

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу  
(повна назва інституту/факультету)

Кафедра системного проектування  
(повна назва кафедри)

«На правах рукопису»  
УДК 004:004.453

«До захисту допущено»

Завідувач кафедри  
А.І. Петренко  
(підпис) (ініціали, прізвище)

« \_\_\_\_\_ » \_\_\_\_\_ 2018 р.

## Магістерська дисертація

зі спеціальності (спеціалізації) 122-комп'ютерні науки та інформаційні технології (Системне проектування сервісів)  
(код і назва спеціальності)

на тему: CRM як сервіс

Виконав: студент 6 курсу, групи ДА-62м  
(шифр групи)

Остапчук Ян Михайлович  
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник зав. Кафедри, д.т.н. проф. Петренко А.І.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Розробка стартап-проекту д.т.н. проф. Петренко А.І.  
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Нормоконтроль \_\_\_\_\_ к.т.н., доцент, с.н.с. Кисельов Г.Д.  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2018 року

**Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Інститут/факультет «Інститут прикладного системного аналізу»  
(повна назва)

Кафедра \_\_\_\_\_ Системного проектування \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною (освітньо-науковою) програмою

Спеціальність (спеціалізація) 122-комп'ютерні науки та інформаційні технології (Системне проектування сервісів)  
(код і назва)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
А.І. Петренко  
(підпис) (ініціали, прізвище)

« \_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту  
Остапчук Яну Михайловичу  
(прізвище, ім'я, по батькові)**

1. Тема дисертації «CRM як сервіс»  
науковий керівник дисертації \_\_\_\_\_ зав. Кафедри. д.т.н. проф. Петренко А.І.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « \_\_\_ » \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Строк подання студентом дисертації \_\_\_\_\_

3. Об'єкт дослідження Хмара Microsoft Azure

4. Предмет дослідження CRM система як сервіс

5. Перелік завдань, які потрібно розробити \_\_\_\_\_

1. Дослідити існуючі CRM рішення та зробити їх аналіз.
2. Дослідження та тестування існуючих хмарних технологій
3. Використання сучасного інструментарію для прототипування
4. Розробка стартап-проекту.

## 6. Орієнтовний перелік публікацій

1. Остапчук Я.М. Стратегія розвитку CRM систем як сервісу у хмарі Microsoft Azure / Ян Остапчук.// Матеріали XX всеукраїнської науково – практичної студентської конференції, 21 травня 2018, Київ, Україна: матеріали. – К. : НТУУ «КПІ», 2018. – С. 184-185.

## 7. Консультанти розділів дисертації\*

| Розділ                     | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|----------------------------|---|----------------|------------------|
|                            |   | завдання видав | завдання прийняв |
| Реалізація стартап-проекту | Петренко А.І., проф.                      |                |                  |
|                            |   |                |                  |

8. Дата видачі завдання 01.02.2018

## Календарний план

| № з/п | Назва етапів виконання магістерської дисертації                      | Строк виконання етапів магістерської дисертації | Примітка |
|-------|--|---|----------|
| 1     | Отримання завдання   | 01.02.2018                                      |          |
| 2     | Збір інформації та аналіз літератури                                 | 15.02.2018                                      |          |
| 3     | Дослідження хмарних технологій                                       | 28.02.2018                                      |          |
| 4     | Дослідження архітектури CRM систем                                   | 11.03.2018                                      |          |
| 5     | Огляд програмних інструментів для роботи з сервісами та їх побудовую | 13.04.2018                                      |          |
| 6     | Створення тестової платформи   | 25.04.2018                                      |          |
| 7     | Оформлення дипломної роботи  | 19.05.2018                                      |          |
| 8     | Отримання допуску до захисту та подача роботи в ДЕК                  |   |          |

Студент

\_\_\_\_\_ (підпис)

Остапчук Я.М.

(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_ (підпис)

Петренко А.І.

(ініціали, прізвище)

\* Консультантом не може бути зазначено наукового керівника

## РЕФЕРАТ НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконану на тему: ” CRM як сервіс”

студентом: Остапчуком Яном Михайловичем

Робота виконана на 87 сторінках, містить 33 ілюстрацій, 23 таблиць. При підготовці використовувалась література з 20 джерел.

### **Актуальність теми**

В наш час кожен на ринку досить висока та при ситуації взаємодії з клієнтами кожна компанія замислюється про впровадження CRM систем, а саме використання хмарних CRM, які доступні не тільки великим компаніям та й малим підприємцям.

Теперешні хмарні CRM рішення займають вже третину ринку та ростуть з великою швидкістю та до кінця 2020 року, хмарні рішення будуть займати половину ринку.

Перенесення CRM у хмари дозволяє зробити систему досить мобільною та ефективною. Особливість хмарного забезпечення в тому, що він з'являється, як альтернатива корпоративного ПО в середовищі працівників компанії, а не насаджується «зверху» як раніше.

Особливий інтерес представляє дослідження нових інноваційних технологій, які застосовуються у діючих CRM рішеннях.

### **Мета та задачі дослідження**

Метою даної роботи є аналіз діючих CRM рішень у хмарі, які дозволяють підвищити рівень маркетингу, продажів, комерції, обслуговування клієнтів і соціальної взаємодії між співробітниками компанії і клієнтами.

### **Рішення поставлених завдань та досягнуті результати**

У даній роботі було представлено прототип CRM системи для показу можливостей, які надаються хмарним провайдером, реалізація базових сервісів та їх впровадження. Розглянуто методи збереження даних у хмарі та показано практичні приклади роботи з ними. Були застосовані та протестовані сервіси, які надаються FiWare завдяки, яким стало можливим перехід від

монолітної архітектури до мікросервісної. Було протестовано робочий стенд, та сервісі, які взаємодіють з ними та розгорнуто його у хмарі на декількох віртуальних машинах із застосування Load Balancer.

### **Об'єкт досліджень**

CRM система як сервіс

### **Предмет досліджень**

Розгляд сучасних CRM рішень зокрема їх окремих сервісів та можливостей у цілому

### **Методи досліджень**

Для вирішення проблеми в даній роботі використовуються нові методи збереження даних та їх реалізація, методи аналізу і синтезу.

### **Наукова новизна**

Наукова новизна роботи полягає у розгляді сучасних CRM рішень, аналіз їх сервісів та інноваційних технологій, які вони застосовують для вирішення задачі підвищення рівнів маркетингу та продажів, а також аналіз впровадження нових інноваційних рішень в галузі CRM на базі сучасних технологій.

### **Практичне значення одержаних результатів**

Отримані результати можуть використовуватись торговими компаніями за напрямком покращення діючих CRM систем та впровадження в них кращих технологій збереження дани та взаємодії

### **Публікації**

1. Остапчук Я.М. Стратегія розвитку CRM систем як сервісу у хмарі Microsoft Azure / Ян Остапчук.// Матеріали XX всеукраїнської науково – практичної студентської конференції, 21 травня 2018, Київ, Україна: матеріали. – К. : НТУУ «КПІ», 2018. – С. 184-185.

### **Ключові слова**

CRM-система, Microsoft Azure, Azure Storage, Azure service bus , REST , SOAP, JSON , опис сервісів

## **ABSTRACT ON MASTER'**

on topic:" CRM as a service"

by: Ostapchuk Yan Mikhailovich

The thesis contains 87 pages, 44 figures, 23 tables, 20 references

### **Relevance**

Nowadays, the cocuration on the market is quite high and with the situation of customer interaction, each company is thinking about the implementation of CRM systems, namely the use of cloud-based CRM, which are available not only to large companies and small businesses.

Current cloud CRM solutions occupy a third of the market and grow at high speeds and by the end of 2020, cloud-based solutions will occupy half of the market.

Moving CRM to the cloud allows the system to be mobile and efficient. the promise of cloud-based security is that it appears as an alternative to corporate software among employees of the company, rather than being tucked "from above" as before.

Of particular interest is the study of new innovative technologies that are used in CRM decision makers.

### **Purpose**

The purpose of this work is to analyze the possibilities of placing crm system in the cloud and introduction of new technologies for data storage and interoperability between services.

### **Results**

In this paper a prototype CRM system was presented to show the opportunities provided by the cloud provider, the implementation of basic services and implementation. Methods of storing data in the cloud are considered and practical examples of work with them are shown.

### **Object of research**

CRM as a service

### **Object of research**

Routine of modern CRM solutions, in particular their individual services and capabilities in general

### **Research methods**

To solve the problem in this paper we use new methods of data storage and their implementation

### **Scientific novelty**

The scientific novelty of the work is the creation of a CRM system in the cloud, and the consideration of new innovative methods of data storage and communication between services, as well as the possibility of introducing new innovative solutions in the field of CRM based on modern technology.

### **Practical value**

The results obtained can be used by research developers to improve existing CRM systems and to implement the best technologies for the conservation of data and interaction.

### **Publications**

1. Ostapchuk Y.M. Strategy for the development of CRM systems for servicing the Microsoft Azure / Yan Ostapchuk // Materials of the XX All-Ukrainian Scientific and Practical Student Conference, 21 herbs 2018, Kyiv, Ukraine: material. - K.: NTUU "KPI", 2018. - P. 184-185.

### **Keywords**

CRM-system, Microsoft Azure, Azure Storage, Azure service bus , REST , SOAP, JSON , description of services

## СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

- API – application programming interface, прикладний програмний інтерфейс
- CRM- customer relationship management , система взаємовідносин з клієнтами
- JSON - JavaScript Object Notation, запис об'єктів Java Script
- REST – Representational State Transfer, передача репрезентативного стану
- HTTP - Hyper Text Transfer Protocol, протокол передачі гіпертексту
- URI – Universal Resource Identifier, унікальний ідифікатор ресурсу
- SOAP – Simple object access protocol, простий протокол доступу даних
- PaaS – Platform as a service, платформа як сервіс



|  |    |
|--|----|
| Зміст  |    |
| ВСТУП .....  | 11 |
| 1 Розгляд сучасних CRM рішень та розгляд їх сервісів.....              | 11 |
| 1.1 Розгляд CRM системи SugarCRM від Integros .....                    | 12 |
| 1.2 Розгляд CRM системи AmoCRM .....                                   | 13 |
| 1.3 Розгляд CRM системи Битрикс24 .....                                | 15 |
| 1.4 Розгляд CRM системи Microsoft Dynamics CRM .....                   | 18 |
| 1.5 Розгляд CRM системи Microsoft Terrasoft.....                       | 20 |
| 1.6 Порівняння CRM рішень.....   | 21 |
| 1.7 Висновки.....  | 22 |
| 2 Аналіз та узагальнення сучасних CRM .....                            | 23 |
| 2.1 Висновок.....  | 26 |
| 3 Функціональність базових сервісів.....                               | 26 |
| 3.1 Висновок .....   | 27 |
| 4 Пошук та вибір необхідних сервісів .....                             | 27 |
| 4.1 Висновок.....  | 29 |
| 5 Інструментарій для прототипування сучасних сервісних додатків .....  | 29 |
| 5.1 Обґрунтування вибору хмарного провайдера та їх огляд.....          | 30 |
| 5.2 Microsoft Azure PaaS.....  | 34 |
| 5.3 Створення основи стенду завдяки Microsoft Cloud Services .....     | 36 |
| 5.4 Використання Azure Load Balancer для прототипування .....          | 43 |
| 5.5 Використання шини для прототипування стенду Azure Service Bus..... | 44 |
| 5.6 Збереження даних в Microsoft SQL Azure .....                       | 46 |
| 5.7 Збереження даних в Microsoft Azure Cosmos DB .....                 | 52 |
| 5.8 Секціонування в Azure Cosmos DB.....                               | 54 |
| 5.9 Робота з Azure Cosmos DB.....                                      | 55 |
| 5.10 Розробка власного сервісу з використання Storage Blobs.....       | 58 |
| 5.10.1 Опис можливостей Azure Storage у цілому.....                    | 58 |
| 5.10.2 Опис можливостей Azure Blobs .....                              | 59 |
| 5.10.3 Реалізація на мові С# і технології ASP.NET MVC.....             | 61 |
| 5.11 Висновок.....   | 64 |

|   |    |
|---|----|
|   | 10 |
| 6. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ “CRM як сервіс” .....          | 65 |
| 6.1 Опис ідеї стартапу .....                                  | 65 |
| 6.2 Технологічний аудит ідеї проекту .....                    | 68 |
| 6.3. Аналіз ринкових можливостей запуску стартап-проекту..... | 68 |
| 6.4 Розроблення ринкової стратегії проекту .....              | 77 |
| 6.5 Розробка маркетингової програми .....                     | 80 |
| 6.6 Висновок.....   | 84 |
| Висновки.....   | 85 |
| ПЕРЕЛІК ПОСИЛАНЬ.....   | 86 |

## **ВСТУП**

Сьогодні в часи жорсткої конкуренції, торгові компанії почали зазнавати втрат збитки та щоб цей процес не продовжувався далі, компанії стали прислухатися до своїх клієнтів та ще шукати нових. І щоб реалізувати ці дії, почали застосовувати системи взаємодії з клієнтом. Також треба враховувати, що більша частина новітніх CRM провайдерів використовують хмарні рішення. Вже не є ефективно використовувати локальні CRM та, як це вже новий виток. При розгляді CRM високої уваги потребують інноваційні сервіси та технології для побудови CRM.

Особливий інтерес представляє огляд нових методів збереження даних, проте кожні дані потребують особливого збереження залежно від їх розміру та класифікації. Також треба враховувати масштабування цих даних.

І головне правильно обрати архітектуру для сервісу та взаємодії між сервісами.

### **1 Розгляд сучасних CRM рішень та розгляд їх сервісів**

На сьогодні із за високої конкурентності на ринку, компанії почали приділяти велику увагу управлінню взаємовідносинами за клієнтом. Зокрема було поставлено головні задачі, це не втратити теперішніх клієнтів та зацікавити нових споживачів для розширення клієнтської бази. Тому у наш час обійтись без CRM- систем досить важко, бо вони дають суттєву перевагу порівняно з введенням справ через записники чи таблиці excel. Проте зараз використовуються нові CRM, які розміщені у хмарі, що є новим витком у сфері взаємодії з клієнтом.

Сьогодні хмарні технології займають третину ринку CRM систем та це відношення буде рости. Головною перевагою хмарних рішень є те що

вони не перестають рости і CRM системи стають доступними простим користувачам і не директорам та бухгалтерам великих компаній, а простим підприємцям для яких важливе автоматизування їх взаємовідносин з клієнтами. Особливість хмарних CRM є те що вони являють собою альтернативу корпоративним програмним забезпеченням.

Для початку розглянемо основні критерії на, які будемо спиратися при виборі CRM систем. Перше нащо треба звернути увагу ,це кількість користувачів , яка може бути для одного бізнесу. Друге це інтерфейс, який має бути інтуїтивно зрозумілим та водночас інформативним. Третє на, що треба звернути увагу , це наявність сервісів для введення зустрічей, спілкування між клієнтами, телефоні розмови , логістика та інше.

Велику увагу треба приділити таким факторам, як тестовий період , щомісячна плата, багатоплатформність, інтеграція з зовнішніми ресурсами, імпорт даних, аналітика, онлайн-допомога у реальному часі, пошук та маркетинг.

### **1.1 Розгляд CRM системи SugarCRM від Integros**

SugarCRM представляє собою CRM систему написану на PHP та застосовує базу даних MySQL. CRM система не застосовує новітніх інноваційних технологій з боку розробки проте реалізовує досить багато сервісів для планування продажів, автоматизації маркетингу, лід менеджменту, автоматизації продажів. Серед її клієнтів є такі компанії, як Coca-Cola, IBM, Chevrolet. CRM надає підтримку для компаній розміром від 1-9 працівників до понад 2000 працівників. Має мобільні клієнти на платформі Android та IOS , що дає деякі переваги та зручність для клієнтів.

Після реєстрації приходить лист на пошту, що показує наявність базового сервісу підтвердження через email. Після реєстрації доступ надається через не одразу , а приходить лист на пошту, що ви будете розглянуті та підтвердження. При заході на сайт присутні базові функції такі

, як інформація про контрагенти. Інтерфейс є простим та інтуїтивно зрозумілим схожим на Microsoft Office 98. Є сервіс продажу, маркетингу та підтримки, що являється плюсом. Також присутні такі сервіси, як контакти, угоди.

CRM система дає змогу управляти продажами, вона складається зі стадій таких, як планування маркетингу, попередній контакт тобто чи цікава ваша продукція. потім переговори та угоди. Вся система розділяється на модулі та під модулі. Головними модулями є заходи, контакти, контрагенти попередні контакти, угоди, проекти, звернення помилки, документи, e-mail та маркетинг. Якихось інноваційних підходів у даній CRM не було знайдено, проте вона є досить зручною, легкою в управлінні та простими словами працює злагоджено, як годинник.

## **1.2 Розгляд CRM системи AmoCRM**

Дане CRM- рішення розроблялося бізнесменами з проекту “бізнес-молодість” та по їх словам вона не являється досить інноваційною в плані розробки, проте являється з боку користувача. Бо головною їх ідеєю є сервіс спілкування з клієнтом, тобто кожен оператор підключений до CRM системи робить виклик до потенційного клієнта з контрагентів. Та питає у клієнта чи цікава йому та чи інша послуга де потім в залежності від вибору клієнта буде проаналізована дана розмова фахівцями та проаналізована.

AmoCRM складається з новітнього та функціонального інтерфейсу (рис 1.1)

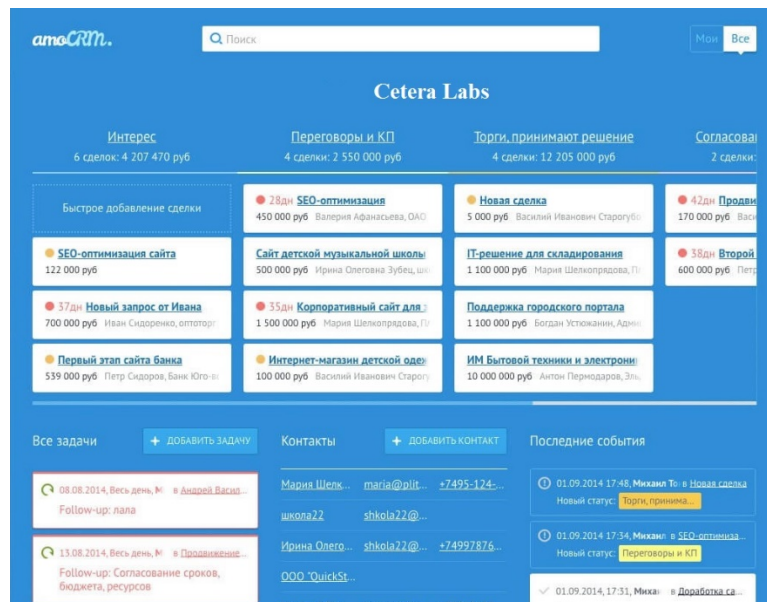


Рисунок 1.1- Интерфейс АмоCRM

Дана CRM є хмарною, складається з таких головних модулів, як робочий стіл, угоди, задачі, списки, пошта, аналітика та налаштування. На робочому столі можна побачити такі підрозділи, як продажі, холодний виклик, партнери та партнерка. Є панель профілю та його редагування, де можна налаштувати свій профіль та змінити пароль. На робочому столі також зображена вся статистика по задачах також показано всі останні івенти, панель пошуку по сайту.

У розділу угоди є такі підрозділи, як продажі, холодний виклик, партнери та партнерка, як на робочому столі. В розділі показані угоди у карточок реального часу, карточка початку взяття номеру та й до заключення угоди. На карточці показано ім'я, статус угоди її час, та сума угоди. Всі угоди розділяються на класи такі, як новий лід, взяли до роботи, класифіковані, оприлюднення цін, Skype перемов, прийнято до роботи, зроблена, акт підписаний, доплата отримана, відмова, угода закрита. Також в процесі роботи можна видалити угоду. Також по угодах є досить багато параметрів фільтрування від часу і до якості угоди, досить схоже на фільтри в excel таблицях. Можна обрати карточку, яка показує всю історія роботи з клієнтом, є повний цикл питань, показано всі архіви розмов, які можна прослухати та список заповнених питань, треба заповнити обов'язкові поля.

Є панель віджети для відправки смс та електронної пошти, а також кастомні віджети. Можна обрати всіх менеджерів, подивитися їх статистику та список угод. Є правила, що уся інформація повинна бути в CRM, у кожній угоді повинна бути задача, завжди перетягувати угоду. У розділі списки є окремі підрозділи по параметрам задач. Є списки по контактам, компаніям, ліцензіях та список товарів та послуг. У списку контактів показане повне налаштування по контактам, загалом показані усіх контактів.

У розділі пошта показані усі контакти завдяки smpt, тобто реалізований сервіс посилення на пошту.

Та в розділі аналітика показана інформація для директорів, щоб бачити повну статистику таку, як тривалість, статистика по гроші, кількості угод, ефективність угод, процент виконаних та невиконаних угод, холодному виклику та інше.

Як висновок можна сказати, що дана система є досить цікавою та новітньою, застосовуються нові інноваційні методи з боку взаємодії з клієнтом, є кроссплатформною, має мобільні клієнти взаємодія з таблицями excel та має сучасний інтерфейс. Про методи збереження інформації та їх реалізації поки не відомо.

