

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ ДЛЯ МАЛОГО ПІДПРИЄМСТВА

Студента 2 курсу, 23 групи,
спеціальності 121«Інженерія
програмного забезпечення»,
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Гоголіна Віталія
Валерійовича

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Харченко Олександр
Анатолійович

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис гаранта

Токар Володимир
Володимирович

КИЇВ – 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки

Криворучко О. В.

"22" січня 2021 р.

Завдання

на випускний кваліфікаційний проєкт студентів

Гоголіну Віталію Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи: «Проектування архітектури хмарної CRM-системи для малого підприємства»

Затверджена наказом ректора від «21» січня 2021 р. № 211

2. Строк здачі студентом закінченої роботи 25 листопада 2021 р.

3. Цільова установка та вихідні дані до роботи

Мета роботи – розробка архітектури хмарної CRM-системи на основі аналізу бізнес-процесів малого підприємства

Об'єкт дослідження – теоретичні і практичні аспекти проектування хмарної CRM-системи

Предмет дослідження – архітектура хмарної CRM-системи для малого підприємства

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1			
2			
3			

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВПРОВАДЖЕННЯ CRM-СИСТЕМИ ДЛЯ МАЛОГО ПІДПРИЄМСТВА

1.1 Постановка проблеми

1.2 Поняття хмарної CRM-системи

1.3 Порівняльний аналіз CRM-систем

1.4 Архітектура типових хмарних CRM-систем

1.5 Висновки до розділу 1

РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ

2.1 Обґрунтування вибору програмних засобів для реалізації хмарної CRM-системи

2.1.1 Вимоги малого підприємства до автоматизованої системи управління взаємовідносинами з клієнтами

2.1.2 Вибір програмних та апаратних засобів для CRM-системи «Сервіт»

2.2 Класифікація завдань хмарної CRM-системи для малого підприємства «Сервіт»

2.3 Проектування архітектури хмарної CRM-системи

2.4 Висновки до розділу 2

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ

3.1 Розробка баз даних, знань, системи управління операціями хмарної CRM-системи малого підприємства «Сервіт»

3.1.1 Мова запитів SQL як основа обробки даних в БД

3.1.2 Object-Relational-Mapping як рішення для відображення бізнес об'єктів

3.2 Визначення складу та структури бази даних

3.3 Програмна реалізація хмарної CRM-системи та опис програмних модулів

3.4 Захист CRM-системи як об'єкта права інтелектуальної власності

3.5 Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	
		за планом	фактично
1	2	3	4
1	Вибір теми випускного кваліфікаційного	21.01.2021	21.01.2021
2	Розробка та затвердження завдання на проект магістра	22.01.2021	22.01.2021
3	Вступ та перелік літературних джерел	27.02.2021	27.02.2021
4	Розробка технічного завдання	20.03.2021	20.03.2021
5	Розділ 1. Аналіз існуючих рішень для впровадження CRM-системи для малого підприємства	16.04.2021	16.04.2021
6	Написання наукової статті	22.05.2021	22.05.2021
7	Розділ 2. Розробка архітектури хмарної CRM-системи	24.05.2021	24.05.2021
8	Розділ 3. Програмна реалізація архітектури хмарної CRM-системи	21.06.2021	21.06.2021
9.	Розробка програми та методики тестування	18.10.2021	18.10.2021
10	Висновки та пропозиції	01.11.2021	25.10.2021
11	Здача випускного кваліфікаційного проекту на кафедру (перша перевірка)	03.11.2021	03.11.2021
12	Підготовка автореферату та презентації Доповіді	03.11.2021	03.11.2021
13	Попередній захист випускного кваліфікаційного проекту	22.11.2021 — 25.11.2021	24.11.2021
14	Здача зброшурованої випускного кваліфікаційного проекту	25.11.2021	25.11.2021
15	Зовнішнє рецензування випускного кваліфікаційного проекту	26.11.2021	26.11.2021
16	Підготовка до публічного захисту випускного кваліфікаційного проекту		

7. Дата видачі завдання «22» січня 2021 р.

8. Науковий керівник випускного кваліфікаційного проекту
Харченко О.А.

(прізвище, ініціали, підпис)

1. Гарант освітньої програми Токар В.В.
(прізвище, ініціали, підпис)

2. Завдання прийняв до виконання студент Гоголін В.В.
(прізвище, ініціали, підпис)

[illegible]

Відмітка про попередній захист Харченко О.А.
(ПІБ, підпис, дата)

Випускний кваліфікаційний проект студента Гоголіна В.В.
(прізвище, ініціали)

Гарант освітньої програми _____ Токар В.В.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О.В.
(підпис, прізвище, ініціали)

«_____» 20__ г.

АНОТАЦІЯ

Гоголін В.В. Проектування архітектури хмарної CRM-системи для малого підприємства. – На правах рукопису.

У магістерській роботі проаналізовано проблему побудови архітектури хмарної CRM-системи, уточнено сутність поняття хмарна CRM-система, визначено її структуру, вимоги яким вона має відповідати. У роботі визначено можливості хмарних сервісів для реалізації CRM-системи, досліджено інструментарій технічного та правового забезпечення безпеки та захисту хмарних CRM-системи. Особлива увага приділена виокремленню складових хмарної CRM-системи та їх логічним зв'язкам. Розглянуті програмні засоби побудови хмарної CRM-системи.

Ключові слова: CRM-система, хмарні технології, AWS сервіси, архітектура системи.

Табл. 2 Рис. 19 Джерел: 35 найм.

ANNOTATION

Gogolin V.V. Designing the architecture of a cloud CRM-system for a small business. - Qualifying work on the rights of the manuscript.

The master's thesis analyzes the problem of building the architecture of the cloud CRM-system, clarifies the essence of the concept of cloud CRM-system, defines its structure, the requirements to which it must meet. The paper identifies the capabilities of cloud services for the implementation of CRM-system, explores the tools of technical and legal security and protection of cloud CRM-systems. Particular attention is paid to the separation of components of the cloud CRM system and their logical connections. Software tools for building a cloud CRM-system are considered.

Keywords: CRM-system, cloud technologies, AWS services, system architecture.

