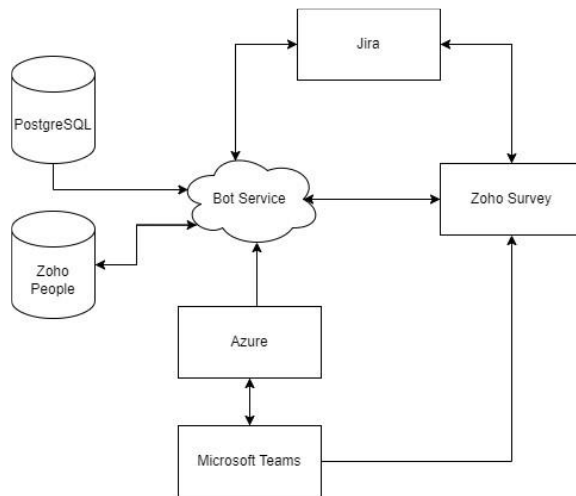
- Інтеграція з системою бронювання паркувальних місць
- Швидкий доступ до важливих ресурсів

6. Уніфікація платформ

- Єдина платформа для взаємодії.
- Можливість розширення функціоналу.

7

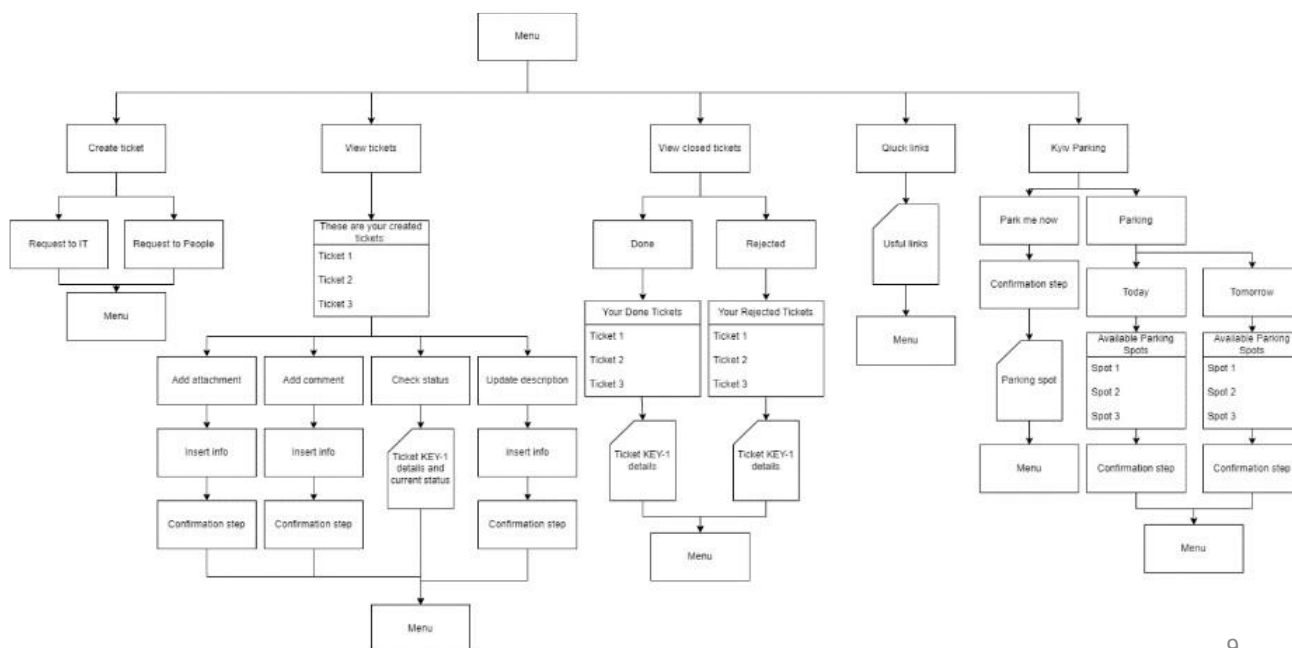
Використані технології та їх взаємодія



ПО	Функція ПО в системі чат-боту
Python	Управління логікою та обробка запитів користувачів.
Microsoft Bot Framework SDK	Забезпечення інтерактивних діалогів та обробки команд користувачів.
Microsoft Teams	Інтерфейс для спілкування користувачів із ботом.
PostgreSQL	Зберігання допоміжної інформації для інтеграції даних в різних системах та для персоналізації
Azure	Розповсюдження та реєстрація в системах Microsoft. Збереження профілів користувачів.
Jira	Створення, оновлення та відстеження заявок користувачів.
Zoho Survey	Збір інформації від користувачів для створення заявок
Zoho People	Бронювання паркувальних місць та інтеграція з внутрішньою інфраструктурою