### **1.3 Розгляд CRM системи Битрикс24**

Дана CRM система доступна на українській мові, має безкоштовний акаунт з місцем у хмарі на 5гб, необмежений чат, кількість бізнес користувачів до 12, відео повідомлення, email листування, опитування, подяки, інтеграція з OneDrive, Google Drive, DropBox, редагування документів Microsoft Office, пошук по вмісту документа, календар, мобільні клієнти Android, IOS, підключення зовнішніх поштових сервісів, впровадження завдань та ін.

В системі присутній дружній та зрозумілий інтерфейс. Реєстрація проходить досить швидко та можлива реєстрація через зовнішні сервіси такі, як Google. На головній сторінці (рис 1.2) такі модулі, як

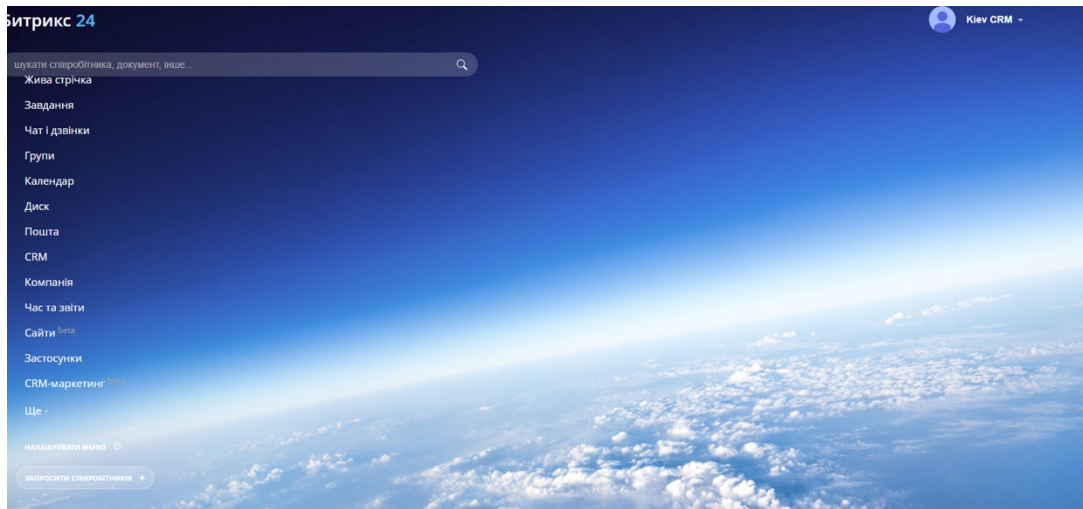


Рисунок 1.2- Головна сторінка Бітрікс24

жива стрічка, завдання, чат і дзвінки, групи, календар, диск, пошта, CRM, компанія, час та звіти, сайти, застосунки, CRM-маркетинг та ін.

У розділі жива стрічка представлені віджети де показані оголошення, завдання та популярні розділи. У розділі завдання продемонстрований сервіс видачі клієнтів з бази даних, та планування взаємодії з ними, а також є фільтрація даних. В розділі групи просто показано створення групи та її пошук. Також представлений сервіс календар, де можна запланувати подію. Присутній розділ диск для збереження даних, як було в описі. Є сервіс пошти, з підв'язкою відомих провайдерів цієї послуги.

Найцікавіший розділ виявився CRM, він і відкриває усю суть. В даному розділі є підрозділ “старт”, у якому показано навантаження по менеджерах та динаміка комунікацій. У підрозділі ліди, показана методологія канбан та календар з годинником. Підрозділ угоди демонструю етапи угоди, як нова, підготовка документів, рахунок на передоплату, в роботі, фінальний рахунок. Підрозділ рахунки розділений на дві частини, нові та відправлені клієнту. Підрозділ пропозиції, також розділений на дві ті самі частини. Підрозділ для заповнення контактів, а також такі підрозділи, як компанії, мої



справи та стрічка, де показана нова інформація. У підрозділі звіти показана, інформація по звітах операцій, по товарах, по лідам, по рахунках і справах.

У розділі компанія можна графічно побачити, як влаштована компанія (рис 1.3)

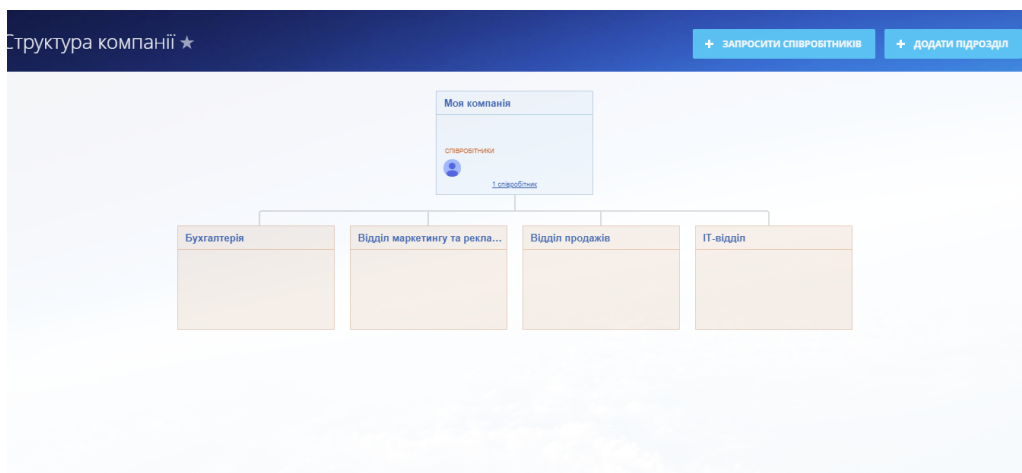


Рисунок 1.3- Розділ компанії

Тут можна додати співробітника та підрозділ. Є розділ час та звіти, де можна подивитись графік відсутностей, робочий час, робочі звіти та наради.

Розділ сайти працює у бета режимі та являється інноваційним можна створити свій сайт за даних шаблонів (рис 1.4), після чого створюється сайт є

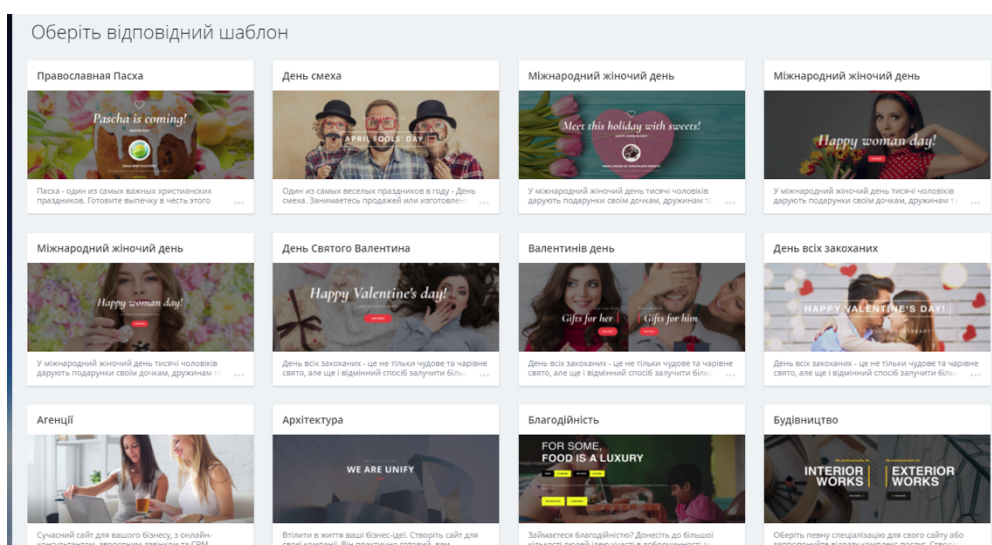


Рисунок 1.4- Шаблони для сайту

досить інноваційним та цікавим, впровадження CMS системи до сайту, що не реалізовував ніхто. Є CRM маркетинг у якому представлені сервіси для email послань, смс розсилок, месенджер та інфо дзвінків. Також в системі представлена телефонія та налаштування профілю.

Дана CRM система є досить цікавою, є безкоштовна версія для користування, присутня інтеграція з сучасними сервісами та використовується досить доступний для розуміння інтерфейс.

#### **1.4 Розгляд CRM системи Microsoft Dynamics CRM**

CRM від Microsoft є досить інноваційною в якій застосовуються найсучасніші технології, проте вона не є безкоштовною. Дана CRM система реалізує усі функції яким повинна мати CRM система завдяки впровадженню нових сервісів та можливостям Microsoft, особливо таким, як Skype та Microsoft Azure.

Система досить проста у навчанні та являється найшвидшою з усіх представлених вище CRM рішень, налаштування проходить досить швидко і все це завдяки вбудованій системі xRM. Оновлення та розширення функціональності відбувається регулярно, присутній максимально інтуїтивний інтерфейс. Інтеграція з Microsoft office 365 та з іншими системами та OneDrive. При розгляді CRM видно, що вона покриває великий спектр бізнес задач. Можна побачити всю взаємодію з клієнтами, повна аналітика та моніторинг.

Система поєднує в собі такі розділи, як продажі, сервіс та маркетинг. Розділ продажів дозволяє збільшити прибуток завдяки вбудованим системам продажів, командна робота, як джерело, рахунок усіх особливостей індустрії. Розділ сервіс займається плануванням звернень, планування та ведення у календарі. Особливістю є швидкий пошук, вбудовані бізнес-процеси, outlook синхронізація, можна працювати у ньому через CRM. Система має ієрархічну візуалізацію. Система має зручну навігацію даних. Дані у CRM можна проглядати у вигляді excel таблиць та в ворді. Використання OneDrive для

бізнесу та управління особистими документами. Огляд релевантних відомостей завдяки office Delve.

Можна створювати документи в один клік, без програмування шаблонів, можна створити шаблон через Office word. При інтеграції в excel, можна працювати у звичній формі, а CRM сама все перенесе. CRM використовує систему сповіщення Delve, яка використовує Machine Learning Azure. CRM App for Outlook

CRM має мобільні версії для IOS, Android, Windows Phone. Можна легко фільтрувати дані, має карточки kanban.

Система має такі розділи, як панелі моніторингу, дії, організації, контакти, інтереси, можливі угоди, конкуренти, продукти, профілі у соціальній мережі соціальній мережі, є елементи черги задач, управління продажами, календар задачі дзвінки, зустрічі, угоди можливість працювати в офлайн.

Інтерактивний сервіс обслуговування та аналітики (рис.1.5), спрощена навігація, показано звернення, дзвінки, задачі, електронна пошта, зустрічі, записи статті бази знань, контакти організацій, звернення, нові можливості запису, процес управління знаннями, інтерактивні панелі моніторингу.

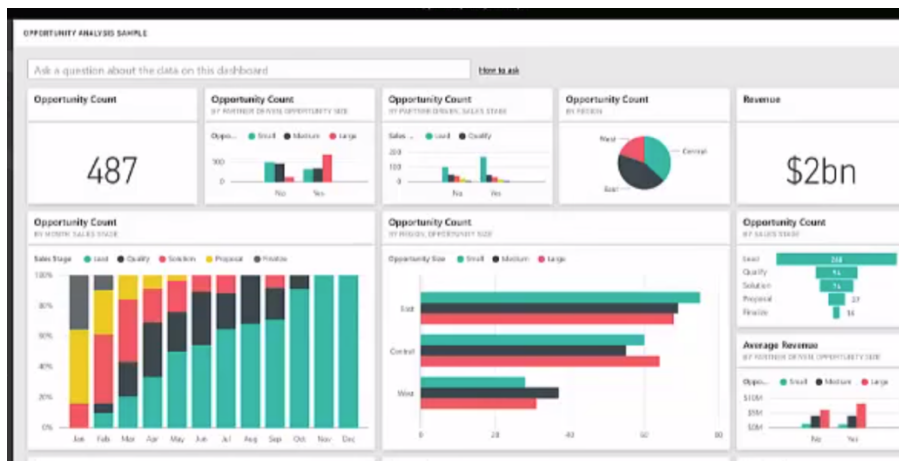


Рисунок 1.5- Microsoft Dynamics

Дана система має відмінну інтеграцію з сервісами та використовує усі новітні інновації завдяки розміщенню на платформі Microsoft Azure.

## 1.5 Розгляд CRM системи Microsoft Terrasoft

Дана CRM присутня на ринку України та має свій офіс у Києві, це рішення є досить ефективним при роботі з клієнтами та має досить зрозумілий інтерфейс(рис 1.6).

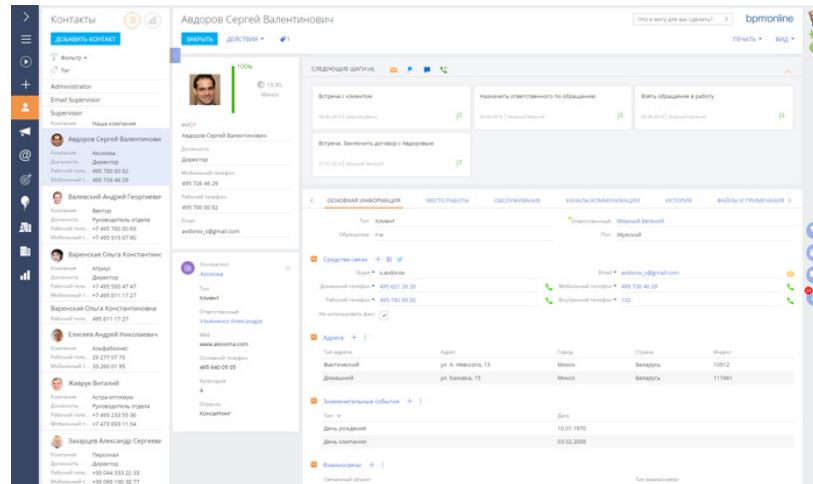


Рисунок 1.6- інтерфейс програми Terrasoft

Компанія, являється однією з провідних виробників CRM рішень, її основний продукт, це платформа bpm-online, яка поєднує у собі досить багато технологій CRM для управління взаємовідносинами з клієнтами. Завдяки гнучкості і масштабованості, платформа дозволяє створювати на її основі власні CRM-рішення, налаштовувати і допрацьовувати систему відповідно до особливостей галузі та специфіки завдань конкретної організації. Можна використовувати один з вже готових продуктів з широкої лінійки: bpm'online sales для управління продажами будь-якого типу, bpm'online marketing для автоматизації маркетингової активності компанії, bpm'online service для організації ефективного обслуговування клієнтів, а також bpm'online studio - для моделювання бізнес-процесів. Нас саме цікавить платформа bpm'online service так, як для дослідження сервісів, це рішення є досить ефективним.

Інформація про клієнтів в більшості організацій - це сукупність невідомих параметрів. І параметри ці до того ж можуть постійно змінюватися. Таким чином, завдання стоїть не в тому, щоб автоматизувати

взаємини з клієнтами, а в тому, щоб навчитися якомога ефективніше їх залучати та утримувати.

Система дозволяє створити єдину базу даних клієнтів, в якій зберігаються всі необхідні відомості: контактні дані, історія покупок, переваг і взаємодій з покупцями. Завдяки функції пошуку і злиття дублів, ваша клієнтська база завжди буде актуальною, а інформація в ній не буде повторюватися. А технологія інтелектуального збагачення даних дозволить «підтягувати» додаткову інформацію про клієнта з соціальних мереж і відкритих джерел.

Також система дозволяє, автоматизувати бізнес-процеси завдяки такій технології, як Business Process Management та Dynamic Case Management. Система окрім автоматизації процесів роботи з клієнтом, дозволяє моделювати, автоматизувати та керувати усіма бізнес-процесами підприємства, включаючи неструктуровані динамічні процеси. При моделюванні бізнес-процесів в системі утворюються нотації, які зрозумілі простим користувачам.

У даній системі присутня телефонія, інтеграція з зовнішніми сервісами та досить зрозумілий інтерфейс, доступна тріальна версія на 14 днів. Також у системі присутні усі базові сервіси для роботи з клієнтом.

## **1.6 Порівняння CRM рішень**

В попередніх підрозділах було досліджено доступні хмарні CRM-рішення, де кожна система мала свої переваги та недоліки. У даній таблиці(таб 1.1) досліджена порівняльна характеристика діючих CRM.

|                                   | Бітрикс24 | amoCRM | Terrasoft | SugarCRM | Dynamics CRM |
|-----------------------------------|-----------|--------|-----------|----------|--------------|
| Доступ до таблиць Excel           | ++        | ++     | +         | +        | +++++        |
| Інтеграція з обліковими системами | +         | +      | ++        | ++       | ++++         |
| Ціна грн/міс                      | 499       | 500    | 560       | 19 500   | 7 261        |
| Наявність Хмари                   | +++       | +      | ++        | +        | +++++        |
| Зручний інтерфейс                 | ++        | +++    | ++        | ++       | +++          |
| Телефонія                         | +         | +      | +         | +        | +            |
| Календар                          | +         | +      | +         | +        | +            |
| Аналітика                         | +         | +      | +         | +        | +            |
| Пошук                             | +         | +      | +         | +        | +            |
| Імпорт даних                      | Excel     | Excel  | Excel     | Excel    | Excel, Word  |
| Онлайн -допомога                  | +         | +      | +         | +        | +            |
| Наявність CMS                     | +         | -      | -         | -        | -            |
| Наявність соціальної мережі       | +         | -      | -         | -        | -            |
| Кросплатформність                 |           |        |           |          |              |

Таблиця 1.1-Порівняльна характеристика CRM систем

### 1.7 Висновки

В даному розділі було досліджено дійсні хмарні CRM, які на сьогодні є одними з найуспішніших та ефективними. Проте можна сказати, що усі дослідженні CRM мають свої переваги та недоліки. Була виконана порівняльна характеристика з якої можна було б сказати спершу, що Microsoft Dynamics CRM найкраща, проте вона і найдорожча.

Усі рішення мають інтеграцію з зовнішніми сервісами, досить легкі у користуванні, мають сервіси телефонії, збереження даних про користувача та

сервіс аналітики. SugarOrm показала себе, як найпростіша та надійніша система, проте не використовує нових технологій. AmoCRM показала себе, як система, яка має найкращий бізнес підхід та стратегію. Битрикс24, як єдина система, яка має CMS у своєму складі. Так, як наша ціль не тільки аналіз діючих CRM, а й розгляд інноваційних методів для розробки пропонуємо подивитися, які технології використовує кожна CRM.

## 2 Аналіз та узагальнення сучасних CRM

У розділі де розглядалися сучасні CRM-рішення, ми ознайомились з головними функціями та можливостями, проте аналіз та узагальнення ми ще не робили. Тому для аналіз CRM-рішень будемо спиратися на наступні показники:

- Яку технологію застосовує
- Являється чи являлась відкритою
- Який метод зберігання інформації використовує
- Чи має у своєму складі хмару
- Чи має у B2B сервіси
- Зберігаю інформацію та історію взаємовідносин з клієнтами
- Контроль роботи менеджерів
- Скорочення часу угоду
- Збір аналітичних даних та побудова звітів
- Автоматизація бізнес-процесів
- Чи має можливість запису телефонних розмов
- Має API для інтеграції з соціальними мережами, Telegram та іншими месенджерами.

Спираючись на ці показники, проаналізуємо наші розглянуті CRM-рішення.

**SugarCRM** використовувала в основному такі технології, як PHP та MySQL, Java Script та HTML. Являлася відкритою, проте зараз перейшла на платну версію, зокрема з'явилося багато фірм, які її переробили та використовують зі своїми модифікаціями зокрема, це такі фірми, як Відісофт, Integros та ін. Доречі сучасна CRM, SuiteCRM, являється також переробленою SugarCRM. Для зберігання інформації про клієнтів використовує реляційну базу даних. У своєму складі не має хмарного сервісу. Має сервіси B2B та зберігає інформацію про взаємовідносини з клієнтами. Контроль роботи менеджерів здійснюється не усіма версіями, проте здійснюється версією від української компанії Integros. Скорочення часу на угоду не передбачене, так як здійснюється тільки повний цикл. Автоматизація бізнес-процесів присутня та сервіс телефонії є завдяки інтеграції сервісів від Intelco. Можливість здійснювати запис розмови відсутній та можливий тільки у кастомних версіях. Не має API для інтеграції з соціальними мережами та месенджерами.

**AmoCRM** також використовує технологію PHP та MySQL, Java Script та HTML. Не являється відкритою та зроблена чисто у комерційних цілях. Для зберігання файлів теж використовує реляційну таблицю. Дозволяє скороти час угоди при взаємодії з клієнтом. Не має можливості хмарного збереження файлів. Має сервіси B2B та автоматизацію бізнес процесів. Зберігає інформацію та історію взаємовідносин з клієнтами. Здійснює повний контроль менеджерів, аналізує ефективність їх роботи та здійснює оцінку їх діяльності. Виконує збір аналітичних даних та будує по них звіти, присутня досить продумана система телефонії та сервіс, який здійснює запис телефонних розмов. Має інтеграцію з соціальними мережами та месенджерами.