Table. 2 Fig. 19 Sources: 35 items.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВПРОВАДЖЕННЯ CRM-СИСТЕМИ ДЛЯ МАЛОГО ПІДПРИЄМСТВА	7
1.1 Постановка проблеми.....	7
1.2 Поняття хмарної CRM-системи	8
1.3 Порівняльний аналіз CRM-систем	10
1.4 Архітектура типових хмарних CRM-систем.....	16
1.5 Висновки до розділу 1	19
РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ.....	21
2.1 Обґрунтування вибору програмних засобів для реалізації хмарної CRM-системи.....	21
2.1.1 Вимоги малого підприємства до автоматизованої системи управління взаємовідносинами з клієнтами	22
2.1.2 Вибір програмних та апаратних засобів для CRM-системи «Сервіт».....	25
2.2 Класифікація завдань хмарної CRM-системи для малого підприємства «Сервіт»	30
2.3 Проектування архітектури хмарної CRM-системи	32
2.4 Висновки до розділу 2	33
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ	35
3.1 Розробка баз даних, знань, системи управління операціями хмарної CRM-системи малого підприємства «Сервіт».....	35

					КНТЕУ 121– 023-01.МР			
					Проектування хмарної CRM-системи для малого підприємства	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			22.01.21		3	2	53
Керівник	Харченко О.А.			22.01.21				
Гарант	Токар В.В.			22.01.21				
Розроб	Гоголін В.В.			22.01.21	ЗМІСТ	Факультет інформаційних технологій, 2 курс, 2м група		

3.1.1 Мова запитів SQL як основа обробки даних в БД.....	36
3.1.2 Object-Relational-Mapping як рішення для відображення бізнес об'єктів..	38
3.2 Визначення складу та структури бази даних.....	39
3.3 Програмна реалізація хмарної CRM-системи та опис програмних модулів.	45
3.4 Захист CRM-системи як об'єкта права інтелектуальної власності.....	47
3.5 Висновки до розділу 3.....	49
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ	62

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпись	Дата		3

ВСТУП

Актуальність теми. Нагальним для сучасного підприємства є можливість планування завдань для подальшої автоматичної обробки даних, програмування унікальних функцій для оптимізації управління бізнес-процесами. Нині широкого вжитку набули стандартні типізовані CRM-системи на основі яких підприємства вимушені доопрацьовувати такі готові рішення з урахуванням особливостями ведення окремого бізнесу. Однак, для ефективної роботи конкурентоспроможного підприємства важливим є підвищення якості управління взаємовідносинами з клієнтами, а отже, розробка своєї CRM-системи з оптимальною архітектурою є пріоритетним завданням.

Аналіз наукових досліджень. Системи управління взаємовідносинами з клієнтами, їх завдання та функції розглядали у своїх працях вітчизняні науковці, зокрема М.Лянцевич, А. Марданов, Р. Мунасіпов, А. Погорелов, О. Ходек, та закордонні науковці Chen, I. J., Chen P.P., Popovich, K., King, S. F., Burgess, T. F., O'Reilly K. Paper D., Payne, A., Frow, P. Однак, питання проектування архітектури хмарної CRM-системи на знайшло достатнього висвітлення.

Об'єктом дослідження є теоретичні і практичні аспекти проектування хмарної CRM-системи.

Предметом дослідження є архітектура хмарної CRM-системи для малого підприємства.

Мета роботи: розробити архітектуру хмарної CRM-системи на основі аналізу бізнес-процесів малого підприємства. Для досягнення поставленої мети необхідно вирішити наступні **завдання:**

- дослідити поняття хмарної CRM-системи;

					<i>КНТЕУ 121–023-01.МР</i>			
					Проектування хмарної CRM-системи для малого підприємства	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		В	4	53
Зав. каф.		Криворучко О.В.		27.02.21				
Керівник		Харченко О.А.		27.02.21				
Гарант		Токар В.В.		27.02.21				
Розроб		Гоголін В.В.		27.02.21	ВСТУП	Факультет інформаційних технологій, 2 курс, 2м група		

- проаналізувати CRM-системи та надати їх класифікацію;
- побудувати архітектуру типової хмарної CRM-системи;
- з'ясувати вимоги та завдання хмарної CRM-системи для малого підприємства, які ставляться до сучасних хмарних CRM-систем;
- розробити архітектуру хмарної CRM-системи для малого підприємства та її програмно реалізувати.

Методи дослідження: при написанні рукопису використовувалися як загальнонаукові методи пізнання, зокрема порівняння, опису, формалізація, аналізу, синтезу, моделювання, абстрагування, узагальнення, так і спеціально-наукові методи пізнання: теорія моделювання складних процесів і систем, методологія структурного опису, теорії алгоритмів, побудови компонентів і систем.

Наукова новизна одержаних результатів визначається тим, що: рукопис є одним з перших теоретико-практичних досліджень розробки архітектури хмарної CRM-системи для малого підприємства, в межах якого запропоновані підходи щодо проектування архітектури хмарних CRM-систем, зокрема:

уперше:

1) запропоновано визначення хмарної CRM-системи – розташована в хмарних сервісах інформаційна система управління відносинами з клієнтами, функціонал якої надає можливості збору, зберігання й аналізу інформації про споживачів, постачальників, партнерів, а також забезпечення безпеки та цілісності даних.

2) запропонована архітектура хмарної CRM-системи для малого підприємства, основою якої стали компоненти системи та виконувані ними сервіси, які відповідають вимогам гнучкості, кросбраузерності та безпеки.

набуло подальшого розвитку:

3) підстави розмежування завдань хмарної CRM-системи, в основу яких покладено бізнес процеси системи.

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						5
Вим.	Аркуш	№ докум	Підпись	Дата		

4) наукова концепція безпеки хмарної CRM-системи яку запропоновано розглядати як цілісний об'єкт захисту, а також проаналізована можливість захисту окремих її складових.

удосконалено:

5) класифікація CRM-систем, що надало можливість оцінити раціональність використання тої чи іншої для спеціалізованого завдання підприємства, засоби захисту таких систем, способів звернення до неї та інструментарій обробки даних.

6) наукові ідеї та підходи про визначення гнучкості та її значення для хмарної CRM-системи, зокрема виділені такі її складові як інтеграція, сумісність, багатозадачність, стандартизація, модульність, масштабованість.

Практична цінність. Результати роботи використані в поточній діяльності малого підприємства «Сервіт», зокрема розроблена база даних, відповідно до документації підприємства; спроектована та програмно-реалізована архітектура хмарної CRM-системи підприємства.

Апробація роботи. Окремі результати магістерської роботи апробовані та оприлюднені на IV Міжнародній науково-практичній конференції «Підготовка фахівців у сфері інтелектуальної власності» (м. Київ, 14 травня 2021 року) публікацією тез доповідей на тему «CRM-система як об'єкт права інтелектуальної власності». Основні положення та результати кваліфікаційної магістерської роботи опубліковані у друкованій праці на тему «Класифікація CRM-систем» у збірнику наукових статей «Software engineering комп'ютерних систем» (Київ, 2021).

Структура та обсяг роботи. Магістерська робота складається з вступу, трьох розділів, списку використаних джерел та додатків. Загальний обсяг магістерської роботи складає 53 сторінки, список використаних джерел (35 найменувань) – 3 сторінки, додатки – 6 сторінок.