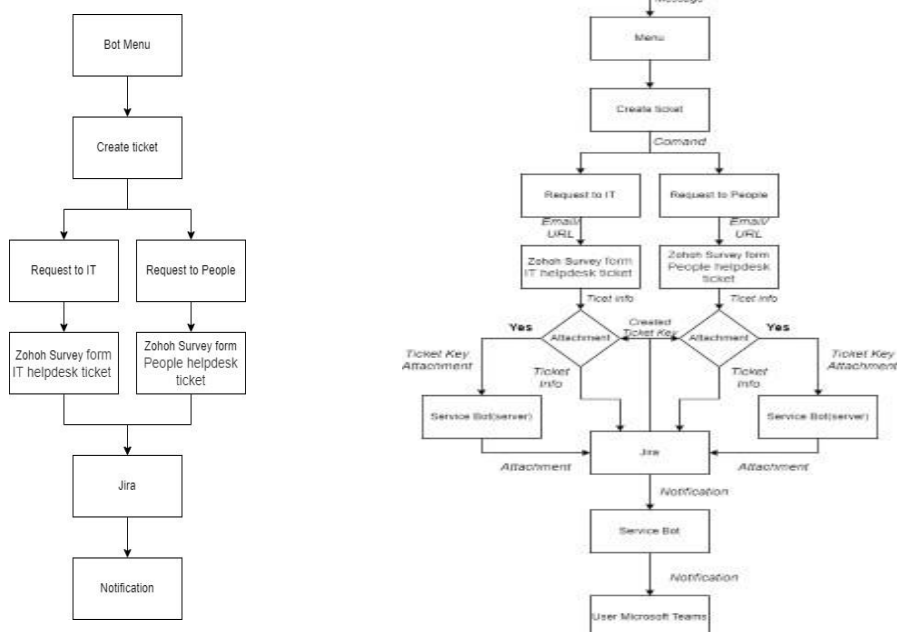
8

Дерево діалогів в чаботі



9

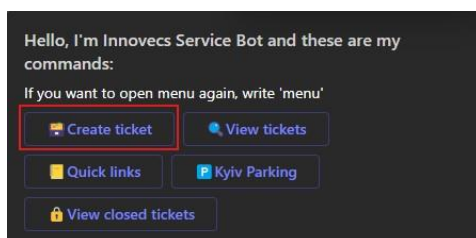
Схема потоку даних на прикладі функції створення заяв



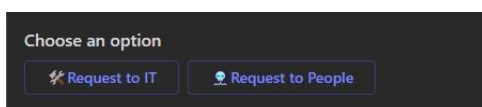
10

Створення заявки

Меню в боті



Меню після вибору кнопки створення заявки



Форма в Zoho Survey

11

Розповсюдження та доступність

- **Доступність в межах організації**
Впроваджений бот доступний для всіх співробітників компанії, які використовують Microsoft Teams у межах корпоративного середовища
- **Обмеження зовнішнього доступу**
Оскільки бот зареєстрований та опублікований у внутрішньому каталозі застосунків організації, зовнішні користувачі (ті, хто не належить до домену або організації) не зможуть його побачити, а тим більше взаємодіяти з ним.
- **Зручний доступ користувачів**
Після впровадження бота, він може відображатися в панелі навігації Teams, а також у списку чатів.
- **Доступ за посиланням**
Існує можливість надати користувачам пряме посилання на застосунок бота (наприклад, у внутрішньому порталі або розсилці).

12

Економічні результати

- У таблиці 1 наведено розрахунки витрат на використання Jira Service Management та пов'язаних продуктів у поточному стані. Загальні витрати у цій моделі складають **1,977.78 доларів США на місяць** або **23,733.36 доларів США на рік**.
- При масштабуванні використання Jira для більшої кількості користувачів витрати значно зростають. У таблиці 2 наведено прогнозовані витрати у цій моделі. Загальні витрати у цій моделі складають **4,600.88 доларів США на місяць**, або **55,210.56 доларів США на рік**. Це майже у 2.5 рази більше, ніж у моделі "As Is".

Витрати компанії на продукти Atlassian As Is

Продукти та доплати	Кількість користувачів	Наступна оцінка вартості	Платіжний цикл	Розробник
<i>Jira Service Management</i>	26	USD 512.80	Monthly	Atlassian
<i>Jira</i>	159	USD 1,225.05	Monthly	Atlassian
<i>Jira Misc Workflow Extension</i>	159	USD 239.93	Monthly	Atlassian

Витрати компанії на продукти Atlassian To Be

Продукти та доплати	Кількість користувачів	Наступна оцінка вартості	Платіжний цикл	Розробник
<i>Jira Service Management</i>	159	USD 3,135.9	Monthly	Atlassian
<i>Jira</i>	159	USD 1,225.05	Monthly	Atlassian
<i>Jira Misc Workflow Extension</i>	159	USD 239.93	Monthly	Atlassian

13

Економічні результати

- Для порівняння, розробка внутрішнього сервісного бота коштувала компанії лише **4,200 доларів США**, що включало оплату праці молодших спеціалістів протягом трьох місяців, якщо їх заробітна плата складає **700 доларів США**. Після впровадження бота витрати на підтримку мінімальні, оскільки він працює на існуючій корпоративній інфраструктурі. І може бути легко зміненим відповідно потребам та обставинам бізнесу.
- Економія у порівнянні з моделлю "To Be" склала **37,630.80 доларів США на рік** та **6,153.8 доларів США на рік** у порівнянні з моделлю "As Is", що дозволяє компанії спрямовувати ці кошти на інші стратегічні завдання.