**Бітрікс24** використовує такі технології, як PHP, MySQL, HTML, JavaScript. Не є відкритою системою, проте має перевагу у вигляді своєї хмари для зберігання файлів від 5 до необмеженого, але для зберігання



інформації про клієнтів використовує реляційну базу даних. Проводить контроль менеджерів. Дозволяє скоротити час угоди з клієнтом на його життєвому циклі. Має можливість повного збору аналітичних даних та побудову звітів. Має можливість автоматизації бізнес-процесів та дає можливість запису телефонних розмов для подальшого аналізування. Перевагою є повна інтеграція у соціальні мережі та месенджери, зокрема можна зазначити, що вся система сама нагадує соціальну мережу та надає можливість працівникам коментувати дію робітників та їх оцінку.

**Terrasoft** використовує технології .NET зокрема мову програмування C# базу даних MS SQL та Oracle, ASP.NET MVC, .NET core, RabbitMQ, JavaScript та HTML. Не являється відкритою та для зберігання інформації про клієнтів використовує SQL та NoSQL бази даних. На жаль у своєму складі хмари не має. Має у своєму складі сервіси B2B та автоматизацію бізнес-процесів зі зручним графічним інтерфейсом. Зберігаю певну історію взаємодії з клієнтами та дозволяє аналізувати ці дані, дає можливість скоротити життєвий цикл угоди. Має у своєму складі телефонію, та можливість запису розмов. Добре інтегровані API для взаємодії с соціальними мережами та можливість взаємодії з Excel. Здійснює контроль роботи менеджерів та їх оцінку. Загалом досить просунута система.

**Microsoft Dynamics CRM** дана CRM використовує майже всі сучасні технології від Microsoft такі, як мову програмування C#, SQL Azure, Azure Storage, ASP.NET MVC, SignalR та ін. Дане рішення є повністю комерційним, для зберігання інформації використовує SQL, NoSQL бази даних та сховище Azure Storage. Має у своєму складі хмару завдяки інтеграції з One Drive, та інтеграцію з усіма програмами пакету Microsoft Office. Має сервіси B2B та показу усього бізнес-процесу на Workflow діаграмах, а також автоматизує усі бізнес-процеси. Зберігає усю інформацію та історію взаємовідносин з клієнтом. Дає змогу проконтролювати роботу менеджера на всіх етапах життєвого циклу роботи з клієнтами та дозволяє скоротити життєвий цикл

угоди при потребі. Проводить збір аналітичних даних та будує по ним звіти. Має сервіси телефонії, зокрема інтеграція з Skype та дає можливість записувати телефоні розмови для подальшого аналізу. Має API для інтеграції з соціальними мережами та месенджерами.

## 2.1 Висновок

Було досліджено та проаналізовано сучасні CRM-рішення, показано, які технології використовує кожна CRM система. Проаналізовано сучасні можливості такі, як чи має система інтеграцію з excel, соціальними мережами, чи дає змогу прослідкувати за роботою менеджера. Розглянуто можливості телефонії та можливості запису дзвінків, а також проаналізовано основну можливу інформацію для побудови та аналізу сервісів для CRM систем.

## 3 Функціональність базових сервісів

На сьогодні досить багато CRM-рішень мають високу функціональність, проте кожне з них використовує базовий функціонал такий, як збереження контактної інформації про клієнта бо, як відомо попередником CRM була система Contact Management. Вона реалізовувала базові сервіси такі, як говорилося, збереження контактної інформації, інформації по виконаним угодам та календар з плануванням. Потім додавалися сервіси для контролю та управління продажами, де базовими сервісами були, внесення та визначення потенціальної угоди, аналіз та оптимізація циклу продажів, прогнозування продажів та згодом усі ці сервіси комбінувалися між собою.

Для початку розглянемо базовий сервіс календар. Він дозволяє на відповідну вибрану дату, назначити подію та зберегти її до бази даних, в зазначеній події буде запущений таймер, де після його спрацьовування буде виконана запланована дія така, як повідомлення через email або смс. Другий

сервіс, який досліджувався був сервіс комунікації. До сервісу комунікації можна віднести, як чат так і телефонію. Базовим сервісом є занесення клієнта до бази даних та базових CRUD операцій з ним. Створення сервісу циклу взаємодії з клієнтом. Сервіс для збереження файлів у хмарі та його ефективно розміщення у ній. Окремий сервіс оповіщення клієнта за допомогою смс. Також досить ефективним є сервіс занесення інформації з бази даних до excel таблиці та завантаження.

Треба зазначити, що усі базові сервіси повинні реалізувати одне із трьох рішень:

1. Автоматизація продажів
2. Автоматизація маркетингу
3. Автоматизація обслуговування клієнта

### **3.1 Висновок**

В даному розділі були досліджено функціонал базових сервісів, які зустрічаються у CRM-системах. Такі як, сервіси телефонії, сервіси збереження інформації про клієнта, створення циклу угод з клієнтом, сервіс повідомлень через пошту або повідомлення на телефон та календар, як сервіс.

## **4 Пошук та вибір необхідних сервісів**

При аналізі діючих CRM-рішень постало питання, щоб використати діючі сервіси, які є на просторах інтернету та додати для покращення існуючих CRM рішень. Та для аналізу та пошуку даних сервісів, треба було визначитися, які саме сервіси нам потрібні. Тому перше, що ми зробили це аналізували, які сервіси найголовніші у CRM системі і першим був, сервіс по управлінню контактами, який дозволяє CRM системі здійснювати збереження контактів про клієнтів, замовників, конкурентів та ін., де вся інформація є унікальною. Другим був сервіс по управлінню взаємодією з

клієнтами, який дозволяє отримати повну картинку взаємовідносин з окремим клієнтом, з моменту його появи, що дозволяє проводити аналіз майбутніх потреб клієнта, та буде ґрунтуватися на тих даних, які уже є в системі. Потім також потрібні сервіси, які будуть управляти угодами та планувати усі дії в залежності від дати, це може бути сервіс календар у якому можна, це все зробити. Не треба забувати, що кожна CRM система потребує автоматизування, тому важливим буде сервіс, який дозволить зменшити затрати на підготовку угод та зробити, це максимально швидко. І головним сервісом буде сервіс для аналізу та формування цільової аудиторії, у якому буде контактна інформація, яка зберігається у системі та дозволяє зробити розсилки за допомогою смс та на пошту і мати функцію телефонії.

З попереднього аналізу будемо здійснювати пошук можливих сервісів, і для початку проаналізуємо інструментарій, який надає нам FiWare. На сервісі FiWare, який налічує у своєму складі більш ніж сотню готових сервісів. Було проаналізовано більшу частину сервісів та запущено їх завдяки Докер контейнеризації. За основу було взято такі сервіси, як сервіс взаємодії з сутностями та сервіс для роботи з CRUD операціями. Для обрання сервісу для роботи з CRUD операціями був обраний сервіс CRUD-Operations, який написаний на Shell script та має багатий функціонал. Також був обраний сервіс для роботи з хмарними збереженнями даних, таких як Google Drive та Microsoft One Drive.

Проте одним ресурсом FiWare ми не обійшлись, через те що його можливостей було не достатньо. Таким чином ми почали шукати сервіси на github та Докер хаб. При дослідженні сервісів на Докер хаб, були знайдені основні сервіси SugarCRM. (рис 4.1) Серед взятих сервісів були обрані сервіси взаємодії з клієнтом, сервіс аналізу клієнта та сервіс планування справ.

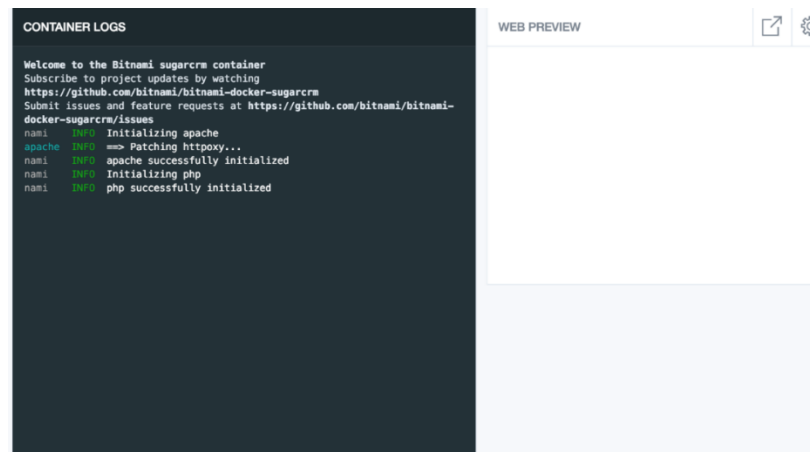


Рисунок 4.1- Аналіз сервісів SugarCRM

Були дослідженні та застосовані такі сервіси, як Redis для збереження даних у оперативній пам'яті, MongoDB яка являє собою NoSql базу даних. Серед знайдених сервісів були, сервіси для здійснення смс повідомлень та передачі повідомлень через Email. Таким чином в мережі інтернет були знайдені та дослідженні основні сервіси для CRM-рішень, які надалі будуть протестовані та застосовані у дослідженнях.

#### 4.1 Висновок

В даному розділі були знайдені головні сервіси для CRM-рішень, які застосовуються при розробці системи. Були проаналізовані такі ресурси, як гітхаб, докер хаб та головним ресурсом виявився Fiware , який має збірку сервісів для взаємодії зокрема сервіси CRUD. Зокрема сервіси були протестовані та запуснені за допомогою Docker.

## 5 Інструментарій для прототипування сучасних сервісних додатків

З попередніх розділах ми оглянули та дослідили основні сучасні CRM системи, які знаходяться у хмарі та зробили їх аналіз і узагальнення, зокрема знайшли частину сервісів для використання для хмарних CRM-рішеннях. Тепер візьмемо за основу CRM-рішення Бітрикс24 та аналізуючи його поведінку та сервіси, зробимо дослідну роботи по можливості його покращення, а також використаємо експериментальний стенд для роботи з

сервісами. Для прототипування CRM системи, перше, що повинно бути зроблено, це технічне завдання для системи. Тобто повинна бути показана інформація для чого це рішення, яким чином буде проходити створення проекту описання процесів та методології. Який результат отримаємо на виході. Як цільова аудиторія, які рамки проекту та вимоги. Та головний функціонал. Прототип, являється рішенням для зменшення витрат у кінцевому продукті. Головним інструментарієм для побудови CRM у наш час є вирішення, якого хмарного провайдера обрати. Які базові технології при розробці будуть застосовані та, як будуть зберігатися дані

### 5.1 Обґрунтування вибору хмарного провайдера та їх огляд

Для прототипування додатку треба обрати хмарного провайдера, за критерієм ціни та можливостей, які він надає. Дослідимо основних хмарних провайдерів на ринку таких, як Microsoft Azure, Google Cloud App Engine та Amazon.

**Google App Engine** є платформою веб-рамок та хмарних обчислень для розробки та розміщення веб-додатків у центрах обробки даних, керованих Google. Програми ізольовані один від одного та працюють на декількох серверах. App Engine пропонує автоматичне масштабування для веб-програм, оскільки при додаванні додатків зростає кількість запитів, App Engine автоматично виділяє більше ресурсів для веб-додатків та обробки додаткового запиту.

Google App Engine має певний рівень споживаних ресурсів. Плата стягується за додаткове зберігання, пропускну спроможність або години використання ресурсів, що вимагаються заявкою. Вперше вона була випущена як попередня версія у квітні 2008 року та вийшла з попереднього перегляду у вересні 2011 року.

Підтримувані мови програмування включають Python, Ruby, Java (і, до того ж, інші мови JVM, такі як Kotlin, Groovy, JRuby, Scala, Clojure), Go та

PHP. Node.js також доступний у гнучкому середовищі. Google заявив, що планує підтримувати більше мов у майбутньому і що Google App Engine написано як незалежний від мови. С # також підтримується, проте не повністю. Підтримуються довільні контейнери Docker .

Python фреймворки, які працюють на Google App Engine включаючи Django, Flask та інші, а також призначені для користувача Google складено структуру WebAPP і ряд інших, розроблених спеціально для платформи, яка виникла з моменту випуску. Будь-яка платформа Python, яка підтримує WSGI за допомогою CGI-адаптера, може бути використана для створення програми; Рамки можна завантажити разом із розробленою програмою. Можна також завантажити сторонні бібліотеки, написані на чистому Python.

Google App Engine підтримує багато стандартів і рамок Java. Основним до цього є технологія серверлета 2.5, що використовує Jetty Web Server із відкритим вихідним кодом та супутні технології, такі як JSP. Програма JavaServer Faces працює з деякими обхідними шляхами. Підтримує Java8, Servlet 3.1 та Jetty9.

Хоча інтегрована база даних, Google Cloud Datastore, може бути незнайома програмістам, вона доступна і підтримується з JPA, JDO та простим API. Є кілька альтернативних бібліотек і рамок, які можна використовувати для моделювання і відображення даних в базу даних, такі як об'єктивізації, Slim3 та ін.

Spring Framework працює з GAE. Однак модуль Spring Security вимагає обхідних шляхів. Підтримка Apache Struts, підтримується Struts 2, який працює з обхідними шляхами.

Веб-структура Django та програми, що працюють на ньому, можуть бути використані в App Engine з модифікацією. Django-nonrel має на меті дозволити Django працювати з не реляційними базами даних, а проект включає підтримку App Engine.

**Amazon AWS** надає платформу хмарних обчислень на вимогу приватним особам, компаніям та урядам на платній підписці. Хмарний

провайдер дозволяє користувачам мати у своєму розпорядженні повний віртуальний кластер комп'ютерів, який постійно доступний через Інтернет. Версія віртуальних комп'ютерів AWS має більшість атрибутів реального комп'ютера, включаючи апаратне забезпечення для обробки, вибір операційних систем, мережі і попередньо завантажені прикладні програми, такі як веб-сервери, бази даних. Кожна система AWS також віртуалізує консольний ввід / вивід, що дозволяє підключенням AWS підключитися до своєї системи AWS за допомогою сучасного браузера. Браузер виступає в якості вікна на віртуальний комп'ютер, дозволяючи абонентам заходити до системи, налаштовувати і використовувати їхні віртуальні системи так, як і на реальному фізичному комп'ютері. Вони можуть вирішити розгортати свої системи AWS, щоб надавати послуги Інтернету для своїх власних потреб і для своїх клієнтів.

Технологія AWS реалізується на серверних фермах у всьому світі та підтримується дочірніми компаніями Amazon. У рамках підписної угоди, компанія Amazon керує, оновлює та забезпечує стандартну безпеку для кожної абонентської системи. AWS працює з багатьох глобальних географічних регіонів, у тому числі 6 в Північній Америці.

У 2017 році AWS включала понад 90 сервісів, що охоплюють широкий спектр, включаючи обчислення, зберігання, мережу, базу даних, аналітика, служби додатків, розгортання, керування, мобільні засоби, інструменти для розробників та інструменти для Інтернету речей. Найпопулярніші включають Amazon Elastic Compute Cloud (EC2) та Amazon Simple Storage Service (S3). Основна частина служб не піддаються безпосередньо кінцевим споживачам, але замість цього пропонують функціонал можливостей через API, які потім розробники будуть використовувати у своїх програмах. Треба зазначити, що Amazon Web Services надає доступ через HTTP, використовуючи REST чи протокол SOAP.

Amazon AWS markets для абонентів, як спосіб отримання великомасштабної обчислювальної потужності більш швидко та дешево, ніж



побудова фактичної фізичної серверної ферми. Всі послуги оплачуються по мірі використання та кожна служба окремо вимірює використання різними способами.

**Microsoft Azure** надає готову модель PaaS де є готові рішення при використанні технології ASP.Net. Проте треба також розуміти, що кожна CRM потребує збереження даних, а саме інноваційним методом завдяки якому ,буде досить легко їх масштабувати і головне захищені та репліковано , як мінімум тричі [1].

І тут платформа Microsoft Azure пропонує нам готові рішення такі, як Microsoft Azure Storage, SQL Azure та Azure Cosmos DB. Кожен тип збереження даних є інноваційним завдяки модернізаціям.

Azure Cosmos DB представляє собою базу даних NoSQL, що дозволяє гнучко і незалежно масштабувати пропускну здатність і ресурси сховища в будь-яких регіонах Azure, варто лише натиснути кнопку. Вона гарантує пропускну здатність, мінімальну затримку, доступність і узгодженість відповідно до угод про рівень обслуговування (SLA), які часто не можуть запропонувати інші служби баз даних. Та головною особливістю Cosmos DB є те що вона підтримує багато API, що надає можливість поводити себе у різних ситуаціях, по різному (API SQL, API MongoDB, API Cassandra, API Gremlin (Graph), API таблиць) [1].

Azure Storage надає нам можливість збереження даних у Tables, Blobs, Queues. Table's дозволяє нам зберігати інформацію у форматі ключ, значення, як звичайна NoSql. Blobs розділяється на blocks та pages. При використанні blocks інформація записується у окремі бінарні блоки, а в pages на окремі бінарні сторінки , що надає нам можливість зменшувати розмір відео файлів, так як часто бувають сторінки містять 0 байти, які Azure видаляє та запам'ятовує номер сторінки та при потребі повертає їх назад , що дозволяє нам економити пам'ять [7].

Також завдяки Azure SDK можна створювати сервісну хмару, де використовується дві моделі, Worker-role та Web-role, де про них буде описано у наступних підрозділах.

Головною перевагою Microsoft Azure є оплата ресурсів по їх використанню, але також надає цю можливість Amazon та Google, проте за більшу ціну.

Таким чином із запропонованих рішень для дослідження та прототипування даних оберемо хмару Microsoft Azure для побудування експериментального стенду з яким будуть взаємодіяти сервіси.

## 5.2 Microsoft Azure PaaS

Платформа як послуга (PaaS) - це повноцінне середовище розробки і розгортання в хмарі з ресурсами, які дозволяють надавати будь-які додатки, від простих хмарних додатків до просунутих хмарних додатків промислового класу. Ви купуєте необхідні ресурси у постачальника хмарних служб, платите в міру використання ресурсів. Як і у випадку з IaaS, PaaS включає інфраструктуру (сервери, сховище та мережеве обладнання), а також ПО проміжного шару, засоби розробки, бізнес-аналітику (BI), служби системи управління базами даних і інше. Послуга PaaS призначена для підтримки повного життєвого циклу розробки, тестування, розгортання, управління та оновлення.(рис 5.1)

PaaS також дозволяє уникнути витрат і труднощів, пов'язаних з придбанням ліцензій програмного забезпечення і управлінням ними, базової інфраструктури додатків, ПЗ проміжного рівня, засобів розробки і інших ресурсів. Ви керуєте додатками і службами, які розробляєте, а постачальник хмарних служб зазвичай керує всім іншим [7].

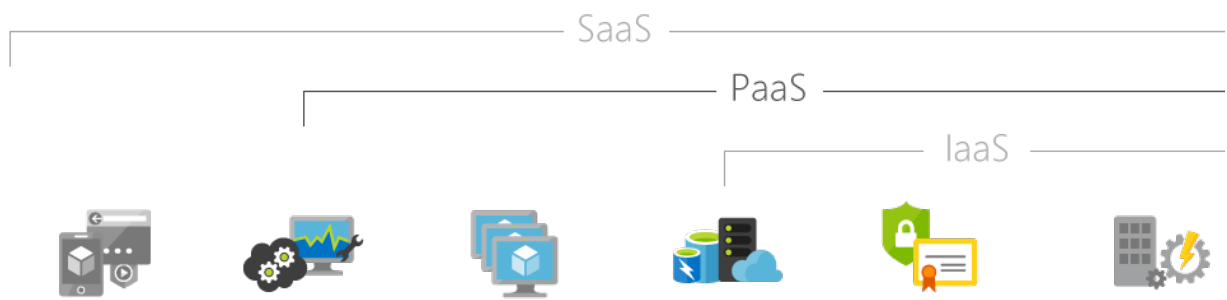


Рисунок 5.1- Зображення хмарних моделей

Середовище для розробки. PaaS надає середу, яку розробники використовують для розробки або налаштування хмарних додатків. Аналогічно тому, як створюється макрос для Excel, PaaS дозволяє розробникам створювати додатки з використанням вбудованих компонентів програмного забезпечення. Хмарні функції, такі як масштабування, висока доступність і підтримка декількох користувачів, вже включені і знижують обсяг коду, який необхідно розробляти [2].

Бізнес-аналітика. Кошти, надані в рамках PaaS, дозволяють організаціям аналізувати дані, знаходити тенденції і робити прогнози з метою поліпшення планування, рішень по продуктах, підвищення повернення від інвестицій і прийняття інших бізнес-рішень.

Додаткові служби. Постачальники PaaS можуть пропонувати інші служби, які підвищують можливості додатків, такі як робочі процеси, каталоги, безпеку і планування.

Надаючи інфраструктуру, як послугу, PaaS пропонує більше переваг, ніж IaaS.

Скорочення часу програмування. Засоби розробки PaaS можуть скоротити час, необхідний для програмування нових додатків завдяки заздалегідь підготовленим компонентам, вбудованим в платформу, включаючи робочі процеси, служби каталогів, компоненти безпеки, засоби пошуку та т. Д.

Додавання можливостей розробки без збільшення числа співробітників. Компоненти платформи як послуги надають вашій команді розробників нові

можливості без необхідності наймати співробітників з відповідними навичками.