					<i>КНТЕУ 121–023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпись	Дата		6

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВПРОВАДЖЕННЯ CRM-СИСТЕМИ ДЛЯ МАЛОГО ПІДПРИЄМСТВА

1.1 Постановка проблеми

Завдання кваліфікаційної роботи полягає у розробці архітектури хмарної CRM-системи та її програмної реалізації з урахуванням моделі даних малого підприємства «Сервіт». Для досягнення поставленої мети необхідно дослідити особливості функціонування такого підприємства та визначити інструментарій цифрової оптимізації управління взаємовідносин з клієнтами:

- 1) розглянути архітектуру типових хмарних CRM-систем та з'ясувати вимоги до їх функціональної складової;
- 2) з'ясувати поняття хмарної CRM-системи;
- 3) надати класифікацію CRM-систем та визначити;
- 4) побудувати модель архітектури типової хмарної CRM-системи;
- 5) дослідити програмні засоби для реалізації хмарної CRM-системи та на основі аналізу зробити вибір програмних та апаратних засобів для програмної реалізації CRM-системи малого підприємства «Сервіт»;
- 6) розробити проект архітектури хмарної CRM-системи для малого підприємства;
- 7) виявити завдання хмарної CRM-системи малого підприємства «Сервіт», що окреслить правила бізнес логіки підприємства;

					<i>КНТЕУ 121– 023-01.МР</i>			
					<i>Проектування хмарної CRM-системи для малого підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>РІ</i>	<i>7</i>	<i>53</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>16.04.21</i>	<i>Аналіз існуючих рішень для впровадження CRM-системи для малого підприємства</i>	<i>Факультет інформаційних технологій, 2 курс, 2м група</i>		
<i>Керівник</i>		<i>Харченко О.А.</i>		<i>16.04.21</i>				
<i>Гарант</i>		<i>Токар В.В.</i>		<i>16.04.21</i>				
<i>Розроб</i>		<i>Гоголін В.В.</i>		<i>16.04.21</i>				

- 8) розробити проект архітектури хмарної CRM-системи малого підприємства «Сервіт»;
- 9) програмно реалізувати хмарну CRM-систему відповідно до її архітектурного проекту;
- 10) протестувати програмно реалізовану CRM-систему та на основі результатів тестування виправити недоліки системи;
- 11) дослідити можливості захисту хмарної CRM-системи.

1.2 Поняття хмарної CRM-системи

Завданням даного підрозділу є висвітлення наявних поглядів на поняття «CRM-система», їх аналіз з метою з'ясування, яке з них найвлучніше характеризує сутність досліджуваного явища.

Методами досягнення поставлених цілей є наступні: логічний метод, яким аргументується істинність чи хибність наукових суджень відносно CRM-систем, їх характеризуючих ознак та сутності; діалектичний метод надає можливість 1) розвинути уявлення про таке явище як CRM-система; 2) розкрити цілісність об'єкта та виявити суть CRM-системи, його внутрішні зв'язки, елементи; формально-догматичний метод забезпечує тлумачення окремих понять, термінів, категорій.

Поняття CRM-системи досліджувалося як зарубіжними так і вітчизняними науковцями. Так, Chen, I. J., & Popovich, K. визначають управління взаємовідносинами з клієнтами як інтеграцію процесів, людського капіталу та технологій з метою найкращого розуміння клієнтів компанії [8, с. 673]. На наше бачення, запропоноване визначення хоч і вказує на мету CRM-системи, проте не розкриває сутність, функціональну складову та інструментарій.

В свою чергу King, S. F. та Burgess T. F. розглядали CRM-систему як один з найпопулярніших інструментів для управління корпоративною інформацією для підвищення ефективності продаж, маркетингу та для більш тісної взаємодії з

					КНТЕУ 121–02з-01.МР	Аркуш
						8
Вим.	Аркуш	№ докум	Підпис	Дата		

клієнтами та управління даними про клієнтів [13, с. 423]. Таке визначення характеризує в цілому функціональну направленість CRM-системи.

На думку Reinartz W. та Krafft, M., CRM – це систематичний процес управління ініціюванням, обслуговуванням та припиненням відносин з клієнтами в усіх точках зв'язку з клієнтами для максимізації цінності інформації про ведення клієнта компанією [21, с. 297]. На нашу думку, запропоноване визначення CRM-системи як певної процедури, направленої на збагачення інформативної бази про клієнта не відображає повною мірою необхідний інструментарій, адже окрім збору інформації про клієнта надважливо здійснити операції з отриманими даними, які дозволять вибудувати стратегію бізнесу.

К. Шалендер зауважив, що ефективна CRM-система надає змогу вибудувати довготривалі відносини з клієнтами, пом'якшити вплив змін на бізнес процеси, та врешті-решт, зміцнити ринкові позиції підприємства завдяки укріпленню його бренду [22, с. 40].

На наш погляд найбільш вдало надано визначення CRM-системи Мозговою Г. В. та Морозовим А. О. Вони поклали в основу визначення концепцію інформативної спрямованості CRM-системи на вивчення ринку і конкретних потреб клієнтів: управління відносинами з клієнтами (CRM) – поняття, що охоплює концепції, які використовуються компаніями для управління їхніми взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними [31, с. 89].

Таким чином, ефективне управління відносинами з клієнтами можливе за наявності інструментарію по збору достатньої та актуальної інформації про клієнтів і партнерів, а також можливості її зберігання та обробки для підвищення ефективності роботи підприємства з кінцевим споживачем та обрання бізнес стратегії.

Оскільки дослідженню піддається саме CRM-система, інструментарій якої

					<i>КНТЕУ 121–02з-01.МР</i>	Аркуш
						9
Вим.	Аркуш	№ докум	Підпис	Дата		

знаходиться у хмарному сховищі, а доступ до неї відбувається потужностями хмарного сервісу, можна дійти висновку, що хмарна CRM-система – розташована в хмарних сервісах інформаційна система управління відносинами з клієнтами, функціонал якої надає можливості збору, зберігання й аналізу інформації про споживачів, постачальників, партнерів, а також забезпечення безпеки та цілісності даних.

1.3 Порівняльний аналіз CRM-систем

Метою даного підрозділу є з'ясування видових характеристик CRM-систем та на основі обраних критеріїв надання їх класифікації. Для виконання поставленої мети будуть застосовані наступні методи наукового пізнання: діалектичний метод – для визначення місця хмарної CRM-системи серед їх видового розмаїття; метод аналізу та синтезу – для виявлення сутності та ознак CRM-системи, а також уточнення інструментарію їх побудови; системно-структурний метод – для визначення місця й ролі хмарної CRM-системи з урахуванням структурних зв'язків її елементів; порівняльний метод – для розмежування CRM-систем відповідно до обраного критерію класифікації; також порівняльний метод застосовується для порівняння обчислювальних можливостей хмарних CRM-систем та можливостей їх баз даних.

Для вибору ефективної моделі CRM-системи для малого підприємства необхідно: зробити порівняльний аналіз можливостей хмарних і локальних CRM-систем; з'ясувати вимоги, які ставляться до функціональності сучасних хмарних CRM-систем; зробити класифікацію CRM-систем та надати характеристику їх призначенню.