Рішення	Кількість користувачів	Річні витрати (USD)
<i>As Is</i>	26	USD 23,733.36
<i>To Be</i>	159	USD 55,210.56
<i>Chat-bot</i>	159	USD 17,579.76

14

Практичні результати

Кількість резервувань пакувальних місць за день

Added time	Booking start	Booking end	Modified By	Modified time	Is_deleted	Parking place
27-Nov-2024 10:42:50	28-Nov-2024	28-Nov-2024	People	27-Nov-2024 10:42:50	No	112
27-Nov-2024 12:04:11	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 12:04:11	No	109
27-Nov-2024 09:52:30	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 09:52:30	No	112
27-Nov-2024 09:24:55	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 09:24:55	No	110
26-Nov-2024 19:48:52	27-Nov-2024	27-Nov-2024	People	26-Nov-2024 19:48:52	No	68
27-Nov-2024 08:29:51	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 08:29:51	No	111
26-Nov-2024 23:42:08	27-Nov-2024	27-Nov-2024	People	26-Nov-2024 23:42:08	No	67
27-Nov-2024 09:04:33	27-Nov-2024	27-Nov-2024	People	27-Nov-2024 09:04:33	No	108
26-Nov-2024 13:24:43	27-Nov-2024	27-Nov-2024	People	26-Nov-2024 13:24:43	No	66

Кількість створених тікетів з релізу бота за місяць

 People helpdesk ticket	129
Modified on: Nov 18, 2024 Created on: Jul 08, 2024	Responses
Owned by:   	
 IT helpdesk ticket	253
Modified on: Nov 13, 2024 Created on: Jun 27, 2024	Responses
Owned by:  	

15

Виклики та проблеми

- **Спротив до змін.** Користувачі, звиклі до Jira, неохоче переходили на нову систему. Це проявлялося в особливих ситуаціях, коли виникали технічні проблеми, і бот сприймався як менш надійний інструмент. У додаток, мінімалістичний інтерфейс та майже його відсутність в боті викликають супротив після використання попередньої системи.
- **Проблеми з формами опитувань.** Користувачі зберігали посилання на опитувальники і використовували їх напряму, минаючи бота. Це було вирішено шляхом впровадження шифрування, яке дозволяє відкрити форми лише через бота. Користувачі вводили некоректні email-адреси, що призводило до помилок. Було впроваджено автоматичне заповнення email із профілю Microsoft Teams.
- **Проблеми з комунікацією.** Деякі співробітники ігнорували текстові повідомлення. Для вирішення цього бот був вдосконалений: додано емодзі, кольорові тригери та інтерактивні картки для кращого сприйняття. Замість стандартних рекомендованих відповідей у Teams, які не завжди помічали, впроваджені інтерактивні картки для підтвердження дій.
- **Технічні нюанси.** У Jira виникали проблеми із відсутністю акаунтів для деяких співробітників, через що заявки не могли бути прив'язані до конкретного автора. Ця проблема вирішувалася за допомогою збереження даних у формі опитування, що дозволяло ідентифікувати автора.

Висновки та апробація

- Розробка чат-бота продемонструвала свою економічну ефективність, дозволивши знизити витрати втрічі порівняно з комерційними SaaS-рішеннями. Впровадження системи сприяло автоматизації основних бізнес-процесів, таких як обробка заявок, проактивні повідомлення про зміни статусів і бронювання паркувальних місць. Завдяки модульній архітектурі бот легко адаптується під нові функції та потреби компанії, забезпечуючи масштабованість та гнучкість.
- Єдиний інтерфейс у Microsoft Teams суттєво спростив доступ співробітників до внутрішніх сервісів, покращив комунікацію та зробив взаємодію більш зручною. Водночас впровадження бота виявило низку проблем, таких як необхідність адаптації користувачів до нової системи та доопрацювання деяких технічних деталей.
- У майбутньому планується розширення функціоналу чат-бота для підвищення ефективності внутрішніх сервісів, поліпшення користувацького досвіду та забезпечення відповідності зростаючим вимогам бізнесу.
- Результати дослідження апробовані на VII Всеукраїнська науково-технічна конференція «Комп'ютерні технології: інновації, проблеми, рішення» (02-03 грудня 2024 р.), Житомир, Державний Університет «Житомирська Політехніка».

17

Дякую за увагу!

18