Спрощена розробка для декількох платформ, включаючи мобільні платформи. Деякі постачальники служб надають вам варіанти розробки для декількох платформ, наприклад комп'ютерів, мобільних пристроїв і браузерів, спрощуючи і прискорюючи таким чином кроссплатформену розробку додатків.

Економічне використання просунутих засобів. Оплата по мірі використання дозволяє фізичним і юридичним особам використовувати просунуті засоби розробки і бізнес-аналітики, які можуть бути занадто дорогими для придбання у власність.

Підтримка географічно розподілених команд розробників. Оскільки доступ до середовища розробки здійснюється через Інтернет, команда розробників може працювати над одними проектами, навіть коли члени команди перебувають в різних місцях.

Ефективне управління життєвим циклом додатків. PaaS забезпечує всі можливості, які будуть потрібні для підтримки повноцінного життєвого циклу веб-додатків: створення, тестування, розгортання, управління і поновлення всередині одного інтегрованого середовища.

### **5.3 Створення основи стенду завдяки Microsoft Cloud Services**

Досить багато сервісів повинні взаємодіяти один з одним, особливо коли це стосується CRM системи і щоб це реалізувати на платформі Azure, відкриємо вкладку у Visual Studio та оберемо Azure Cloud Services. Для створення Хмарного сервісу потрібно завантажити Visual Studio з офіційного сайту Microsoft, для цього попередньо треба завести Microsoft акаунт. Після цього треба завантажити azure sdk, теж з офіційного сайту, після створити новий проект та натиснути на вкладці Cloud та обрати Azure Cloud Service(рис 5.2).

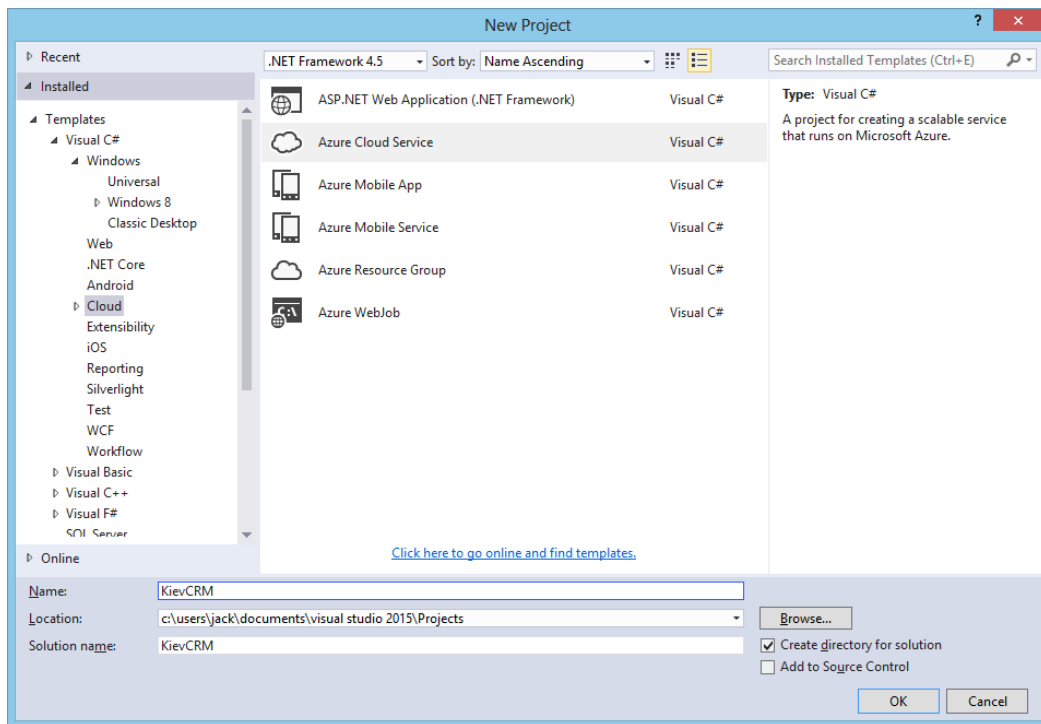


Рисунок 5.2 – Створення Cloud Service

Після чого оберемо ролі Web Role та Worker Role і натиснемо Ок (рис 5.3)

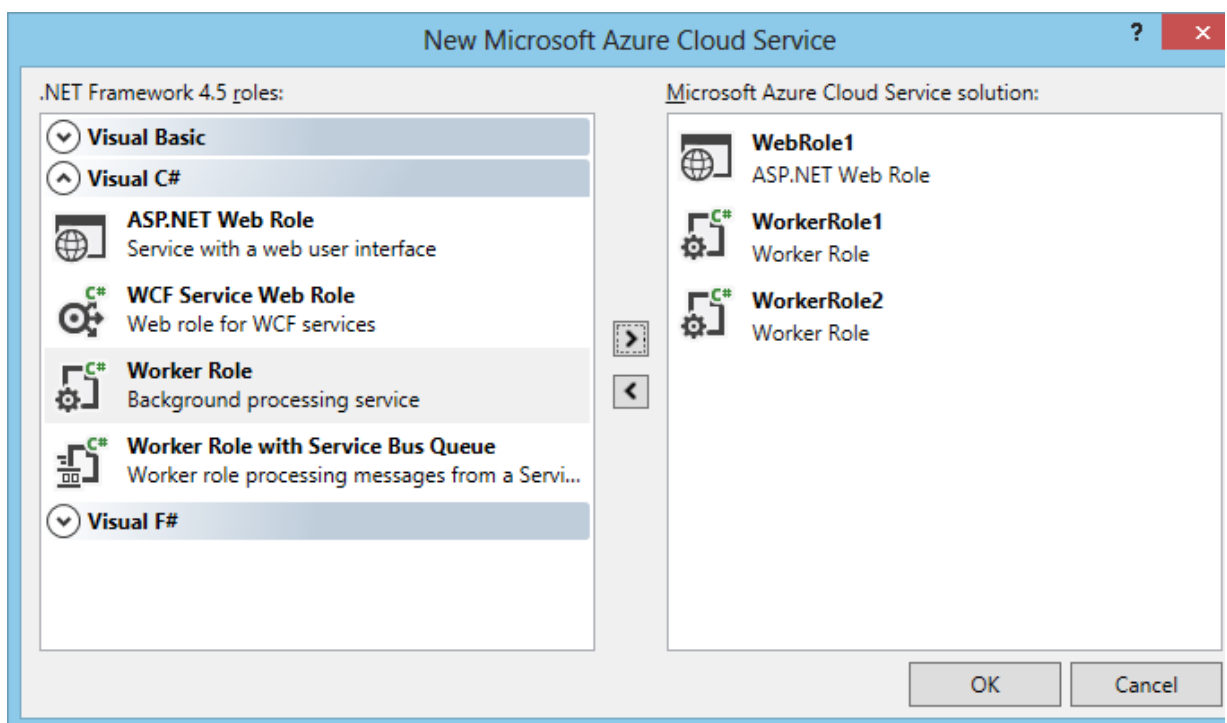


Рисунок 5.3 – Обрання ролей

Потім оберемо вид ASP.NET шаблону, в даному розділі був обраний MVC (Рис 5.4)

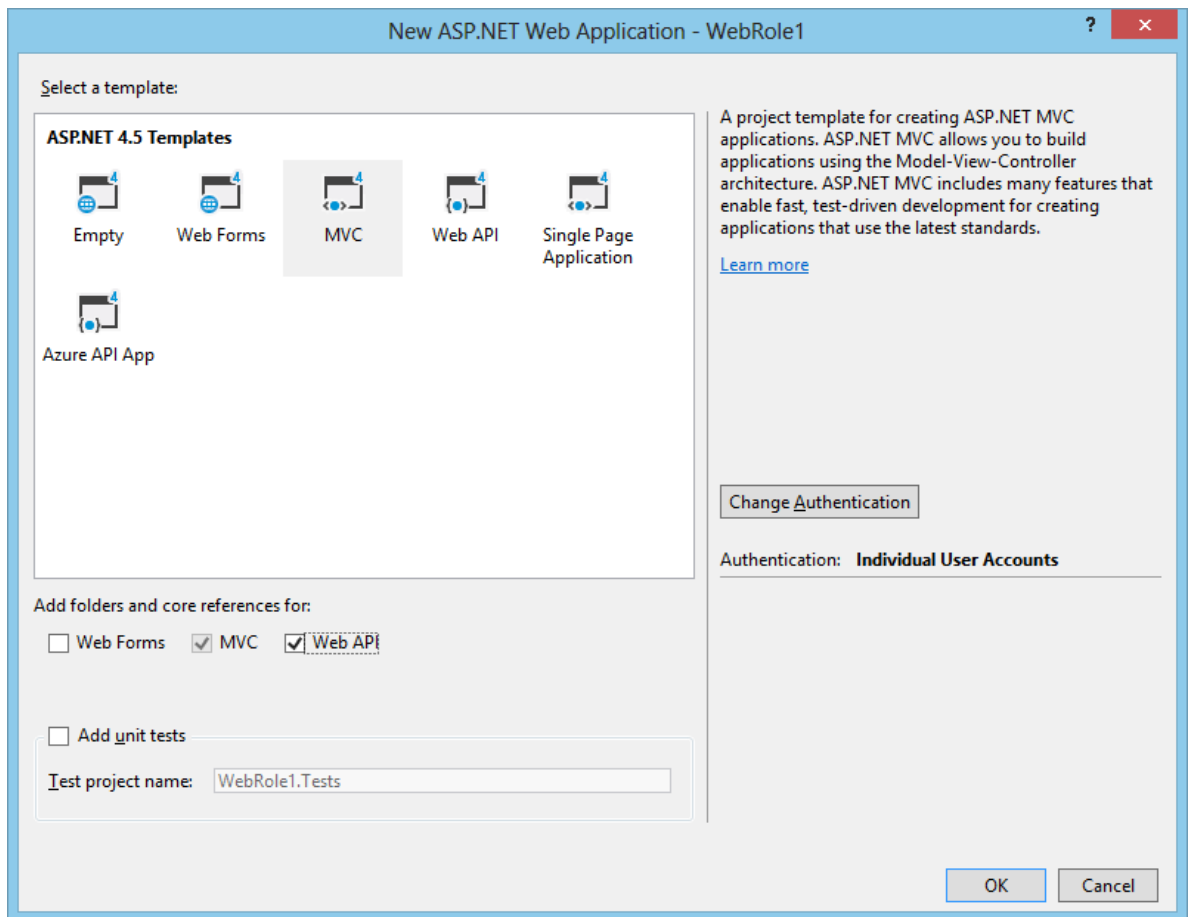


Рисунок 5.4 Обрання Шаблону ASP.Net

Після чого у нас створюються ролі ASP.Net(Web-role) проект та два проекти C#(Worker-role) (рис 1.5)

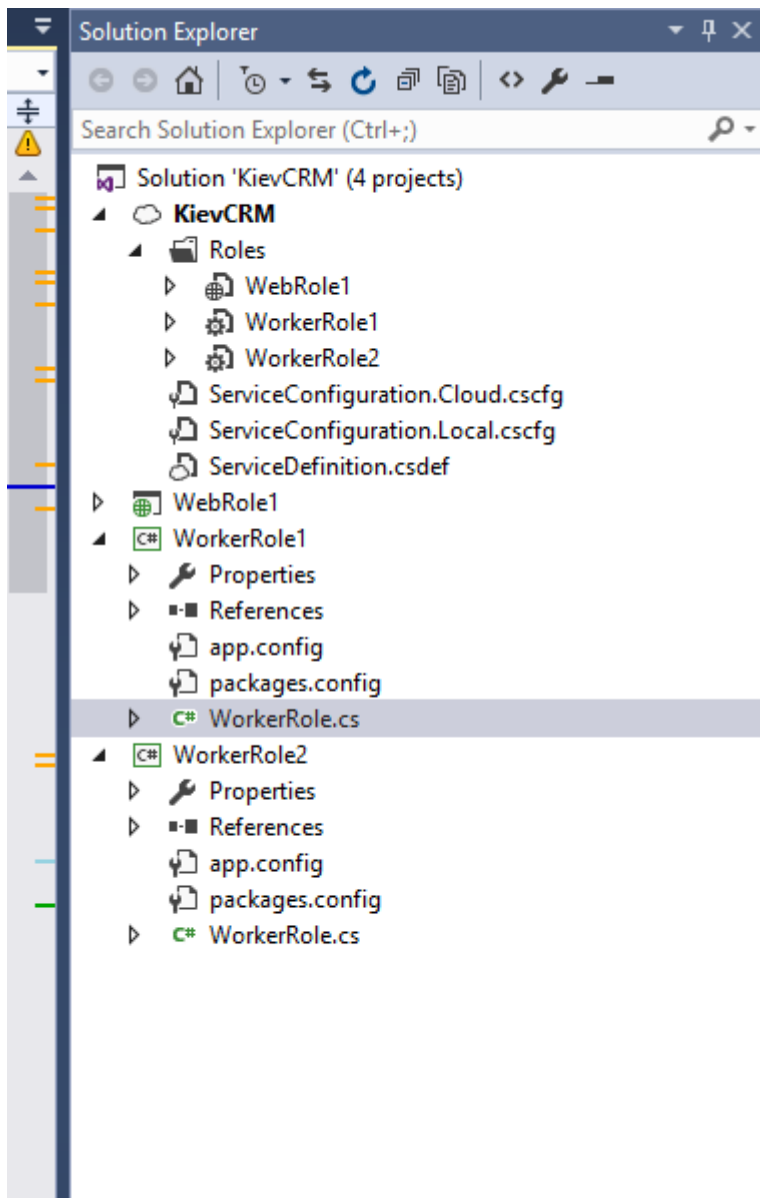


Рисунок 5.5 – Ролі Azure

Існує два типи ролей хмарних служб Azure, єдина відмінність між ними - спосіб розміщення ролі у віртуальній машині.

**Веб-роль:** автоматично розгортає і розміщує додатки за допомогою IIS.

**Робоча роль:** не використовує IIS і запускає додаток автономно.

Наприклад, простий додаток може використовувати тільки одну веб-роль, яка обслуговує веб-сайт. Більш складний додаток може використовувати веб-роль для обробки надходження від користувачів запитів, а потім передавати ці запити в робочу роль для обробки. Це повідомлення може використовувати службова шина Microsoft Azure Service Bus або сховище черг Azure.

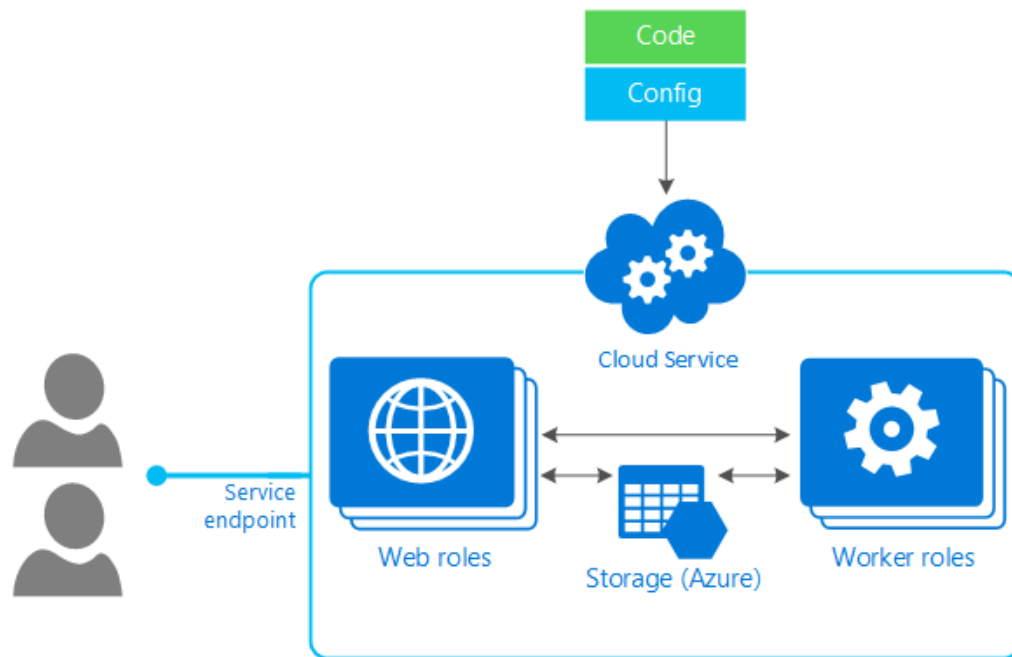


Рисунок 5.6 – Взаємодія між ролями

Як видно з (рисунка 5.6), все віртуальні машини в рамках окремого додатка виконуються в одній хмарній службі. Користувачі здійснюють доступ до додатка через окремих загальнодоступний IP-адреса, при цьому для віртуальних машин додатку виконується автоматичне балансування навантаження. Платформа масштабує і розгортає віртуальні машини в додатку хмарних служб Azure таким чином, щоб виключити єдину точку відмови обладнан

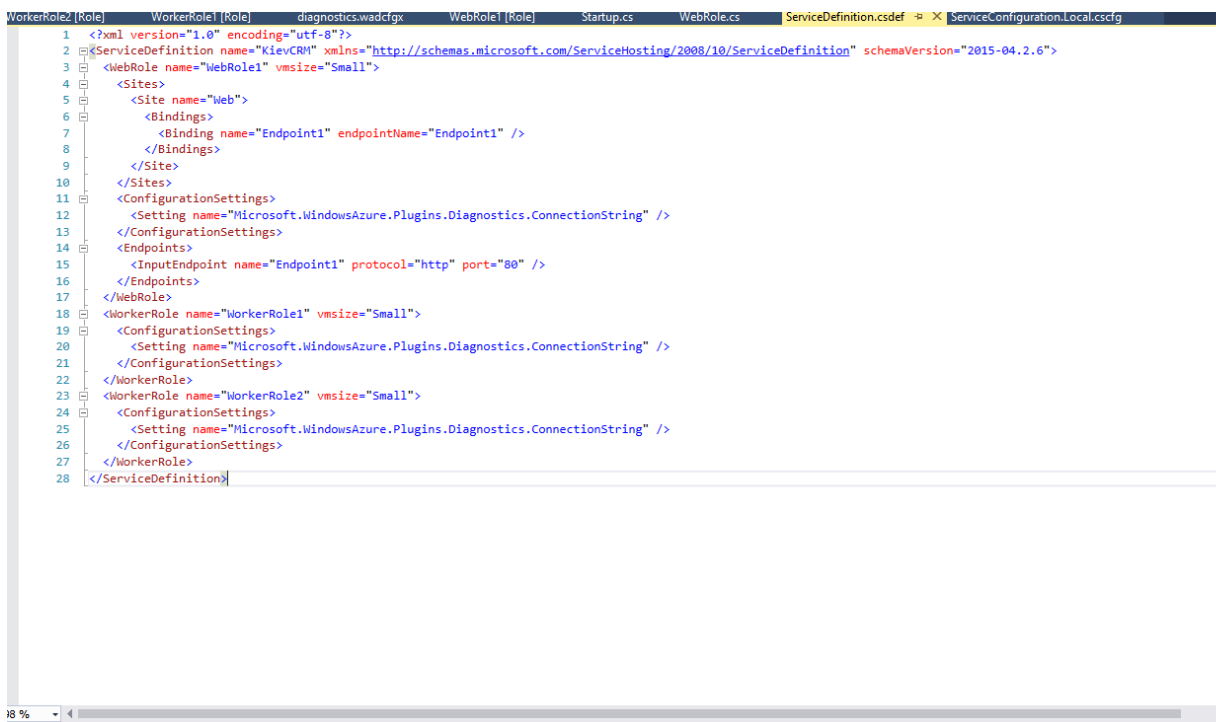
Всупереч тому, що програми виконуються в віртуальних машинах, важливо розуміти, що хмарні служби надаються не як IaaS, а PaaS. Нижче розглядається один зі способів. У моделі IaaS, такий як віртуальні машини Azure, спочатку створіть і налаштуйте середу, в якій буде працювати додаток. Потім розгорніть додаток в цьому середовищі. При такому підході на вас лягає безліч обов'язків, наприклад, по розгортанню нових виправлених версій операційної системи в кожній віртуальній машині. У PaaS ж середовище вже існує. Вам потрібно всього лише розгорнути свій додаток. Управління платформою, на базі якої воно виконується, включаючи розгортання нових версій операційної системи, здійснюється за вас.



При розгортанні сервісу на Azure всі ролі перетворюються на dll бібліотеки та генерується один файл \*.cspkg та один файл конфігурації \*.cscfg.

У файлі конфігурації вкажемо скільки віртуальних машин, будемо використовувати на кожну роль.