Питання ефективності та класифікації CRM-систем досліджували Пейн Е., Базюк О.Д., Михалевич В.М., Цмось І.Г., Яворський А.В., Лейко А.О., Марковець О. В, Паздерська Р. С., Мозгова Г. В., Морозов А. О.

В сучасній науково-технічній літературі часто вживається такий дихотомічний поділ CRM-системи як eCRM і CRM [31, с. 90] [17, с. 505].

					<i>КНТЕУ 121–02з-01.МР</i>	Аркуш
						10
Вим.	Аркуш	№ докум	Підпис	Дата		

Основою такого поділу є можливість доступу до глобальної мережі. Таким чином, CRM-систему розуміють як систему, що функціонує offline, а eCRM як систему що функціонує on-line. Проте такий поділ, на нашу думку, не надає змоги охопити всі існуючі види CRM-систем і не відображає якісно їх поділ за інформаційним навантаженням на таку систему, інструментом зв'язку між компанією і клієнтами та співробітників між собою[28, с.37].

Однак, на нашу думку, розглядати eCRM як таку що здатна функціонувати у віртуальному просторі дещо звужує її функціональні можливості. Як явище eCRM доцільно розглядати у контексті її електронної складової, тобто eCRM може функціонувати в межах інформаційної системи як одного стаціонарного ПК (будь-якого іншого гаджету), так і об'єднувати декілька користувачів з персоналізованими технічними пристроями. Таким чином, головною ознакою eCRM є її електронна складова без урахування можливості її функціонування в on-line чи off-line режимі[28, с. 38].

Під час вибору виду CRM-системи розробник має зважувати на очікувані навантаження на систему при обробці інформації, наявні та можливі завдання такої обробки, інструменти захисту конфіденційної інформації та конкурентоспроможність технології з огляду на сучасні виклики бізнес середовища.

ECRM мають підвидове розмаїття, яке варто розглянути в контексті її класифікації з урахуванням вищенаведеної аргументації та важливості для компанії обрання ефективної CRM-системи для виконання специфічних її функцій:

I. За інструментом зберігання інформації CRM-системи поділяються на:

- документарні CRM-системи – це CRM-системи, які існують поза електронною системою та використовуються за допомогою паперових записників;
- електронні CRM-системи – CRM-системи, доступ до яких реалізується з

					<i>КНТЕУ 121– 02з-01.МР</i>	Аркуш
						11
Вим.	Аркуш	№ докум	Підпис	Дата		

використанням електронних інформаційних систем.

II. За способом доступу до даних та інструментом їх збереження електронні CRM-системи виділяють:

- eCRM-система стаціонарна – система без будь-якого мережевого з'єднання, де доступ користувача до системи і обробка даних здійснюється на тому ж пристрої, потужностями якого зберігаються данні;

- локальна eCRM-система – система, доступ користувача до якої можливо здійснити з іншого пристрою, під'єданого до локальної мережі, а дані такої CRM-системи зберігаються на стаціонарному сервері цієї мережі, або на вузлу (вузлах) такої локальної мережі;

- хмарна CRM-система – система, доступ до якої здійснюється засобами Інтернет-технологій, а дані зберігаються, як правило, в хмарних сховищах.

Схематично ці види та підвиди CRM-систем можна представити у вигляді (рис.1.1):

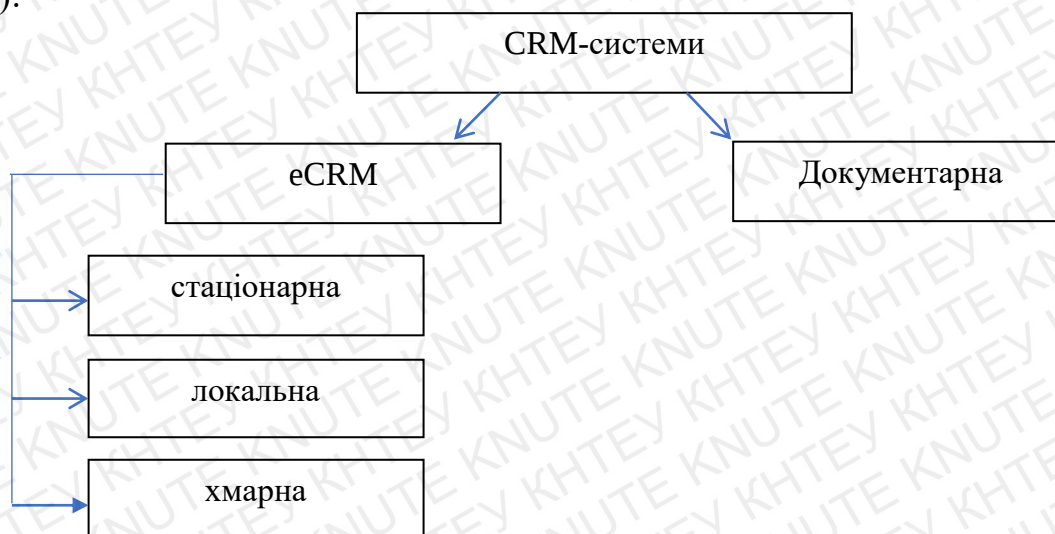


Рис. 1.1 Види CRM-систем за способом доступу до даних

III. Електронні CRM-системи можна класифікувати за програмним критерієм:

- ті, що не використовують спеціальне ПО для управління СУБД, шифрування даних, розподіл політик та планувальник задач (універсальні);

- ті, що використовують інструменти мов програмування та структурованих

					<i>КНТЕУ 121– 02з-01.МР</i>	Аркуш
						12
Вим.	Аркуш	№ докум	Підпис	Дата		

баз даних; як правило проектуються під конкретні специфічні вимоги компанії (спеціалізовані).

IV. За функціональним призначенням розрізняють наступні CRM-системи

а) операційні – метою яких є автоматизація бізнес-процесів, що відбуваються в місцях контакту з клієнтами. Вони забезпечують автоматизацію продажів, маркетингу і обслуговування клієнтів. Як зазначив Пейн Е., операційні CRM-системи в історичному плані стали головною статтею витрат [32].

б) аналітичні CRM-системи забезпечують пошук, накопичення, організацію, аналіз, інтерпретацію і використання даних, отриманих в операційній частині сервісу.

в) спільні CRM-системи покликані забезпечити використання спільних сервісів та інфраструктури для взаємодії компанії через її численні канали з клієнтами та співробітниками. [29, с. 10]. Вони передбачають використання різноманітних каналів зв'язку (радіочастотні, аналогові, оптичні).

В рамках спільних CRM-систем все більшого поширення набуває підвид – Social CRM. Інтегрованість соціальних мереж одна до одної надає можливість застосувати всі потужності онлайн середовища: YouTube канал, Facebook, LinkedIn, Twitter, Instagram, блогі, Telegram, WhatsUp тощо. Тепер збирати дані про потенційного клієнта, про його вподобання, прагнення можливо засобами соціальних мереж, адже вся інформація, що вільно надається користувачами, тобто поширюється ними особисто публічно, набуває правового режиму відкритої (публічної).