Тепер для налаштування сервісу виберемо файл- ServiceDefinition.csdef , який зображений на (рис 1.4).Після чого у відкритому вікні ми побачимо



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <ServiceDefinition name="KievCRM" xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition" schemaVersion="2015-04.2.6">
3   <WebRole name="WebRole1" vmSize="Small">
4     <Sites>
5       <Site name="Web">
6         <Bindings>
7           <Binding name="Endpoint1" endpointName="Endpoint1" />
8         </Bindings>
9       </Site>
10    </Sites>
11    <ConfigurationSettings>
12      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString" />
13    </ConfigurationSettings>
14    <Endpoints>
15      <InputEndpoint name="Endpoint1" protocol="http" port="80" />
16    </Endpoints>
17  </WebRole>
18  <WorkerRole name="WorkerRole1" vmSize="Small">
19    <ConfigurationSettings>
20      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString" />
21    </ConfigurationSettings>
22  </WorkerRole>
23  <WorkerRole name="WorkerRole2" vmSize="Small">
24    <ConfigurationSettings>
25      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString" />
26    </ConfigurationSettings>
27  </WorkerRole>
28 </ServiceDefinition>

```

Рисунок 5.7 – Файл конфігурації ServiceDefinition.csdef

ноди (рис 1.7), які описують наші ролі. Також є файли конфігурації один локальний, а інший для хмари у якому як ми казали можна обрати на скількох віртуальних машинах будуть запущені наші ролі (рис 1.8)

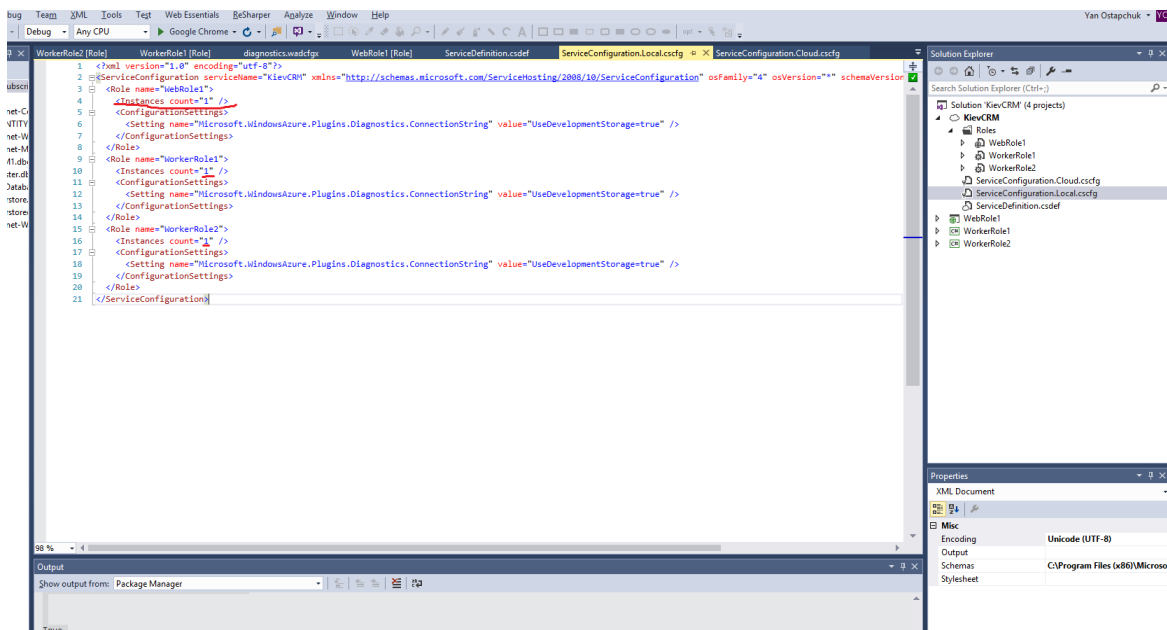


Рисунок 5.8 – Файл конфігурації ServiceConfiguration.Local.cscfg

Після всього цього ми можемо завантажити наш сервіс, до хмари Microsoft Azure. Для завантаження натиснемо праву кнопку миші та виберемо Publish (рис 5.9)

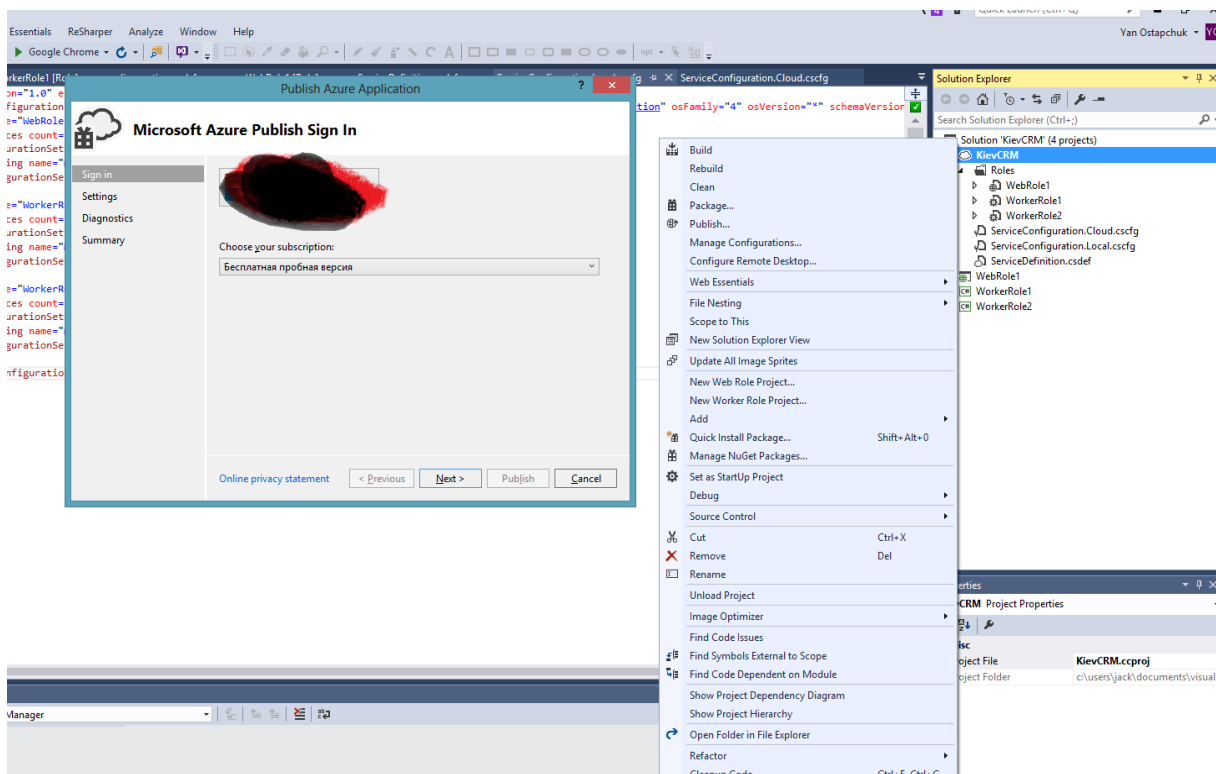


Рисунок 5.9 – публікація сервісу на Azure

## 5.4 Використання Azure Load Balancer для прототипування

При навантаженні платформи досить часто виникає питання, як боротися з тафיקом та Ddos атаками для цього використаємо інструментарій Microsoft Azure для покращення роботи Бітрікс24.

Azure Load Balancer дозволяє масштабувати додатки і забезпечувати високий рівень доступності служб. Load Balancer підтримує як вхідні, так і вихідні сценарії, забезпечує низьку затримку і високу пропускну здатність, а також збільшує масштаб до мільйонів потоків для всіх додатків, які використовують протоколи TCP і UDP.

Load Balancer буде розподіляти нові вхідні потоки, що надходять на зовнішній інтерфейс підсистеми балансування навантаження, відповідно до правил і перевірки працездатності.

Крім того, загальнодоступний Load Balancer може також надати вихідні підключення для віртуальних машин в рамках віртуальної мережі шляхом перетворення їх приватних IP-адрес в загальнодоступні IP-адреси.

Доступно два типи SKU Azure Load Balancer рівня: "Базовий" і "Стандартний". У них є деякі розбіжності за масштабом, можливостями і ціною. Будь-який сценарій, який можна виконати в рамках Load Balancer рівня "Базовий", також можна створити за допомогою Load Balancer рівня "Стандартний", хоча підхід може дещо відрізнятись. У міру отримання додаткових відомостей про Load Balancer, необхідно ознайомитися з основами і відмінностями в SKU.

Load Balancer дозволяє використовувати такі функції :

- балансування навантаження вхідного інтернет-трафіку віртуальних машин. Така конфігурація називається загальнодоступним Load Balancer.
- Балансування навантаження трафіку між віртуальними машинами у віртуальній мережі. Крім того, можна підключитися до зовнішнього інтерфейсу Load Balancer з локальної мережі в

гібридному сценарії. В обох сценаріях використовується конфігурація, відома як внутрішній Load Balancer.

- Перенаправлення трафіку за допомогою порту на певний порт в певній віртуальній машині з використанням правил перетворення мережевих адрес для вхідних підключень.
- Забезпечення вихідного підключення для віртуальних машин у віртуальній мережі шляхом використання загальнодоступного Load Balancer.

## 5.5 Використання шини для прототипування стенду Azure Service Bus

Надалі ми не будемо використовувати шину із за її вартості і перейдемо до інших технологій, проте для огляду з точки зору покращення роботи дійсної CRM використаємо її у експериментальному стенді для тестування. Через те, що при роботі у хмарі додатки взаємодіють з іншими додатками та службами [2].

Service Bus Azure - це високонадійна хмарна система для обміну повідомленнями між додатками і службами, яка ефективно працює, навіть коли кілька з них недоступні в мережі. Ця повністю керована служба доступна у всіх регіонах Azure. Для неї не потрібно купувати ліцензії або витрачати ресурси на обслуговування серверів. Її асинхронна робота дозволяє клієнтам гнучко обмінюватися повідомленнями з сервером через брокер, а обробка повідомлень в порядку надходження і можливості публікації і підписки роблять це рішення відмінним вибором для таких сценаріїв, як обробка замовлень [2].

У різних ситуаціях, при взаємодії між сервісами можуть бути використані, як звичайні черги так і http запити. Інколи кращим рішенням є обмін повідомленнями між додатками через звичайну чергу. В іншій ситуації може бути таке, що звичайної черги може не вистачити і треба

використовувати чергу з можливістю публікації та підписки. В деяких випадках необхідно лише підключення між додатками, а черга не потребується. Службова шина, яка представлена Microsoft має три варіанта, даючи можливість додаткам взаємодіяти між собою різними способами [3].

Сама шина Microsoft Service Bus являє собою хмарну службу, якою можуть користуватися декілька користувачів у режимі реального часу. Кожен користувач створює простір імен та обирає механізм взаємодії, які повинні у ньому використовуватися. Ця архітектура показана на (рис.1.5)

У рамках простору імен можна використовувати один чи декілька екземплярів трьох механізмів взаємодії, які пов'язують додаток по різному

Слідуючі варіанти

1. Черга яка підтримує одно направлену взаємодію. Кожна черга діє як посередник, який зберігає відправлені повідомлення , поки вони не будуть отримані. Кожне повідомлення отримує один користувач.
2. Розділи які підтримують одно направлену взаємодію з використанням підписок. В одному розділі може буту декілька підписок.

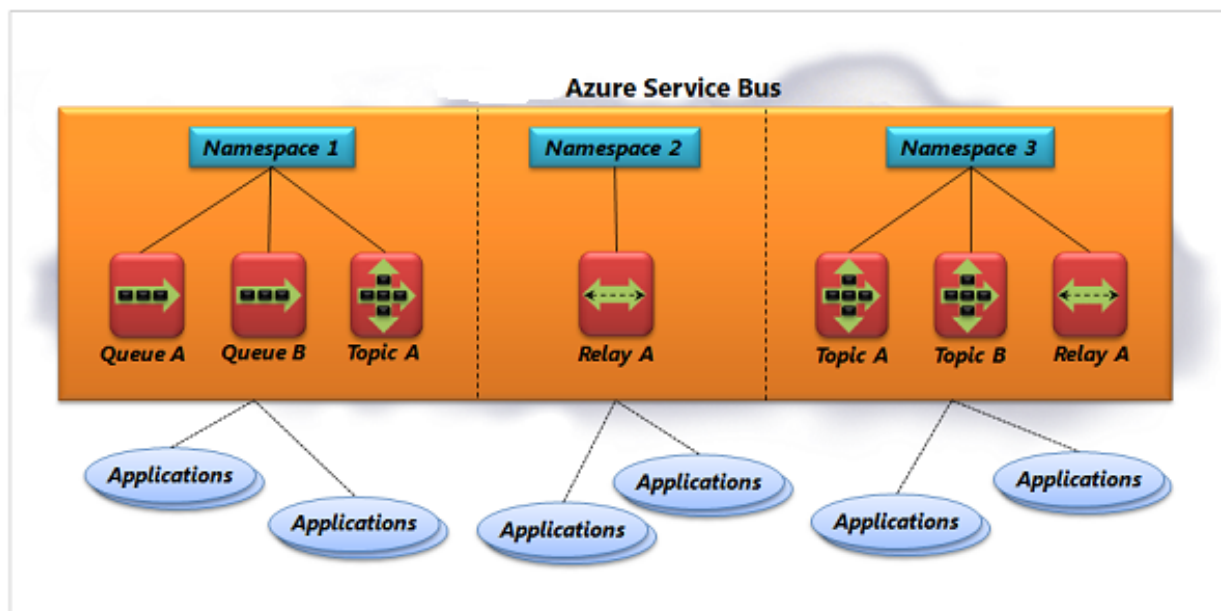


Рисунок 5.10- Службова шина надає мультітенантну службу для зв'язку Додатків через хмару

## 5.6 Збереження даних в Microsoft SQL Azure

У кожній CRM системі виникає потреба у реляційній базі даних та Microsoft Azure дає таку можливість. SQL Azure представляє собою реляційну базу даних, як послугу на базі ядра MS SQL, проте більш модифіковану та високопродуктивну, надійну та безпечну базу даних за допомогою якої можна створювати додатки у веб-сайти на основі даних, використовуючи об'єктно-орієнтоване програмування, без потреби управління інфраструктурою.

Вона підтримує такі структури, як реляційні дані, JSON, просторові дані та XML. Керовані бази даних SQL в еластичному пулі і керовані екземпляри бази даних SQL Вона забезпечує динамічно масштабовану продуктивність і надає певні можливості, індекси columnstore для екстремального аналітичного аналізу та звітності і виконувати в пам'яті OLTP для екстремальної обробки транзакцій. Майкрософт швидко і ефективно обробляє виправлення та оновлення бази коду SQL, а також спрощує обслуговування базової інфраструктури.[6]

Індекси Columnstore – це стандарт для збереження та запиту великих об'ємів даних у таблицях фактів. Збереження даних та обробка запитів по індексам дозволяє до 10 разів збільшити продуктивність запитів до сховища даних по відношенню з традиційним сховищем, у якому дані зберігаються по строкам та зменшити розміри даних до 10 разів порівняно з нестиснутими даними. Починаючи з кінця 2016 року, індекси columnstore дозволяють отримати оперативну аналітику та аналізувати робоче навантаження по обробці транзакцій у режимі реального часу. Columnstore index – це технологія збереження даних, отримання даних здійснюється за допомогою збереження даних в один стовбець, який називається columnstore.

База даних SQL використовує свою базу коду спільно з ядром СУБД Microsoft SQL Server. Новітні можливості SQL Server спочатку випускаються для бази даних SQL Azure, а потім для самої платформи SQL Server. Такий

підхід дозволяє використовувати новітні можливості SQL Server без зайвих витрат на внесення виправлень або оновлень. При цьому ці функції перевірені на мільйонах баз даних. Додаткові відомості про нові можливості ви можете дізнатися за допомогою наступних джерел:

База даних SQL надає передбачувану продуктивність на кількох рівнях обслуговування, що забезпечує динамічне масштабування без простою, вбудовану інтелектуальну оптимізацію, глобальні масштабованість і доступність, а також розширені параметри безпеки. Все це без необхідності адміністрування. Ці можливості дозволяють вам не витратити дорогоцінний час і ресурси на управління віртуальними машинами та інфраструктурою, а зосередитися на швидкій розробці додатків і прискорити вихід на ринок. Служба бази даних SQL розміщується в 38 центрах обробки даних по всьому світу. До більшості баз даних можна підключитися в будь-який час, що дозволяє запускати базу даних в прилеглому центрі обробки даних.

При використанні SQL Azure, кожна база даних ізольована одна від одної і переноситься - з власним рівнем обслуговування та гарантованим рівнем продуктивності. База даних SQL надає різні рівні продуктивності для різних потреб, а також дозволяє поміщати бази даних в пул для оптимізації використання ресурсів і економії коштів.

При використанні Керованого примірника бази даних SQL кожен з примірників ізольований від іншого і має гарантовані ресурси [8].

У моделі придбання на основі DTU набір обчислювальних операцій, пам'яті і ресурсів введення-виведення надається на трьох рівнях обслуговування: "Базовий", "Стандартний" і "Преміум". Кожен рівень передбачає підтримку різних робочих навантажень баз даних. Для кожного рівня продуктивності на рівнях обслуговування передбачено окреме поєднання цих ресурсів, до яких можна додати ресурси сховища.

Модель придбання на основі віртуального ядра дозволяє вибрати число віртуальних ядер, об'єм пам'яті, а також обсяг і швидкість сховища.

Ви можете створити свій перший додаток на основі невеликої однієї бази даних за оптимальною ціною, а потім змінити рівень обслуговування вручну або програмним способом в будь-який час відповідно до вимог вашого рішення. Ви можете налаштувати продуктивність без простою для програми та роботи клієнтів. Динамічна масштабованість дозволяє базі даних прозоро реагувати на швидко мінливі вимоги до ресурсів. Крім того, таким чином ви можете платити тільки за необхідні ресурси(рис 5.11), і тільки коли вони вам потрібні.

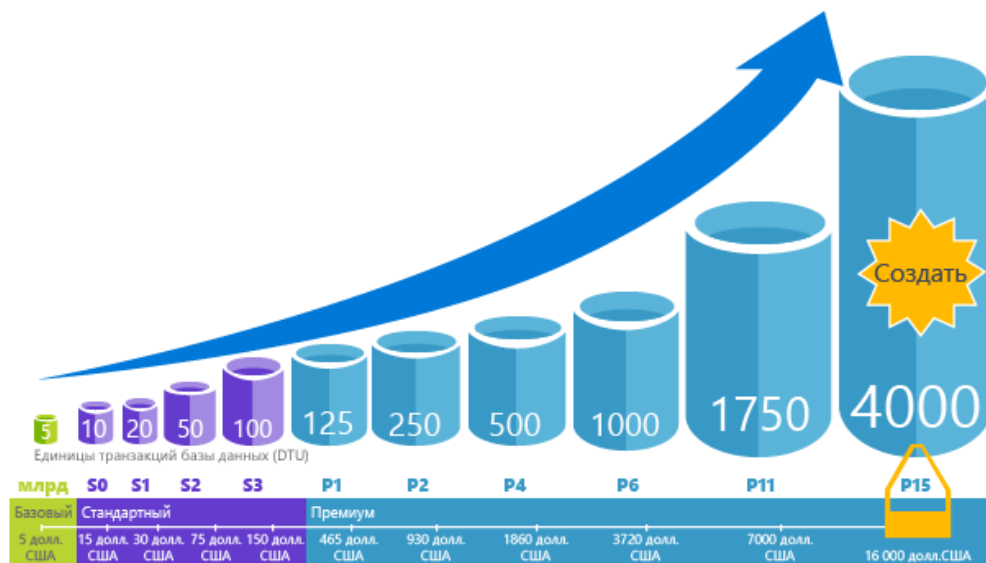


Рисунок 5.11 Обрання категорій баз даних.

Керований екземпляр бази даних SQL знаходиться в попередньої версії і пропонує єдиний рівень обслуговування. Еластичні пули для максимального використання ресурсів

Для багатьох організацій і додатків досить мати можливість створювати окремі бази даних і зменшувати або збільшувати їх продуктивність за запитом, особливо, якщо закономірності використання бази даних передбачувані. Але якщо закономірності використання непередбачувані, це може ускладнити управління витратами і бізнес-моделлю. Еластичні пули призначені для вирішення цієї проблеми. Ідея проста. Ви виділяєте ресурси для підвищення продуктивності пулу, а не окрему базу даних, і оплачуєте ресурси загальної продуктивності пулу, а не продуктивність окремих баз даних.(рис 5.12)





Рисунок 5.12 – Еластичні пули

Завдяки використанню еластичних пулів не потрібно концентруватися на підвищенні або зниженні продуктивності бази даних відповідно до коливань потреб в ресурсах. Бази даних в складі пулу споживають ресурси продуктивності пулу еластичних баз даних по мірі необхідності. Бази даних в складі пулу використовують ресурси пулу, але не перевищують обмежень пулу, тому ваші витрати залишаються прогнозованими, навіть якщо продуктивність окремої бази даних прогнозувати складно. Більш того, можна додавати бази даних в пул і видаляти їх з нього, масштабуючи додаток так, щоб кількість використовуваних баз даних становило від декількох екземплярів до багатьох тисяч, не виходячи за рамки бюджету. Ви також можете керувати мінімальним і максимальним обсягом ресурсів, доступних для баз даних в пулі. Таким чином жодна база даних в пулі буде споживати всі ресурси пулу, і в кожній з цих баз даних буде гарантований мінімальний обсяг ресурсів. Додаткові відомості про шаблони розробки для додатків SaaS, що використовують еластичні пули.

Незалежно від того, що ви вибрали - окремі бази даних або пули еластичних баз даних, - ваші можливості не обмежуються. Ви можете змішувати окремі бази даних з пулами еластичних баз даних і змінювати рівні обслуговування окремих баз даних та пулів еластичних баз даних, що дозволяє легко і швидко адаптувати їх під конкретні завдання. Завдяки можливостям Azure ви зможете поєднувати служби Azure з базами даних SQL, щоб задовольнити унікальні потреби архітектури додатків, підвищити

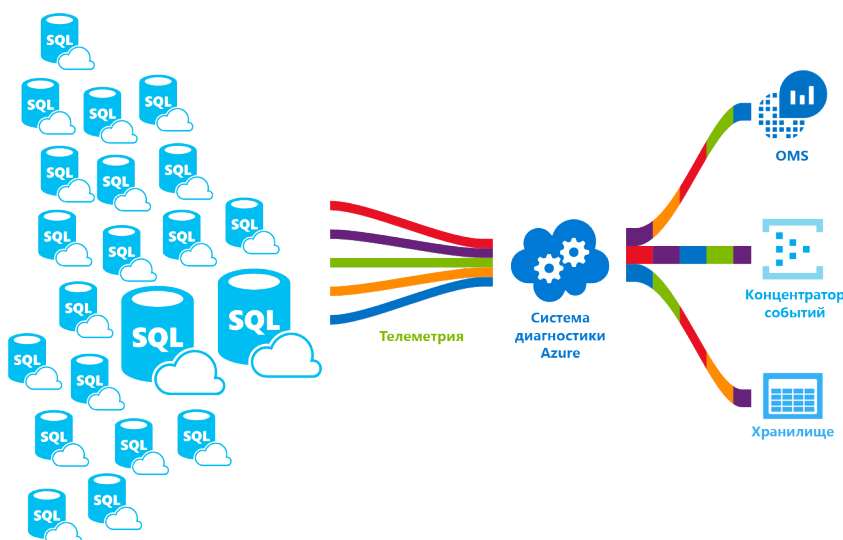
ефективність використання ресурсів і знизити витрати, а також дізнатися про нові можливості для розвитку бізнесу[7].

Але як порівняти відносну продуктивність окремих баз даних та пулів еластичних баз даних? Як дізнатися, де зупинитися при збільшенні або зменшенні масштабу бази даних? Ви можете використовувати вбудовані засоби моніторингу продуктивності та оповіщення в поєднанні з рейтингами продуктивності. Використовуючи ці засоби, ви зможете швидко оцінити ефект від збільшення і зменшення масштабу, виходячи з поточної або запланованої завантаження.

Крім того, база даних SQL може видавати значення метрик і журнали діагностики для спрощення моніторингу. Ви можете налаштувати базу даних SQL для зберігання відомостей про використання ресурсів, про робітників ролях і сеансах, а також налаштувати підключення до одного з цих ресурсів Azure: Служба: для архівації великих обсягів телеметрії за оптимальною вартістю.

Концентратор подій Azure: для інтеграції телеметрії бази даних SQL з налаштованим рішенням для моніторингу або гарячими конвеєрами.

Azure Log Analytics: для вбудованого рішення для моніторингу з можливостями надання звітів, попереджень і виконання виправлень. Log Analytics входить в пакет Operations Management Suite (OMS). (рис5.13)