Головними перевагами у використанні Social CRM називають [25]:

1. Забезпечення підтримки клієнтів на онлайн-платформах, якими користуються клієнти.
2. Залучення клієнтів та взаємодія з ними в режимі реального часу.
3. Швидке вирішення проблем, оперативне відстеження скарг, зауважень в соціальних мережах та реагування на них.

					<i>КНТЕУ 121– 02з-01.МР</i>	Аркуш
						13
Вим.	Аркуш	№ докум	Підпис	Дата		

4. Пошук та винагорода потенційних агентів з просування продукції та послуг (особливо в Instagram), які рекламують бренди на своїх сторінках та інших осіб, зокрема клієнтів, які своєю діяльністю (наприклад, ведення тематичного блогу) допомагають іншим клієнтам та потенційним споживачам зорієнтуватися в продукції та її особливостях.

5. Поглиблення відносин з клієнтами, вивчення їх прагнень.

6. Побудова та просування репутації бренду.

Варто зазначити, що документарні CRM-системи використовувалися тривалий час і були незамінним інструментом до появи електронних інформаційних систем. Але з розвитком електронного інформаційного простору в рази збільшився об'єм бізнесу, що пов'язаний з купівлею-продажем, виконанням робіт, наданням послуг. Змінилися умови ведення бізнесу, його темпи, тому документарна форма не виконувала завдань і не відповідала вже вимогам, що ставилися перед CRM-системою. Однак, стверджувати, що документарні CRM-системи «канули в Лету» зарано, адже дослідження показало, що нині 4% компаній використовують паперову форму звітності [9].

Електронні CRM-системи наприкінці XX століття стали незамінним інструментом та значно вплинули на конкурентоспроможність бізнесу. Еволюція електронних CRM-систем була зумовлена саме викликами шалених темпів росту бізнес-рішень. [33].

Багато фірм, таких як Google, Amazon та Microsoft, широко застосовували хмарні обчислення в різних сферах. Наприклад, Google або Dropbox стали повсякденними інструментами для мільйонів людей. Більше того, багато підприємств в даний час використовують хмари, засновані на CRM, і надають Cloud CRM такі послуги, як Salesforce, Amazon та Microsoft Azure [16, с.178].

Важливою складовою CRM-системи є база даних, яка надає послуги, пов'язані з робочими навантаженнями. Тут варто зауважити, що Azure підтримує big data, як NoSQL, так і реляційні бази даних[5]. На основі послуг, що

					<i>KHTEY 121–023-01.MP</i>	Аркуш
						14
Вим.	Аркуш	№ докум	Підпис	Дата		

надаються в домені бази даних, різниця між трьома загальнодоступними хмарами наведена в наступній таблиці (Табл. 1.1)

Таблиця 1.1

Порівняння можливостей хмар для баз даних CRM-систем

Database Services	AWS	Azure	Google
Caching	ElastiCache	RedisCache	CloudCDN
Block Storage	EBS	Page Blobs	Persistent Disks
Object Storage	S3	Blobs and Files	Google Cloud storage Block
NoSQL (Indexed)	DynamoDB	Cosmos DB	Cloud Datastore
NoSQL (Key-value)	DynamoDB SimpleDB	Table Storage	Cloud Bigtable Cloud Datastore
Database Migration	Database Migration Service	Database Migration Service	
Manage Data Warehouse	Redshift	SQL Data Warehouse	
Manage Relational Database-as-a-Service	RDS	SQL Database Database for MySQL Database for PostgreSQL	Google Cloud SQL Cloud Spanner

Щодо обчислення, то будь-яка сучасна хмарна CRM-система має надавати послуги, пов'язані з обчислювальними навантаженнями. Ефективний хмарний провайдер має можливість масштабувати тисячі вузлів лише за кілька хвилин. Amazon EC2 надає основні обчислювальні послуги для конфігурації віртуальних машин (Virtual Machines) з використанням спеціальних або попередньо налаштованих AMI, тоді як Azure надає VHD (віртуальний жорсткий диск), подібний до AMI Amazon для конфігурації віртуальних машин. Google пропонує Google Compute Engine для впровадження хмарних обчислень[5]. В ситуації хмарних обчислень зберігання бази даних клієнтів знаходиться десь на робочому місці клієнта, і зберігання даних та заходи безпеки будуть у постачальника послуг в середовищі хмарних обчислень [24, с. 19].

Щодо інструментів, які на сьогодні використовуються компаніями в Україні, то їх використання можна продемонструвати в діаграмі на рис. 1.2.

					<i>КНТЕУ 121–023-01.МР</i>	Аркуш
						15
Вим.	Аркуш	№ докум	Підпис	Дата		

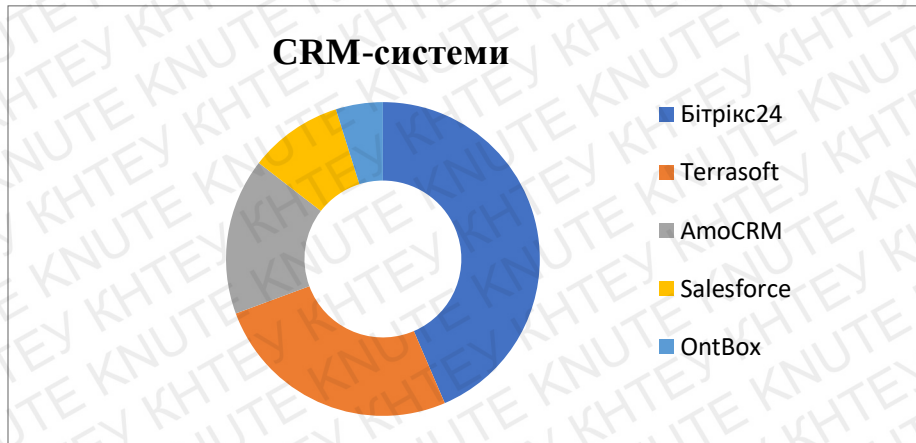


Рис. 1.2 Інструменти керування відносинами з клієнтами

CRM – це інтелектуальна технологія, яка вирішує проблему напрацювання клієнтської бази та її розширення, включає глибокий синтез стратегічного передбачення, корпоративного розуміння природи споживчої цінності в умовах багатоканальних дистриб'юторських ринків, розроблених під CRM додатків [32].

1.4 Архітектура типових хмарних CRM-систем

Даний підрозділ присвячений побудові архітектури типових хмарних CRM-систем, тобто запропонована модель має відображати всі наявні та необхідні складові, їх зв'язки без яких безпечне та продуктивне функціонування хмарної CRM-системи в сучасних умовах вбачається неможливим. Для досягнення поставлено мети будуть використані наступні наукові методи пізнання: логічний метод – для аргументації істинності виявлення модулів та їх логічних зв'язків для проекту архітектури хмарної CRM-системи; діалектичний метод – для розкриття цілісності об'єкта пізнання – самої архітектури типових хмарних CRM-систем; метод аналізу та синтезу стане у нагоді для уточнення місця бізнес логіки та шляху запитів від користувача до СУБД в архітектурному проекті. Отже, необхідно визначитися з складовими, їх зв'язками, інформаційними потоками CRM-системи.

					<i>КНТЕУ 121–023-01.МР</i>	Аркуш
						16
Вим.	Аркуш	№ докум	Підпис	Дата		

Центральним елементом будь-якої архітектури CRM-системи є бізнес логіка підприємства. Саме від виду діяльності підприємства, його мети залежить функціональне навантаження на CRM-систему та інструменти роботи з даними. Визначимо, які саме бізнес-процеси будуть автоматизовані в системі, яким способом вони будуть взаємодіяти всередині системи. Проте в даному підпункту мова піде про універсальне архітектурне рішення для хмарної CRM-системи будь-якого малого підприємства.

Бізнес логіка як ядро CRM-системи визначає автоматизацію бізнес-процесів підприємства, вона ж визначає які саме дані мають бути зібрані, збережені, якими методами оброблені, які бізнес правила застосовані, які інструкції з виводу даних реалізувати.

Звернення до даних відбувається за допомогою модулю «Запити CRUD». Саме в цьому модулі створюється запит до СУБД.

З модуля «Запити CRUD» запит потрапляє до модулю «Розподілювач навантаження». Його основна функція – запобігти втраті даних та забезпечити розподіл навантаження на СУБД. Розподілювач навантаження звертається до основної хмари з CRUD-запитом та перевіряє, чи спроможна хмара прийняти та обробити такий запит. Якщо відповіді від основної хмари не має, то модуль «Розподілювач навантаження» направляє запит-CRUD до резервної хмари з СУБД-реплікантом. Таким чином, модуль «Розподілювач навантаження» виконує роль запобіжника, що суттєво зменшує можливість втрати даних та зменшення швидкості їх обробки.

Наступним модулем, є «Хмарні сховища» – це хмара основна і хмара резервна. В них знаходяться база даних (БД) та Система управління базою даних (СУБД).

CRM-система яка досліджується є хмарною, отже СУБД буде розміщена не на локальному ПК, а в хмарах. Для безпеки інформаційної системи важливо мати резервну копію основної СУБД – СУБД-реплікант. Запит, що надійшов з модулю

					<i>КНТЕУ 121– 02з-01.МР</i>	Аркуш
						17
Вим.	Аркуш	№ докум	Підпис	Дата		

«Запити CRUD» потрапляє в модуль «Розподілювач навантаження», який перевіряє здатність основної хмари (в ній розгорнута основна СУБД) приймати й обробляти запити. За відсутності відповіді від основної хмари, запит перенаправляється в резервну хмару з СУБД-реплікантом. В такому разі СУБД-реплікант змінює свій статус та стає основною СУБД. Якщо обидві хмари не відповідають, то через визначений проміжок часу система повертає відповідь «503 Service Unavailable».

Користувач хмарної CRM-системи ініціює звернення до неї в модулі «Інтерфейс користувача». Інтерфейс користувача (UI - англ. User Interface) – це сукупність засобів, за допомогою яких користувач спілкується з різними пристроями, найчастіше з комп'ютером або побутовою технікою, або іншим складним інструментарієм (системою) [29, с.7].

На рис.1.3 проілюстровано яким чином взаємодіє CRM-система з користувачем. Вплив користувача на CRM-систему бере свій початок в модулі «Інтерфейс користувача», де користувач в залежності від своєї мети активними діями формує запит до CRM-системи. Інформаційний потік даних «Запити користувача» направлений від модулю «Інтерфейс користувача» до модулю «Бізнес логіка». В модулі «Бізнес логіка» засобами CMS перевіряються права доступу до системи поточним користувачем та відповідно до них здійснюється скеровування запиту до відповідного «Бізнес правила» в модулі «Бізнес логіка». «Бізнес правило» автоматично формує операцію, необхідну для обробки запиту користувача. Від модуля «Бізнес логіка» потоки даних потрапляють у модуль «Запити CRUD». Сформований запит CRUD через «Розподілювач навантаження» надсилається у хмару в СУБД та здійснює необхідну вибірку та збереження даних. З хмари запитані дані повертаються у модуль «Бізнес логіка» та за потреби піддаються обробці відповідно до «Бізнес правил» модулю «Бізнес логіка». Відповідь на запит – дані, що запитував у CRM-системи користувач отримує з модулю «Бізнес логіка» в модуль «Інтерфейс користувача».