### Рисунок 5.13 – Аналітика баз даних

Угода про рівень обслуговування (SLA) в Azure, що передбачає найвищий в галузі рівень доступності (99,99%) і глобальну мережу центрів обробки даних під керуванням Майкрософт, забезпечить безперервну роботу програми - 24 години у земний день і 7 днів на одну неділю. Крім того, база даних SQL забезпечує вбудовані функції безперервності бізнес-процесів і глобальної масштабованості, зокрема:

Автоматичне створення резервних копій: база даних SQL автоматично створює повні, різницеві копії і резервні копії журналів транзакцій.

Відновлення до певної точки в часі: база даних SQL підтримує відновлення на будь-який момент часу в межах автоматичного періоду утримання резервної копії.

Активна геореплікація: база даних SQL дозволяє налаштувати до чотирьох доступних для читання баз даних-одержувачів в одному або глобально розподілених центрах обробки даних Azure. Наприклад, при наявності додатка SaaS з базою даних каталогу, що містить великий обсяг паралельних транзакцій, доступних тільки для читання, необхідно використовувати активну геореплікацію, щоб включити глобальний масштаб читання і видалити вузькі місця в базі даних-джерелі, що виникли через робочих навантажень зчитування. Активна геореплікація забезпечує відпрацювання відмови однієї бази даних.

Групи відпрацювання відмови: база даних SQL дозволяє забезпечити глобальні високий рівень доступності та балансування навантаження, включаючи прозору геореплікацію і відпрацювання відмови великих наборів баз даних і еластичних пулів. Групи відпрацювання відмови і активна геореплікація дозволяють створювати глобально розподілені додатки SaaS з мінімальними витратами на адміністрування, залишаючи моніторинг, маршрутизацію і оркестрації відпрацювання відмови базі даних SQL. Групи

відпрацювання відмови дозволяють виконувати відпрацювання відмови декількох баз даних в групі.[5]

Бази даних, надлишкові в межах зони. База даних SQL дозволяє підготувати бази даних і еластичні пули рівня "Преміум" або "Критично важливий для бізнесу" в кількох зонах доступності. Так як ці бази даних або еластичні пули мають кілька надлишкових реплік для забезпечення високого рівня доступності, розміщення цих реплік в декількох зонах доступності гарантує більш високу стійкість, включаючи можливість автоматичного відновлення після масштабних збоїв центру обробки даних без втрати даних.

## 5.7 Збереження даних в Microsoft Azure Cosmos DB

Сьогодні існує досить багато NoSQL баз даних і без них створення сучасних CRM систем неможливе. Новий виток, це створення Azure Cosmos DB. Ця база даних являє собою багатомодульну та глобальну розподілену службу баз даних. Використовується Cosmos DB для розробки баз даних, які містять документи чи пари ключ – значення, баз даних Wide Column Store та графові бази даних та робиться все це за допомогою API-інтерфейсів та моделей програмування. Також вона дозволяє реплікувати дані улюбій кількості по всім дата центрам Microsoft Azure та масштабувати пропускну здатність незалежно від сховища. Таким чином головною особливістю цієї бази є можливість швидко масштабувати дані улюбій кількості, щоб зробити затримку мінімальною. Досить важливим фактором є простота завдяки API – інтерфейсам завдяки яким просто для масштабування треба обрати найближчі регіони, що у подальшому дозволяє йому відправляти дані у найближчі центри обробки даних и все це можна зробити навіть без зміни конфігурації. Все що треба зробити, це обрати регіон запису та скільки регіонів зчитування, а все інше Azure зробить самостійно[10]. Також по мірі додавання регіонів до бази даних Azure та видалення з неї, не потрібно розгортати додаток повторно.

Cosmos DB створена по принципу моделі даних на основі ARS. До не впроваджена підтримка декілька моделей даних, включаючи моделі даних документів, пар ключ-значення, графів, таблиць та сімейств стовбців. API-інтерфейси підтримують пакети SDK, доступні на декілька мовах, для слідуючих моделей даних

- API SQL – тобто ядро СУБД з підтримкою безсхемного формату JSON і широким можливостями відправки SQL-запитів API
- MongoDB - рішення MongoDB як послуга з високою масштабованістю на базі платформи Azure Cosmos DB. Сумісно з існуючими бібліотеками, драйверами, засобами і додатками MongoDB.
- API Cassandra - розподілене рішення "Cassandra як послуга" на базі платформи Azure Cosmos DB. Сумісно з існуючими бібліотеками, драйверами, засобами і додатками Apache Cassandra.
- API Gremlin (Graph) - повністю керована, горизонтально масштабована служба бази даних графа, яка спрощує створення і запуск додатків, що працюють з часто підключаються наборами даних, і підтримує інтерфейси Open Graph API (на основі специфікації Apache TinkerPop, Apache Gremlin).
- API таблиці - служба бази даних пар "ключ-значення", створена для надання можливостей рівня "Преміум" (наприклад, автоматичного індексування, гарантовано низькою затримки і глобального розподілу) існуючим програмам сховища таблиць Azure без зміни додатків.
- Найближчим часом буде реалізована підтримка додаткових моделей даних і інтерфейсів API.

Треба зазначити що Cosmos DB має гнучке і незалежне масштабування пропускної здатності. Azure Cosmos DB гарантує користувачам низьку наскрізну затримку на рівні 99-го процентиля. Має затримку менше ніж 5мс, а доступність на рівні 99,99% відповідно до угоди про рівень обслуговування для всіх облікових записів бази даних в межах одного регіону і доступність для всіх операцій читання на рівні 99,999% для всіх облікових записів бази даних в межах декількох регіонів [10]. Ядро СУБД Azure Cosmos DB взагалі не залежить від схем. Воно автоматично індексує всі прийняті дані, не потребуючи в схемах і індексах і моментально виконуючи запити. А також врахуємо, що вартість дешевше ніж у AWS DynamoDB або Google Spanner.

## **5.8 Секціонування в Azure Cosmos DB**

У цьому розділі представлені відомості про роботу секціонування для всіх моделей даних в Azure Cosmos DB. У ньому також описується налагодження контейнерів Azure Cosmos DB для ефективного масштабування додатків[10].

У базі даних Azure Cosmos DB можна зберігати дані без схеми даних, отримуючи відповідь із затримкою в мілісекундах незалежно від масштабу. Azure Cosmos DB надає контейнери для зберігання даних, які називаються колекціями, графами або таблицями.

Контейнери - це логічні ресурси, вони можуть включати в себе одну або кілька фізичних секцій або серверів. Число секцій визначається Azure Cosmos DB за розміром сховища і пропускної здатності, виділеної для контейнера.

Фізична секція - це фіксована кількість зарезервованого сховища на основі SSD. Кожна фізична секція реплікується для забезпечення високого рівня доступності. Одна або кілька фізичних секцій складають контейнер. Управління фізичними секціями повністю здійснюється Azure Cosmos DB і немає необхідності створювати складний код або керувати секціями. Контейнери Azure Cosmos DB не обмежені в плані зберігання та пропускної здатності [10].

Логічна секція - це секція в межах фізичної пам'яті, яка зберігає всі дані, пов'язані зі значенням ключа однієї секції. Кілька логічних секцій можуть виявитися в одній фізичній секції. У схемі(рис 5.14) зображено контейнер з трьома логічними секціями. Кожна логічна секція зберігає дані для одного ключа-секції (LAX, AMS і MEL відповідно). Кожна з цих секцій не може перевищувати максимальну межу - 10 ГБ.

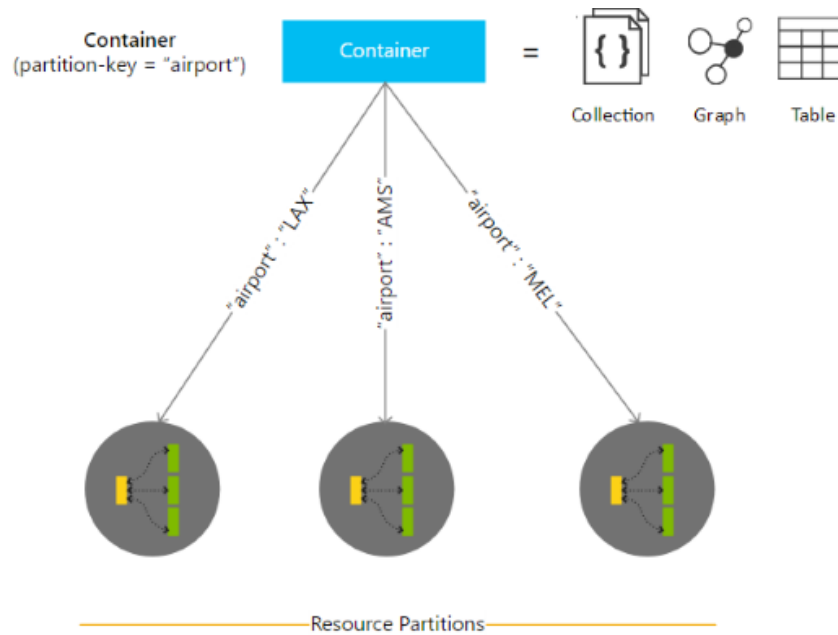


Рисунок 5.14 –Секціювання Cosmos DB

Якщо контейнер відповідає вимогам секціонування, секціонування буде повністю прозоре для застосування. Azure Cosmos DB підтримує високу швидкість читання і запису, запити, логіку транзакції, рівні узгодженості і точне управління доступом за допомогою методів і API для одного ресурсу контейнера[10]. Служба обробляє розподіл даних в фізичних і логічних секціях, а також направляє запити в потрібну секцію.

## 5.9 Робота з Azure Cosmos DB

Для початку роботи з Azure Cosmos DB Створемо колекцію, для цього розглянемо етапи роботи.

### 1. Клацніть Оглядач даних потім Нова колекція. (рис5.15)

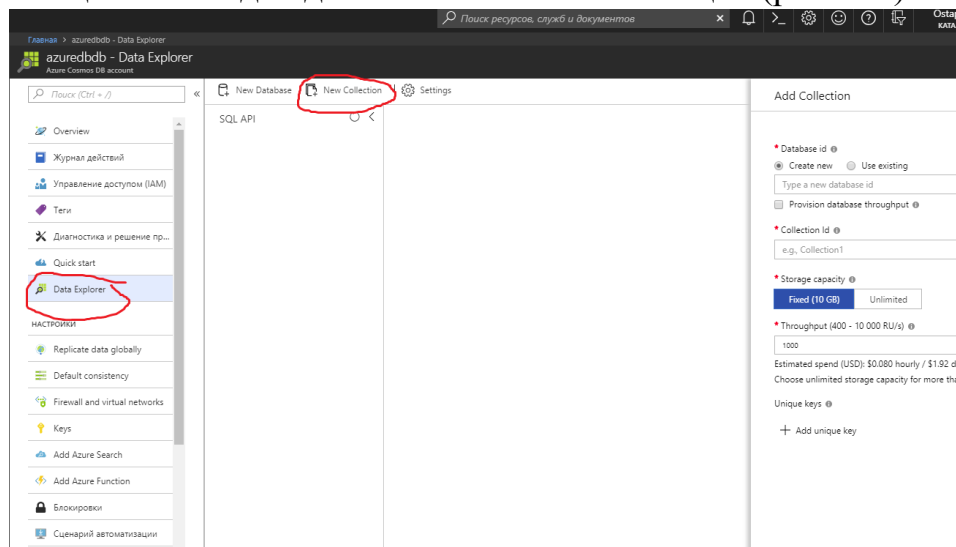


Рисунок 5.15– початок роботи з Cosmos DB

### 2. На сторінці Додавання колекції введіть параметри для нової колекції (рис 5.16)

Рисунок 5.16 -ведення параметрів для Колекції

В параметрах я обрав ємність сховища 10 гігабайт та найменшу пропускну здатність у 400 одиниць запитів у секунду. Після чого створилася колекція (рис 5.17)



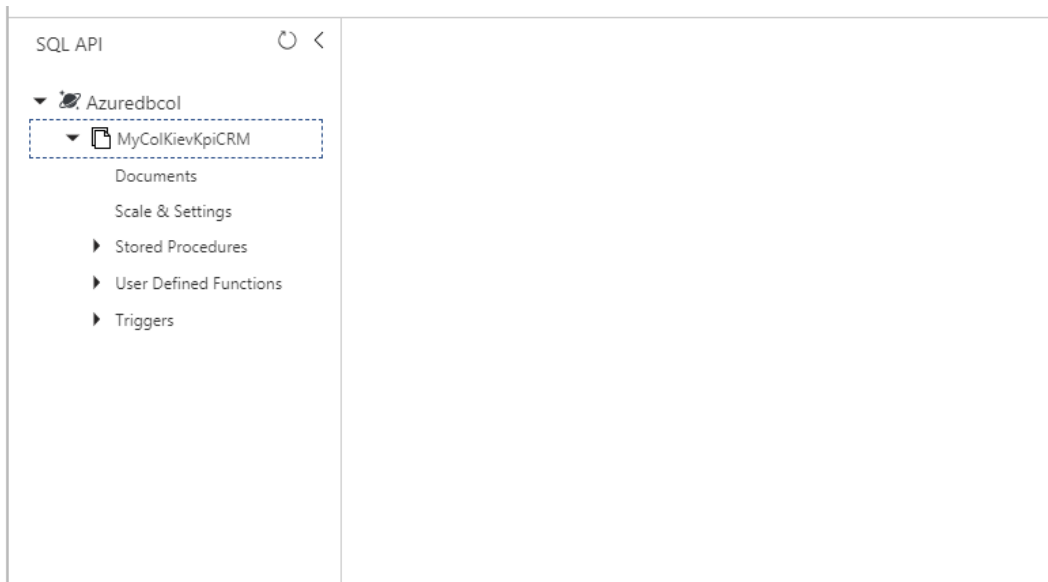


Рисунок 5.17 Створення колекції Cosmos DB

3. Тепер додаємо дані до колекції у форматі JSON та обираємо зберегти (рис1.23)

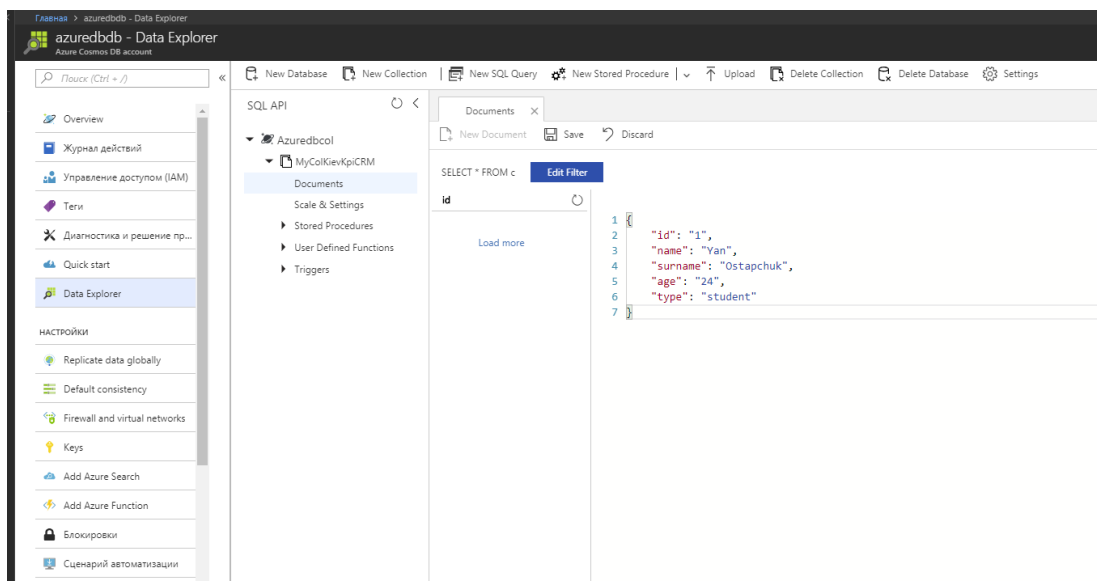


Рисунок 5.18 Додання до колекції даних

4. Обираємо зберегти і тепер ми маємо ці дані у базі даних.

## 5.10 Розробка власного сервісу з використання Storage Blobs

Раніше для збереження великих даних застосовувалися реляційні бази даних або збереження у файловій системі, ці рішення були потужним та залишаються до сих пір. Проте на сьогодні прийшло нове рішення у вигляді сховища даних Azure Storage, який має революційну модель збереження даних у вигляді Blobs. Задача, яку ми вирішимо буде збереження особистих даних клієнта у хмарі, таким чином ми покращимо дійсне рішення від Бітрікс24. Основна модель збереження даних в Azure Storage, це економія пам'яті завдяки видаленню нульових байтів. Для початку розглянемо можливості Azure storage у цілому

### 5.10.1 Опис можливостей Azure Storage у цілому

Кожна CRM потребує збереження даних та швидких операцій з ними і коли дані великі та даними треба обмінюватися між сервісами, на допомогу стає Azure Storage. Це служба сховища Microsoft Azure є хмарною службою під управлінням корпорації Майкрософт, яка надає сховище з високим рівнем доступності, безпеки, надійності, масштабованості і надмірності [9]. Служба сховища Azure складається зі сховища BLOB-об'єктів, таблиць і сховища черг. Це рішення дозволяє здійснити сучасний сценарій збереження даних. У службі сховища Microsoft Azure надається сховище з високою масштабованістю для об'єктів даних, служба файлової системи для хмари, сховище повідомлень для надійного обміну повідомленнями, а також сховище NoSQL. Служба сховища Azure відрізняється наступними перевагами:

- Надійність і висока доступність. Надмірність забезпечує безпеку даних при тимчасових збоях обладнання. Також для додаткового захисту від локальної аварії або стихійного лиха можна виконати

реплікацію даних в інший центр обробки даних або інший географічний регіон. Таким чином можна забезпечити високу доступність реплікованих даних в разі несподіваного збою.

- **Безпека.** Служба сховища Azure шифрує всі дані, що записуються в сховище. У службі сховища Azure надається можливість точного управління доступом до даних.
- **Масштабованість.** Служба сховища Azure розроблена з високим ступенем масштабованості відповідно до вимог до продуктивності і зберігання даних сучасних додатків.
- **Керованість.** Microsoft Azure відповідає за обслуговування і вирішує критичні проблеми, не вимагаючи вашої участі.
- **Доступність.** Доступ до даних в службі сховища Azure можна отримати з будь-якої точки світу по протоколу HTTP або HTTPS. Корпорація Майкрософт надає пакети SDK для служби сховища Azure на різних мовах (NET, Java, Node.js, Python, PHP, Ruby, Go та інші), а також готовий до використання REST API. Служба сховища Azure підтримує написання скриптів в Azure PowerShell і Azure CLI. А портал Azure та Провідник с

### 5.10.2 Опис можливостей Azure Blobs

Зберігання Azure Blob - це рішення для зберігання об'єктів у хмарі. Зберігання Blob оптимізовано для зберігання величезної кількості неструктурованих даних, таких як текстові або бінарні дані.

Зберігання Blob ідеально підходить для:

- Обслуговування зображень або документів безпосередньо в браузері.
- Зберігання файлів для розподіленого доступу.

- Потокове відео та аудіо.
- Введення в журнал файлів.
- Зберігання даних для резервного копіювання та відновлення, аварійного відновлення та архівування.
- Зберігання даних для аналізу за допомогою локальної або Azure-сервісної служби.

Об'єкти в сховище Blob можна отримати з будь-якої точки світу через HTTP або HTTPS. Користувачі або клієнтські програми можуть отримувати доступ до блоків через URL-адреси, API Azure Storage REST, Azure PowerShell, Azure CLI або бібліотеку клієнта Azure Storage. Клієнтські бібліотеки зберігання доступні для декількох мов, включаючи .NET, Java, Node.js, Python, PHP та Ruby.

Зберігання Blob надає три ресурси: ваш обліковий запис для зберігання, контейнери в обліковому записі та розбиття на блоки в контейнері. Наступна діаграма показує взаємозв'язок між цими ресурсами. (рис 5.19)

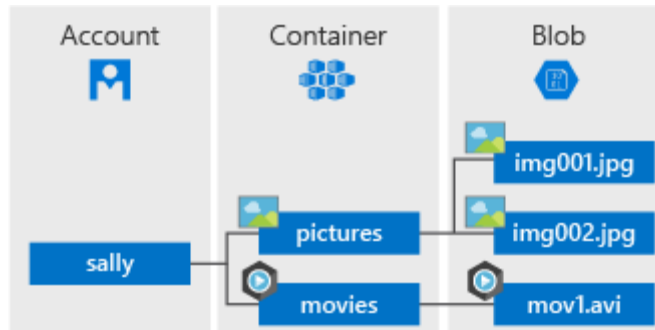


Рисунок 5.19 Зберігання у блобах

Весь доступ до об'єктів даних в Azure Storage, відбувається через обліковий запис пам'яті.

Контейнер організовує набір блоків, подібних до дерикторій у файловій системі, файли перебувають у контейнерах. Обліковий запис може містити необмежену кількість контейнерів, а контейнер може зберігати необмежену кількість блоків чи сторінок. Зауважте, що ім'я контейнера має бути нижнім регістром.

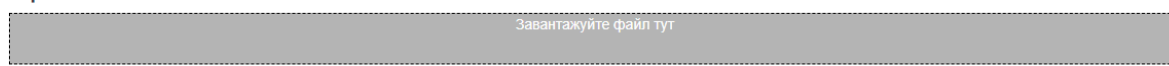
Azure Storage пропонує два типи блобів - блок блоб, блоб сторінки ,які використовуються для збереження файлів VHD. Блок блоб зберігає текстові та двійкові дані, до приблизно 4,7 ТБ. Блоки блобів складаються з блоків даних, якими можна керувати індивідуально. Блоби сторінок зберігають файли довільного доступу розміром до 8 ТБ. Блоби сторінок зберігають файли VHD, які повертаються VM [9].

Всі Блоби перебувають у контейнері. Контейнер схожий на папку у файлової системі. Можна додатково організувати використання Блобів в віртуальних каталогах, і переходити до них , як в файлової системі.

### 5.10.3 Реалізація на мові C# і технології ASP.NET MVC

Покажемо збереження дійсно великих даних , де на стенді платформи розділі архів чи диск відкриється цей сервіс(рис5.20). Даний сервіс дозволяє переносити файли drag & drop та зберігати їх в Azure Storage. Для початку створимо Azure Storage Account (рис 5.21)

Архів



Завантаженні файли:

© 2018 - Kiev Kpi CRM

Рисунок 5.20 – Сервіс по завантаженню файлів

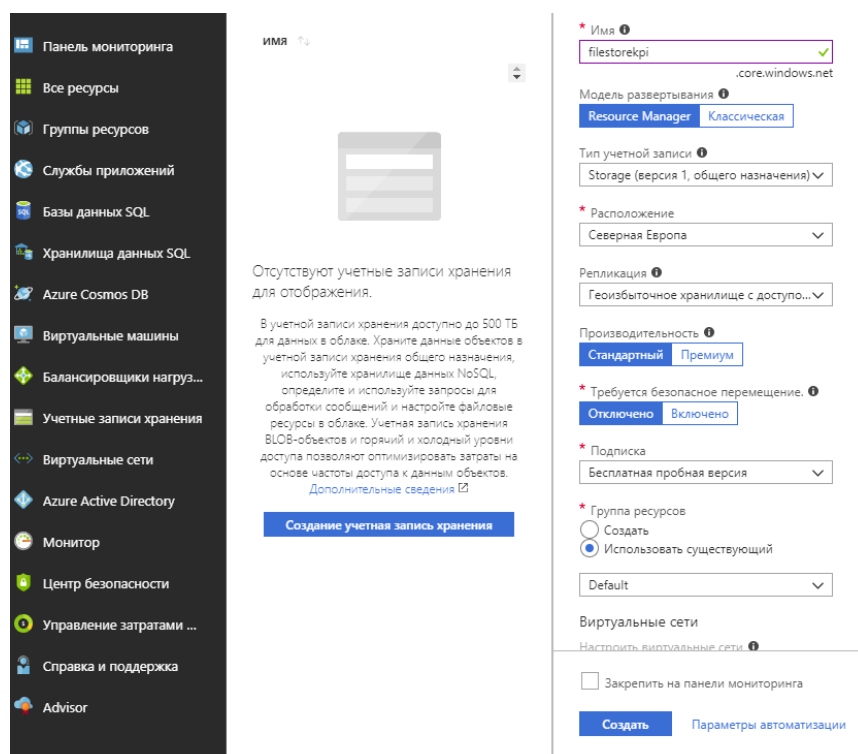


Рисунок 5.21- Створення Account Storage

Після чого відкриємо Visual Studio та скачаємо бібліотеку Azure Storage та введемо значення з облікової Storage.(рис 5.22)

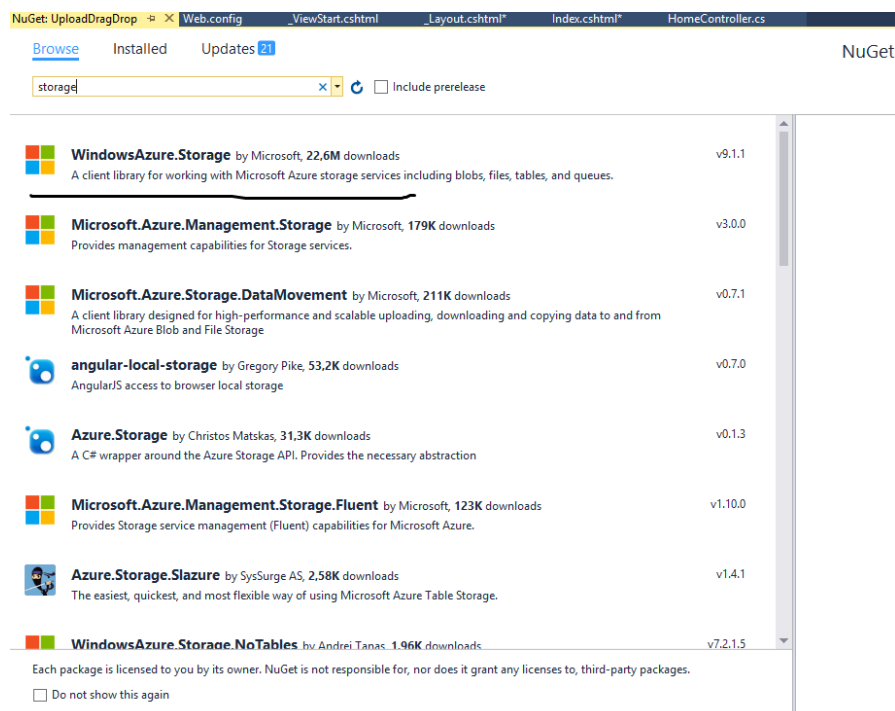


Рисунок 5.22 – Завантаження бібліотеки Azure Storage

на Рис 5.23 наведено код на мові С#, для створення об'єкту аккаунта для роботи з Azure Storage.