					<i>KHTEU 121–023-01.MP</i>	Аркуш
						18
Вим.	Аркуш	№ докум	Підпис	Дата		

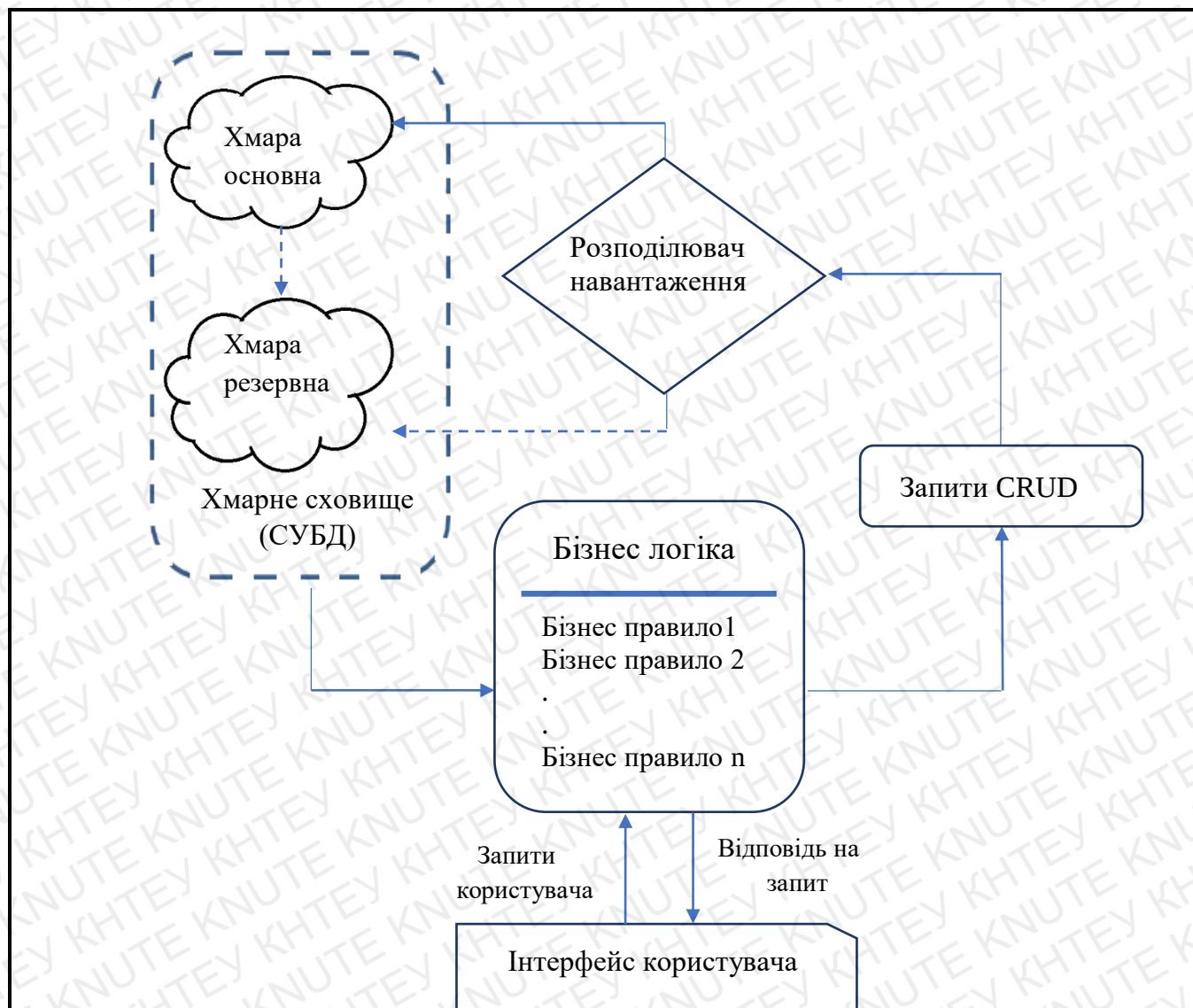


Рис. 1.3 Архітектура типової хмарної CRM-системи

Таким чином, для коректного функціонування хмарної CRM-системи необхідним є виокремлення основних її модулів: інтерфейс користувача, бізнес логіка, запити CRUD, розподільувач навантаження та хмарні сховища, де буде розгорнута СУБД. Зазначені модулі є універсальними для будь-якої хмарної CRM-системи та не залежать від виду та роду діяльності підприємства.

1.5 Висновки до розділу 1

Хмарна CRM-система – розташована в хмарних сервісах інформаційна система управління відносинами з клієнтами, функціонал якої надає можливості

					КНТЕУ 121–02з-01.МР	Аркуш
						19
Вим.	Аркуш	№ докум	Підпис	Дата		

збору, зберігання й аналізу інформації про споживачів, постачальників, партнерів, а також забезпечення безпеки та цілісності даних.

Наведена класифікація CRM-систем базовим розмежуванням виокремила документарні та електронні CRM-системи, які, в свою чергу мають свої підвиди: CRM-система ПК, локальна CRM-система, хмарна CRM-система.

Щодо особливостей програмної складової, то розподіл їх здійснюється на універсальні та спеціалізовані. За функціональним призначенням: операційні аналітичні, спільні. Проте який би вид CRM-системи не був би обраний, головним показником потужності такої системи є актуальність та достатня вибірковість даних, глибина та швидкість обробки таких даних, можливість прогнозування та моделювання поведінкової активності клієнта та оперативність реагування на запити клієнта.

З'ясовано, що потужним інструментом у CRM-системах є їх інтегрованість у Social media простір. Соціальні мережі значно прискорюють обмін інформацією з клієнтом, а отже, забезпечують його оперативну підтримку, надають потужні можливості для залучення нових клієнтів, успішно просувати свої товари та послуги.

Запропоновано використання модульної системи для побудови архітектури типової хмарної CRM-системи. Накреслена схема, яка складається з п'яти модулів, що логічно пов'язані інформаційними потоками: інтерфейс користувача, бізнес логіка, запити CRUD, розподілювач навантаження та хмарне сховище. Розроблена типова хмарна CRM-система є універсальним рішенням для будь-якої хмарної CRM-системи та не залежить від виду та роду діяльності підприємства.

					<i>КНТЕУ 121–02з-01.МР</i>	Аркуш
						20
Вим.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

РОЗРОБКА АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ

Даний розділ присвячено проектуванню архітектури хмарної CRM-системи для потреб малого підприємства «Сервіт». Основою для проекту слугує схема розроблена у п. 1.3 розділу I даного дослідження та представлена на рис.1.3. Для формування уявлення про бізнес логіку хмарної CRM-системи у даному розділі аналізу піддаються вимоги до такої інформаційної системи з урахуванням її функціональних особливостей.

2.1 Обґрунтування вибору програмних засобів для реалізації хмарної CRM-системи

Метою даного підрозділу є ознайомлення з сучасними програмними засобами, які використовуються для розробки хмарної CRM-системи та обрання оптимальних рішень, які якнайкраще відповідатимуть завданням CRM-системи для малого підприємства «Сервіт». Для такого завдання будуть застосовані наступні наукові методи: метод аналізу та синтезу – для виявлення сутності, можливостей гнучкості та інтегрованості для інформаційних систем; логічний метод пізнання – для аргументації істинності чи хибності наукових суджень відносно можливостей та значення гнучкості інформаційних систем, їх характерних ознак; системно-структурний метод – для визначення структурних зв'язків елементів гнучкості; порівняльний метод – для обрання апаратних та програмних засобів на основі їх функціональних характеристик; метод аналізу та моделювання – для проведення класифікації завдань хмарної CRM-системи

					<i>КНТЕУ 121– 023-01.МР</i>			
					<i>Проектування хмарної CRM-системи для малого підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Р2</i>	<i>21</i>	<i>53</i>
<i>Зав. каф.</i>		<i>Криворучко</i>		<i>24.05.21</i>	<i>Розробка архітектури хмарної CRM-системи</i>	<i>Факультет інформаційних технологій, 2 курс, 2м група</i>		
<i>Керівник</i>		<i>Харченко О.А.</i>		<i>24.05.21</i>				
<i>Гарант</i>		<i>Токар В.В.</i>		<i>24.05.21</i>				
<i>Розроб</i>		<i>Гоголін В.В.</i>		<i>24.05.21</i>				

«Сервіт»; діалектичний метод знайде своє застосування при розробці проекту архітектури хмарної CRM-системи «Сервіт».

В даній роботі досліджується бізнес процеси малого підприємства «Сервіт» для досягнення мети – спроектувати та програмно реалізувати хмарну CRM-систему. «Сервіт» – це українська компанія, що пропонує в оренду спеціалізовану техніку для будівельних та дорожніх робіт, а також надає супутні послуги з її обслуговування та використання. Критичним для побудови архітектури будь-якої інформаційної системи є врахування її призначення, особливостей та вимог до її функціональності.