```
CloudStorageAccount account = new CloudStorageAccount(new StorageCredentials("name", " key"), false);
```

Рисунок 5.23 – Створення об'єкту аккаунт

Напишем permission для доступу з мережі рис 5.24

```
BlobContainerPermissions permissions = new BlobContainerPermissions
{
    PublicAccess = BlobContainerPublicAccessType.Container
};
CloudBlobClient client = account.CreateCloudBlobClient();
CloudBlobContainer container = client.GetContainerReference("kpi");
container.CreateIfNotExists();
container.SetPermissions(permissions);
```

Рисунок 5.24- надання дозволу

після чого

збережемо наші файли рис 5.25

```
CloudBlockBlob blob = container.GetBlockBlobReference(filePath);
var length = file.ContentLength;
var buffer = new byte[length];

using (var stream = new MemoryStream(file.InputStream.Read(buffer, 0, length)))
{
    blob.UploadFromStream(stream);
}
```

Рисунок 5.25 збереження файлів

Таким чином ми завдяки сховищу даних Azure Storage зберегли наші дані та зекономили місце на сервері.

## 5.11 Висновок

В даному розділі був розглянутий інструментарій для прототипування CRM рішення. Зокрема був розроблений повноцінний сервіс на базі Azure Storage, з використання Blobs, що дає можливість зекономити пам'ять. Головною особливістю було використання інструментарію від компанії Microsoft та дослідження його можливостей. Зокрема була проаналізована система платформа PaaS розгляд її переваг та особливостей.

Було описано такі технології, як Microsoft Azure Load Balancer, його розгляд. Були показані нові методи збереження даних, які нам надає Azure зокрема SQL Azure його переваги та нові модифікації. Azure Cosmos DB де розповідалось, як її створити та, які інтерфейси вона має та може себе вести, як будь-яка база даних. Було розглянуто сховище Azure Storage та його підрозділи такі як, Tables, Queues, Blobs, коли краще, які застосовувати та, як вони себе поведуть. Було показано, як створюється сервіси за допомогою Azure Cloud Service та його окремі ролі такі як, web-role та worker-role. Як налаштувати файл конфігурації та опублікувати свій сервіс. А також була розглянута Шина Service Bus.



## 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ “CRM як сервіс”

Розділ має на меті проведення маркетингового аналізу стартап проекту “CRM як сервіс” задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту “ CRM як сервіс” в умовах висококонкурентної ринкової економіки глобалізаційних процесів.

### 6.1 Опис ідеї стартапу

Опис стартап-проекту “CRM як сервіс” наведено у Таблиці 6.1.

Таблиця 6.1. Опис ідеї стартап-проекту

| <i>Зміст ідеї</i>   | <i>Напрямки застосування</i>   | <i>Вигоди для користувача</i>  |
|---|--|--|
| Створення та розгортання CRM платформи у хмарі з наданням користувачеві доступу читання та запису через REST api. Надання можливості виконувати базові транзакції , спілкуватися між собою виконувати статистичні розрахунки ,а також вносити дані про клієнта та зберігати | 1. Використання для виконання аналізу даних, та оброблення інформації про користувача                          | REST API надає можливість використовувати систему як сервіс, що значно спрощує доступ до сховища.          |
|   | 2. Використання сховища Microsoft Azure Storage у інших додатках як спосіб збереження сильно пов'язаних даних. | Можливість виконання використання хмарних сховищ дає суттєву перевагу порівняно С простим збереженням в БД |

|                             |  |  |
|-----------------------------|--|--|
| важливі документи у архіві. |  |  |
|-----------------------------|--|--|

Отже, проект “CRM як сервіс” може бути використаним як інструментом для деякого аналізу даних, так і прошарком постійного збереження сильно пов’язаних даних у платформі Microsoft Azure Storage.

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| №<br>п/<br>п | Техніко-<br>економічні<br>характерис-<br>тики ідеї   | (потенційні) товари/концепції<br>конкурентів |                 |                 |                 | W<br>(слабка<br>сторон<br>а) | N<br>(нейтр<br>а- льна<br>сторон<br>а) | S<br>(сильна<br>сторон<br>а) |
|--------------|--|--|-----------------|-----------------|-----------------|------------------------------|--|------------------------------|
|              |  | Мій<br>проект                                | Конкур<br>ент 1 | Конкур<br>ент2  | Конкур<br>ент 3 |                              |  |                              |
| 1.           | Форма<br>виконання                                   | Платфо<br>рма з<br>сервіса<br>ми             | Програ<br>ма    | Веб-<br>додаток | Програ<br>ма    |                              |  | +                            |
| 2.           | Собівартість   | Середн<br>я                                  | Середн<br>я     | Висока          | Висока          |                              | +                                      |                              |
| 3.           | Кроссплатформні<br>сть                               | Ні   | Ні              | Так             | Так             | +                            |  |                              |
| 4.           | Наявність<br>Розміщення у<br>хмарі                   | Так  | Ні              | Ні              | Ні              |                              |  | +                            |
| 5.           | Наявність<br>сховища Azure<br>storage або<br>аналога | Так  | Ні              | Ні              | Так             |                              |  | +                            |
| 6.           | Застосування<br>логічного<br>виведення               | Так/Ні                                       | Ні              | Так             | Так             |                              | +                                      |                              |
| 7.           | Горизонтальне<br>масштабування                       | Так  | Так             | Ні              | Так             |                              |  | +                            |

Сильними сторонами проекту є форма виконання у вигляді Платформи з веб-сервісами, розміщення у хмарі , Наявність Microsoft Azure Storage для оновлення даних, застосування логічного виведення. Слабкою стороною є відсутність кроссплатформності при розробці, нейтральною - наявність застосування логічного виводу та середня собівартість для зчитування даних. Отож, система є конкурентноспроможною.

## 6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

| <i>№ n/n</i> | <i>Ідея проекту</i>                          | <i>Технології її реалізації</i> | <i>Наявність технологій</i> | <i>Доступність технологій</i> |
|--------------|--|---------------------------------|-----------------------------|-------------------------------|
| 1.           | Створення сховища Microsoft Azure Storage    | C#                              | Наявна                      | Безкоштовна, доступна         |
|              |  | Microsoft Azure                 | Наявна                      | Платна                        |
| 2.           | Створення REST API для доступу до бази знань | Asp.net MVC                     | Наявні                      | Безкоштовна, доступна         |
|              |  | Microsoft Azure                 | Наявна                      | Платні                        |
| 3.           | Хмарне розгортання додатку                   | Visual Studio Community         | Наявна                      | Безкоштовна, доступна         |
|              |  | Microsoft Azure                 | Наявна                      | Платні                        |

Обрані технології реалізації ідеї проекту: Asp.net MVC через повну безкоштовність фреймворку та наявність докладної документації, наявність досвіду роботи розробників з даною технологією; Visual Studio, C# через простоту використання, безкоштовність та можливість розгортання додатків на основі таких технологій у хмарі; Azure для розгортання у хмарі через великий функціонал.

## 6.3. Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап-проекту

| <i>№ n/n</i> | <i>Показники стану ринку (найменування)</i>              | <i>Характеристика</i>                         |
|--------------|--|---|
| 1.           | Кількість головних гравців, од                           | 5   |
| 2.           | Загальний обсяг продаж, грн/ум.од                        | 300грн./місяць                                |
| 3.           | Динаміка ринку (якісна оцінка)                           | Зростає                                       |
| 4.           | Наявність обмежень для входу (вказати характер обмежень) | Немає   |
| 5.           | Специфічні вимоги до стандартизації та сертифікації      | Немає   |
| 6.           | Середня норма рентабельності в галузі (або по ринку), %  | $R = (3000000 * 100) / (1000000 * 12) = 25\%$ |

Отже, було проаналізовано наявність попиту, обсяг, динаміку розвитку ринку. Обмеження для входу на ринок відсутні, динаміка ринку зростає, галузь є рентабельною.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 6.5).

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проекту

| <i>№ n/n</i> | <i>Потреба, що формує ринок</i>   | <i>Цільова аудиторія (цільові сегменти ринку)</i>  | <i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i> | <i>Вимоги споживачів до товару</i>  |
|--------------|---|--|--|---|
| 1.           | Необхідне програмне забезпечення (REST API) для доступу до бази знань як до сервісу | Потенційними цільовими групами є дослідницькі центри, ресторани та компанії, специфіка роботи яких потребує взаємовідносин з клієнтами | Цільова група займається дослідженнями або здійснює торгові операції     | Рішення повинне бути придатним до інтеграції з іншими сервісами, бути здатним надавати перелік характеристик товару, мати швидкий доступ до даних |

Згідно проведеної характеристики потенційних клієнтів стартап-проекту впливає, що на ринку є затребуваним використання хмарних CRM як до сервісу і потенційними цільовими групами є трасти, торгові мережі та компанії, специфіка роботи яких потребує взаємовідносин з клієнтами.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. № 6-7). Фактори в таблиці подавати в порядку зменшення значущості.

Таблиця 6.6 – Фактори загроз

| <i>№<br/>n/n</i> | <i>Фактор</i>   | <i>Зміст загрози</i>   | <i>Можлива реакція компанії</i>  |
|------------------|---|--|--|
| 1.               | Конкуренція   | Вихід на ринок великої компанії                                    | 1. Вихід з ринку<br>2. Запропонувати великій компанії поглинути себе<br>3. Передбачити додаткові переваги власного ПЗ для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок |
| 2.               | Зміна потреб користувачів                               | Користувачам необхідне програмне забезпечення з іншим функціоналом | 1. Передбачити можливість додавання нового функціоналу до створюваного ПЗ  |
| 3.               | Зміна тарифів провайдера хмарного розгортання на платні | Необхідність оплати послуг провайдера хмари                        | 1. Пошук іншого безкоштовного провайдера<br>2. Пошук інвестицій для оплати існуючого провайдера  |
| 4.               | Надходження на ринок альтернативних продуктів           | Перехід користувачів нашого товару на інший продукт                | Впровадження нового функціоналу, якого немає у конкурентів   |
| 5.               | Уповільнення  | Скорочення користувачів  | Інвестиції у впровадження  |

|  |             |  |                             |
|--|-------------|--|-----------------------------|
|  | росту ринку | продуктів, що тільки виходять на ринок | ефективної реклами продукту |
|--|-------------|--|-----------------------------|

Отже, було проаналізовано фактори загроз ринкового впровадження проекту, серед яких: конкуренція, уповільнення росту ринку, зміна потреб користувачів, зміна тарифів провайдера хмарного розгортання на платні та надходження на ринок альтернативних продуктів. Було також запропоновано можливі реакції компанії.

Таблиця 6.7 – Фактори можливостей

| <i>№ n/n</i> | <i>Фактор</i>   | <i>Зміст можливості</i>  | <i>Можлива реакція компанії</i>   |
|--------------|---|--|---|
| 1.           | Стрімкий ріст попиту на взаємовідносини роботи з клієнтом | Наявність попиту на хмарні CRM системи   | Змога запропонувати продукт більшої кількості потенційних користувачів  |
| 2.           | Поява нових ризонерів                                     | Надання нового функціоналу для надання результатів роботи нового логічного виведення | Розробка нового функціоналу у вигляді нового HTTP запиту для надання користувачам результатів логічного виведення |
| 3.           | Стрімке зростання росту ринку                             | Компаніям, що тільки виходять на ринок, буде простіше отримати клієнтів              | Змога запропонувати продукт більшої кількості потенційних користувачів  |
| 4.           | Обслуговування додаткових груп споживачів                 | Поява нових потенційних груп споживачів  | Змога розширити продукт для подальшого впровадження у нові галузі   |
| 5.           | Розширення асортименту можливих послуг                    | Поява нового функціоналу, що привабить нових користувачів                            | Розробка нового функціоналу, що є потребою певної групи користувачів  |

У Таблиці 6.7 наведено фактори можливостей ринкового впровадження проекту, серед яких: стрімкий ріст попиту на інструменти обробки даних, поява нових ризонерів, стрімке зростання росту ринку, обслуговування додаткових груп споживачів, розширення асортименту можливих послуг, було також запропоновано можливі реакції компанії.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 6.8).

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

| <i>Особливості конкурентного середовища</i>        | <i>В чому проявляється дана характеристика</i>                        | <i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>   |
|--|---|---|
| 1. Вказати тип конкуренції - досконала             | Існує 3 фірми- конкуренти на ринку                                    | Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами) |
| 2. За рівнем конкурентної боротьби - міжнародний   | Дві компаній – з іншої країни, одна – з України                       | Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок   |
| 3. За галузевою ознакою - внутрішньогалузева       | Конкуренти мають ПЗ, яке використовується лише всередині даної галузі | Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях                                |
| 4. Конкуренція за видами товарів: - товарно-видова | Види товарів є однаковими, а саме – програмне забезпечення            | Створити ПЗ, враховуючи недоліки конкурентів  |
| 5. За характером конкурентних переваг - нецінова   | Вдосконалення технології створення ПЗ, щоб собівартість була нижчою   | Використання менш дорогих технологій для розробки, ніж використовують конкуренти  |
| 6. За інтенсивністю - не марочна                   | Бренди відсутні   | -   |



У Таблиці 6.8 наведено ступеневий аналіз конкуренції на ринку, де було визначено особливості конкурентного середовища та їх вплив а діяльність підприємства. Однією з найбільш важливих дій компанії для досягнення конкурентоспроможності є необхідність створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (Табл. 6.9).

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

| <i>Складові аналізу</i> | <i>Прямі конкуренти в галузі</i>  | <i>Потенційні конкуренти</i>   | <i>Постачальники</i>                         | <i>Клієнти</i>                                 | <i>Товари-замінники</i>  |
|-------------------------|---|--|--|--|--|
|                         | <i>Навести перелік прямих конкурентів</i>   | <i>Визначити бар'єри входження в ринок</i>   | <i>Визначити фактори сили постачальників</i> | <i>Визначити фактори сили споживачів</i>       | <i>Фактори загроз з боку замінників</i>  |
| <b>Висновки:</b>        | Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 3, так як його рішення також має сховище даних | Так, можливості для входу на ринок є, бо наше рішення дуже залежить від хмарних ресурсів | Постачальники відсутні.                      | Важливим для користувача є швидкість роботи ПЗ | Товари-замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару |

Було здійснено аналіз конкуренції в галузі за М. Портером, в результаті чого було визначено, що існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 3, так як його рішення також має сховище даних, але можливості для входу на ринок є, бо наше рішення розміщене у хмарі та має добру мобільність.

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. Також

робиться висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку. Другий висновок враховується при формулюванні переліку факторів конкурентоспроможності у п. 3.6. 3.6) На основі аналізу конкуренції, проведеного в п. 3.5 (табл. 9), а також із урахуванням характеристик ідеї проекту (табл. 2), вимог споживачів до товару (табл. 5) та факторів маркетингового середовища (табл. № 6-7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 10

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

| <i>№ n/n</i> | <i>Фактор конкурентоспроможності</i>        | <i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i> |
|--------------|---|--|
| 1.           | Наявність хмарі Microsoft Azure або аналога | Дозволяє користувачам здійснювати гнучкий доступ до ресурсів   |
| 2.           | Можливість завантажувати великі файли       | Висока швидкість заповнення сховища  |
| 3.           | Наявність REST API                          | Дозволяє інтегрувати сервіс у складні системи завдяки універсальному API                                   |
| 4.           | Хмарне розгортання                          | Дозволяє звертатись до бази знань як до сервісу  |
| 5.           | Горизонтальне масштабування                 | Можливість гнучкого масштабування за допомогою додавання апаратних компонентів                             |

У Таблиці 7 наведено обґрунтування факторів конкурентоспроможності, серед яких: наявність хмари, можливість завантажувати цілі онтології, наявність REST API та хмарне розгортання. Було також наведено обґрунтування цих факторів.

За визначеними факторами конкурентоспроможності (табл. 6.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 6.11).

У наступній таблиці наведено проведення аналізу сильних та слабких сторін стартап-проекту, факторами конкурентоспроможності виступили такі: наявність хмарної платформи для зчитування та оновлення даних, можливість завантажувати великих файлів, наявність REST API, хмарне розгортання, горизонтальне масштабування.

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін проекту

| № п/п | Фактор конкурентоспроможності               | Бали 1-20 | Рейтинг товарів-конкурентів у порівнянні з нашим підприємством |    |    |   |   |   |   |
|-------|---|-----------|--|----|----|---|---|---|---|
|       |   |           | -3   | -2 | -1 | 0 | 1 | 2 | 3 |
| 1.    | Наявність хмарі Microsoft Azure або аналога | 20        |  |    |    | + |   |   |   |
| 2.    | Можливість завантажувати великі файли       | 20        |  |    | +  |   |   |   |   |
| 3.    | Наявність REST API                          | 15        |  |    | +  |   |   |   |   |
| 4.    | Хмарне розгортання                          | 15        |  | +  |    |   |   |   |   |
| 5.    | Горизонтальне масштабування                 | 10        |  |    |    |   | + |   |   |

Отже, серед сильних сторін проекту можна виділити наступні: розміщення CRM у хмарі, можливість завантажувати великі бінарні файли, наявність REST API, можливість хмарного розгортання та горизонтальне масштабування. Серед слабких сторін можна виділити залежність від провайдеру, а саме від хмарних ресурсів.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого

можна зробити 103 прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

У наступній таблиці буде проілюстровано SWOT-аналіз стартап-проекту, тобто його слабкі та сильні сторони, можливості та загрози виходу на ринок.

Таблиця 6.12. SWOT- аналіз стартап-проекту

|  |   |
|--|---|
| Сильні сторони: наявність розміщення у хмарі для зчитування та оновлення даних, можливість завантажувати цілі онтології, наявність REST API, можливість додавання безлічі сервісів та можливість горизонтального масштабування | Слабкі сторони: можливість зміни тарифів провайдером хмарного розгортання на платні, відсутня незалежність від провайдера послуг  |
| Можливості: стрімкий ріст попиту на системи взаємодії з клієнтом, можливість впровадження нових рішень, стрімке зростання ринку, обслуговування додаткових груп споживачів, розширення асортименту можливих послуг             | Загрози: конкуренція, зміна потреб користувачів, зміна тарифів провайдера хмарного розгортання на платні, надходження на ринок альтернативних продуктів, уповільнення росту ринку |

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

| № п/п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|-------|--|--------------------------------|-------------------|
| 1.    | Створення додатку з використанням фреймворків                  | 90%                            | 5 місяців         |

|    |  |     |           |
|----|--|-----|-----------|
|    | Visual Studio, IIS ,Azure, Asp.net                             |     |           |
| 2. | Створення платформи на без використання будь- яких фреймворків | 10% | 24 місяці |

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проекту

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими. Тому обираємо альтернативу (створення додатку з використанням фреймворків Visual Studio ,ASP.NET MVC).

#### 6.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 14).

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

| <i>№<br/>п/п</i> | <i>Опис профілю<br/>цільової групи<br/>потенційних<br/>клієнтів</i> | <i>Готовність<br/>споживачів<br/>сприйняти продукт</i> | <i>Орієнтовний<br/>попит в<br/>межах<br/>цільової<br/>групи<br/>(сегменту)</i> | <i>Інтенсивність<br/>конкуренції в<br/>сегменті</i>                        | <i>Простота<br/>входу у<br/>сегмент</i>                |
|------------------|---|--|--|--|--|
| 1.               | Дослідницькі центри   | Спрощення роботи з високо пов'язаними даними           | Великий  | Існує 1 конкуренти, який надають схожі, але більш вузькі і дорогі рішення. | Наявність REST API, відомої хмари, логічного виведення |
| 2.               | Підприємства  | Спрощення роботи з великою кількістю клієнтів          | Великий  |  | Можливість інтеграції в уже існуючі системи            |

|  |  |  |  |  |   |
|--|--|--|--|--|---|
|  |  |  |  |  | завдяки REST API, зручне хмарне розгортання |
| Які цільові групи обрано: обираємо підприємства та дослідницькі центри |  |  |  |  |   |

Таблиця 6.14 – Вибір цільових груп потенційних споживачів

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку. Для роботи в обраних сегментах ринку необхідно сформувані базову стратегію розвитку. За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Отже, проілюструвати базову стратегію розвитку можна у вигляді Таблиці 6.15

Таблиця 6.15 – Визначення базової стратегії розвитку

| <i>№<br/>п/п</i> | <i>Обрана<br/>альтернатива<br/>розвитку проекту</i>                              | <i>Стратегія охоплення<br/>ринку</i> | <i>Ключові<br/>конкурентоспромо<br/>жні позиції<br/>відповідно до обраної<br/>альтернативи</i> | <i>Базова стратегія<br/>розвитку</i> |
|------------------|--|--------------------------------------|--|--------------------------------------|
| 1.               | Створення веб-сервісу, використовуючи Azure, IIS server, SQL server, Asp.net MVC | Ринкове позиціонування               | Можливість інтеграції в уже існуючі системи завдяки REST API, зручне хмарне розгортання,       | Диференціація                        |

Було обрано таку альтернативу розвитку проекту: створення веб-сервісу, використовуючи ASP.NET , Microsoft Azure, адже завдяки цим технологіям можна досягнути ключових конкурентоспроможних позицій кінцевого продукту.

Таблиця 6.16 - Визначення базової стратегії конкурентної поведінки

| <i>№ n/n</i> | <i>Чи є проект «першопрохідцем» на ринку?</i> | <i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i> | <i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>           | <i>Стратегія конкурентної поведінки</i> |
|--------------|---|---|--|---|
| 1.           | Так   | Так   | Буде, а саме: основною задачею є розробка ПЗ з використанням сховища Microsoft Azure Storage | Зайняття конкурентної ніші              |

Отже, було визначено базову стратегію конкурентної поведінки як зайняття конкурентної ніші.

Визначимо стратегію позиціонування у Таблиці 6.17, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 6.17 - Визначення стратегії позиціонування

| <i>№ n/n</i> | <i>Вимоги до товару цільової аудиторії</i>   | <i>Базова стратегія розвитку</i> | <i>Ключові конкурентоспроможні позиції власного стартап- проекту</i>   | <i>Вибір асоціацій, які мають сформувавши комплексну позицію власного проекту (три ключових)</i> |
|--------------|--|----------------------------------|--|--|
| 1.           | Наявність універсального API, зручне хмарне розгортання, наявність логічного виведення | Диференціація                    | Можливість інтеграції в уже існуючі системи завдяки REST API, зручне хмарне розгортання, наявність логічного виведення | Інтеграція, хмарне розгортання, хмарне сховище   |

Отже, було вибрано такі асоціації, які мають сформувавши комплексну позицію власного проекту: інтеграція (адже завдяки REST API сервіс просто інтегрувати у існуючі системи), хмарне розгортання (оскільки ASP.NET додатки легко розгортаються в хмарах), можливості горизонтального розгортання.

## 6.5 Розробка маркетингової програми

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 5.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 6.18 - Визначення ключових переваг концепції потенційного товару

| <i>№ n/n</i> | <i>Потреба</i>                           | <i>Вигода, яку пропонує товар</i>  | <i>Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)</i> |
|--------------|--|--|---|
| 1.           | Наявність універсального API             | Додаток реалізований у вигляді RESTful сервісу, що надає відповіді у вигляді JSON, що дає змогу користувачам звертатись до сервісу за допомогою стандартних HTTP POST та GET запитів | Перевага в універсальності на можливості інтегрувати сервіс у існуючі системи.      |
| 2.           | Можливість зручного хмарного розгортання | Можливість розгорнути додаток всюди, де є функціонал розгортання ASP.NET додатків, а така можливість є у хмарах які підтримують Windows  | Користувачі мають змогу працювати з системою віддалено у хмарі                      |
| 3.           | Наявність сховища даних                  | Можливість виконання будь-яких запитів, та горизонтальне масштабування   | Підтримка стандарту нових технологій  |
| 4.           | Наявність логічного виведення            | Виведення даних у форматі xml що здійснюється завдяки SOAP   | Можливість виведення XML  |

Отже бачимо, що проект має ключові переваги перед конкурентами, які повністю відповідають потребам цільової аудиторії. Додаток реалізований у вигляді RESTful сервісу, що надає відповіді у вигляді JSON, що дає змогу користувачам звертатись до сервісу за допомогою стандартних HTTP POST та GET запитів, а це є досить універсальним способом для подальшої інтеграції сервісу в інші системи.



Далі у Таблиці 6.19 проілюстрована трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Таблиця 6.19 - Опис трьох рівнів моделі товару

| <i>Рівні товару</i>  | <i>Сутність та складові</i>   |                              |  |
|--|---|------------------------------|--|
| I. Товар за задумом  | Веб-сервіс, що надає доступ до сховища Microsoft Azure , дозволяє працювати багатьма іншими сервісами та надає можливості логічного виведення та хмарного розгортання |                              |  |
| II. Товар у реальному виконанні  | Властивості/характеристики  | М/Нм                         | Вр/Тх /Тл/Е/Ор   |
|  | 1. Наявність універсального API<br>2. Можливість зручного хмарного розгортання<br>3. Наявність Azure Storage<br>4. Наявність логічного виведення                      | 1.Нм<br>2.Нм<br>3.Нм<br>4.Нм | 1.Технологічна<br>2.Технологічна<br>3.Технологічна<br>4.Технологічна |
|  | Якість: згідно до стандарту ISO 4444 буде проведено тестування  |                              |  |
|  | Маркування відсутнє   |                              |  |
|  | Моя компанія: “Kiev Kpi Crm ”   |                              |  |
| III. Товар із підкріпленням  | 1-місячна пробна безкоштовна версія   |                              |  |
|  | Постійна підтримка для користувачів   |                              |  |
| За рахунок чого потенційний товар буде захищено від копіювання: патент |   |                              |  |

Було описано три рівні моделі товару, з чого можна зробити висновок, що основні властивості товару у реальному виконанні є нематеріальними та технологічними. Також було надано сутність та складові товару у задумці та товару з підкріпленням.

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. У даному випадку найбільш вірогідним гарантом буде патент.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 5.20). Аналіз проводиться експертним методом.

Таблиця 6.20 - Визначення меж встановлення ціни

| <i>№ п/п</i> | <i>Рівень цін на товари-замінники, грн/міс</i> | <i>Рівень цін на товари-аналоги, грн.міс.</i> | <i>Рівень доходів цільової групи споживачів, грн.</i> | <i>Верхня та нижня межі встановлення ціни на товар/послугу, грн.</i> |
|--------------|--|---|---|--|
| 1.           | 2000   | 2650  | 100000  | 6240-500   |

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 5.21).

Таблиця 6.21 - Формування системи збуту

| <i>№ п/п</i> | <i>Специфіка закупівельної поведінки цільових клієнтів</i>               | <i>Функції збуту, які має виконувати постачальник товару</i> | <i>Глибина каналу збуту</i>              | <i>Оптимальна система збуту</i> |
|--------------|--|--|--|---------------------------------|
| 1.           | Придбання підписки та оплата щомісячних внесків для продовження ліцензії | Продаж   | 0(напрямую), 1(через одного посередника) | Власна та через посередників    |

Отже, система приносить прибуток завдяки щомісячним внескам для продовження ліцензії та придбанням підписок, продаж будк виконуватись напряму або через одного посередника.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 5.22).

Таблиця 6.22 - Концепція маркетингових комунікацій

| <i>№ n/n</i> | <i>Специфіка поведінки цільових клієнтів</i>   | <i>Канали комунікацій, якими користуються я цільові клієнти</i> | <i>Ключові позиції, обрані для позиціонування</i> | <i>Завдання рекламного повідомлення</i>                      | <i>Концепція рекламного звернення</i>                                 |
|--------------|--|---|---|--|---|
| 1.           | Придбання ліцензії на користування в мережі Інтернет, щомісячне її продовження, користування сервісом у хмарі або ж на власних серверах. | Інтернет  | Інтеграція , хмарне розгортання,                  | Показати переваги сервісу, у тому числі і перед конкурентами | Демо-ролик із використання, рекламні оголошення на популярних сайтах. |

Отже, в Таблиці 6.22 наведено концепцію маркетингових комунікацій, було визначено, що придбання ліцензії на користування буде здійснюватись в мережі Інтернет, необхідним буде щомісячне її продовження, користування сервісом можливе у хмарі або ж на власних серверах.

## 6.6 Висновок

Згідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також, варто відмітити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, а проект має дві значні переваги перед конкурентами. Для успішного виконання проекту необхідно реалізувати програму із використанням засобів Microsoft Azure, ASP.NET MVC, Azure storage. Для успішного виходу на ринок у продукту повинні бути наступні характеристики:

- наявність універсального API
- можливість зручного хмарного розгортання
- зручне та ефективне збереження даних
- наявність логічного виведення

В рамках даного дослідження були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

Було визначено такі сильні сторони: наявність хмари для збереження та оновлення даних, можливість завантажувати великі файли, наявність REST API, можливість хмарного розгортання. Серед слабких сторін можна виділити можливість зміни тарифів провайдером хмарного розгортання.

Можливості для виходу на ринок включають стрімкий ріст попиту на інструменти взаємодії з клієнтами у вигляді CRM систем у хмарі, можливість впровадження нових технологій та приєднання нових інноваційних технологій, стрімке зростання росту ринку, обслуговування додаткових груп споживачів, розширення асортименту можливих послуг. Наявні такі фактори загроз: конкуренція, зміна потреб користувачів, зміна тарифів провайдера хмарного розгортання на платні, надходження на ринок альтернативних продуктів, уповільнення росту ринку.

## Висновки

У дипломному проекті були розглянуті усі доступні методи та інструменти для створення сервісів для CRM рішень, а також розроблений стенд для взаємодії між ними, який був розгорнутий у хмарі Microsoft Azure на декількох віртуальних машинах з використанням нових іноваційних підходів.

За основу взято дійсне CRM рішення та на основі функціоналу його сервісів, покращити його роботу та на основі іноваційних рішень впровадити це у реальність, розроблено два альтернативних підходи. Перший був реалізований завдяки технологіям від Microsoft Azure із застосуванням технологій, які надає хмара таких, як Azure Storage, Load Balancer, SQL Azure, Cosmos DB та Service Bus. А другий по прототипу першого, проте з використанням готових сервісів від FiWare та своїх власних з використанням технологій ASP.NET Core на відміну від ASP.NET MVC, який був використаний у першому випадку та за допомогою застосувань Entity Core на відміну від Entity Framework .NET, також другий випадок нам дозволив використати Docker контейнеризацію так, як в першому випадку використовувався Azure App Service Fabric та Kubernetes.

Основний внесок роботі полягає в тому, що було дослідження іноваційний підхід по покращанню даних CRM рішень завдяки прототипуванню та використанню сервісів від FiWare, а подруге були застосовані новітні методи збереження даних та їх реалізація. Що дало змогу підвищити рівень маркетингу, продажів, комерції, обслуговування клієнтів і соціальної взаємодії між співробітниками компаній і клієнтами.

Також був розроблений стартап проект завдяки, якому були оглянуті основні ризики, аналіз конкурентів та можливість виходу на ринок.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Scott Kostojohn, Mathew Johnso and Brian Paulen CRM Fundamentals 2012 223 pages
2. Розгляд шини даних Microsoft Service Bus  
<https://azure.microsoft.com/en-us/services/service-bus/>
3. Kumar, V., Reinartz, Werner CRM 2018 404 pages
4. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, (2004). Semantic markup for web services. , 2007–04.
5. Joel Scott Michael Dellisa Microsoft CRM 2009 pages 323
6. Sriram Krishnan Windows Azure O'Reilly 2010 337 pages
7. Огляд хмарної платформи Microsoft Azure  
[https://ru.wikipedia.org/wiki/Microsoft\\_Azure](https://ru.wikipedia.org/wiki/Microsoft_Azure)
8. Опис хмарних сервісів та роботи з ними  
[https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing)
9. Застосування Azure Storage <https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>
10. Застосування бази даних Azure Cosmos DB  
<https://azure.microsoft.com/ru-ru/services/cosmos-db/>
11. Остапчук Я.М. Стратегія розвитку CRM систем як сервісу у хмарі Microsoft Azure / Ян Остапчук.// Матеріали XX всеукраїнської науково – практичної студентської конференції, 21 травня 2018, Київ, Україна: матеріали. – К. : НТУУ «КПІ», 2018. – С. 184-185.
12. Вибір проекту у системі Microsoft Azure  
<https://azure.microsoft.com/ru-ru/solutions/>
13. Scott Klein, Herve Roggero Pro SQL Database for Windows Azure 2013 283 pages
14. Cloud Computing Principles and Paradigms Rajkumar Byya, James Broberg, Andrzej Goscinski 2010 542 pages

15. Cloud Computing: Concepts, Technology & Architecture Thomas Erl  
2013 528 pages
16. Mastering Zoho CRM Shabdar Ali 2017 328 pages
17. SugarCRM Building on John Martic 2012 325 pages
18. Microsoft Dynamics CRM Administration Mathew Witterman, Geoff  
Ables 2011 712 pages
19. Microsoft Dynamics by Marc Wolenik 2013 pages 1176 pages
20. The Guide for Business and Technology Managers by Vivek Kale  
2015 25 pages