2.1.1 Вимоги малого підприємства до автоматизованої системи управління взаємовідносинами з клієнтами

На думку П. Ван-дер Алста, призначення PAIS – підтримка оперативних бізнес-процесів, яка відбувається завдяки поєднанню досягнень інформаційних технологій з останніми досягненнями науки менеджменту[23, с 99]. Виходячи з наведеного можна стверджувати, що неможливо ефективно провадити бізнесову діяльність без можливості ситуативного впливу на інформаційну систему. Отже, інформаційна система має бути гнучкою.

Провідними вимогами до інформаційних систем сьогодні є здатність їх до масштабування та гнучкість. Проявом необхідності апаратного чи програмного втручання є, наприклад, потреба замовника у чаті, який функціонує без використання сторонніх готових рішень, або можливість інтеграції готового рішення в функціонуючу інформаційну систему. В такому випадку розробнику можуть знадобитись інші типи серверів, які здатні підтримувати постійне з'єднання. Також, у замовника може виникнути необхідність здійснювати відстеження гео-локації об'єктів. Для реалізації такого нововведення критичним буде використан-ня серверу оптимізованого під роботу з оперативною пам'яттю та спеціалізо-ваною БД (наприклад, Redis).

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						22
Вим.	Аркуш	№ докум	Підпис	Дата		

Одним з важливих проявів гнучкості інформаційної системи є здатність до масштабування. Прикладом потреби у масштабуванні є вимога до інформаційної системи витримати трафік з піковим навантаженням, який в сотні разів перевищує звичайні навантаження. Для ефективного управління взаємовідносинами з клієнтами інформаційна система має відповідати сучасним вимогам до таких систем для конкурентоспроможності підприємства та його функціональної мобільності та безпеки.

Щодо поняття «гнучкості» CRM-системи, то наукова думка охоплює наступні тези.

К. Шалендер зауважив, що гнучкість CRM-системи сприяє ефективній підтримці клієнта та задоволенню його вимог попри швидкий розвиток ринків та скорочення життєвого циклу продуктів. Тому, ідеально гнучкою CRM-системою він називав таку систему, яка безперешкодно з'єднує бази даних, процеси та процедури між відділами підприємства та його ланками [22, с. 40]. Прахальд і Рамасвами припустили, що сьогодні цінність продукту та послуги створюється компаніями разом зі своїми клієнтами, тому, як вони зауважили, для CRM-системи бажано мати необхідний рівень гнучкості для швидкого її пристосування до сучасних викликів [19, с. 217]. П. Фроу визначив поняття «гнучкість» для CRM-системи через здатність інтеграції її функцій і ресурсів: впровадження гнучкої CRM ґрунтується на міжфункціональній інтеграції процесів, людей, операцій, маркетингових можливостей, інформації, технологій та додатків [18, с. 168]. Берд Т. і Тернер Д. зауважили, що інформаційна система має бути не тільки гнучкою, а й чутливою [6, с. 167], Данкон Н. та Кін П. запропонував емпіричне дослідження технічних аспектів конструкції гнучкості архітектури інформаційної системи [10, с. 42][12, с. 179]. Головними такими аспектами вони назвали зокрема здатність інформаційної системи до:

1) інтеграції: під'єднання CRM-системи до внутрішніх і зовнішніх платформ, локальних та віддалених. Вимога інтегрованості хмарної CRM-системи здатна

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						23
Вим.	Аркуш	№ докум	Підпис	Дата		

проявлятися в можливості підтримувати зв'язок з провідними каналами продажів, зокрема соціальними мережами (Facebook, Instagram), месенджерами (Viber, Telegram, WhatsApp), системами аналітики (Google Analytics), системами оповіщення клієнтів (розсилками) тощо;

2) сумісності: налагодження конфігурації таким чином, щоб CRM-система була здатна використовувати мережеві та телекомунікаційні технології;

3) мультикористувацького та багатозадачного режиму;

4) стандартизації: данні та програмні модулі мають бути програмно та технічно незалежними та доступними для спільного використання;

5) модульності: без значного впливу на інформаційну систему та втрат даних можливість додавати, змінювати, видаляти будь-яку програмну та/або апаратну складову CRM-системи, компоненти даних архітектури.

Важливою ланкою для гнучкості архітектури CRM-системи є кваліфіковані ІТ фахівці, які обслуговують таку систему та здатні професійно реагувати на вимоги ринку та критичні ситуативні випадки.

Система «Сервіт» реалізує вимоги гнучкості наступним чином:

1) масштабування: використанням засобів Amazon Web Services (далі – AWS), зокрема здатність масштабувати оперативну пам'ять, розмір диску на хмарі та обчислювальну здатність через придбання додаткового процесорного часу.

2) інтеграція: хмарна CRM-система здатна витримувати будь-яку інтеграцію, як апаратну (наприклад, ір-телефонія), так і програмну шляхом створення спеціалізованих мікро-сервісів.

3) сумісність: CRM-система розташована в хмарі, реалізована програмними засобами, які підтримують протоколи доступу TCP та UDP;

4) мультикористувацький та багатозадачний режим – передбачений і дозволяє аутентифікувати та авторизувати користувача на використання CRM-

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						24
Вим.	Аркуш	№ докум	Підпис	Дата		

системи з визначеними політиками. Багатозадачність підтримується як апаратними засобами (кілька серверів), так і програмними;

5) стандартизація: програмна частина хмарної CRM-системи стандартизована за принципами SOLID, для обміну даними – JSON, для роботи з БД – ORM Doctrine.

6) модульність реалізовано у Three-tier архітектурі в трьох незалежних рівнях, які є змінними: БД може бути замінена наступним чином: MySQL на PostgreSQL; програмна частина – Php на C++ або GoLang, візуалізація – Bootstrap і jQuery можна замінити на Vue.js.

Three-tier архітектура розділяє програми на три логічні та фізичні обчислювальні рівні, є переважною архітектурою програмного забезпечення для традиційних програм клієнт-сервер.

2.1.2 Вибір програмних та апаратних засобів для CRM-системи «Сервіт»

Враховуючи перелічені у п. 2.1.1 вимоги до автоматизованої системи управління взаємовідносинами з клієнтами обрано наступні апаратні та програмні засоби.

Amazon Route 53 – направляє запити користувачів до інфраструктури AWS, балансувальників навантаження Elastic Load Balancing або кошиків Amazon S3. Також, його використовують для перенаправлення користувачів в інфраструктуру за межами AWS[34]. Також є можливість використовувати запити Amazon Route 53 до ресурсів AWS, таких як ELB (лише загальнодоступні зони)[20, с. 135]. Перевагою AWS є можливість використання широкого спектру глобальних хмарних продуктів, зокрема обчислювальних сервісів, сховищ, баз даних, мережових технологій, інструментів аналітики, мобільних сервісів, інструментів для розробників, інструментів управління, технологій Інтернету речей, засобів безпеки тощо [34]. Вищезазначені переваги стануть у нагоді

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						25
Вим.	Аркуш	№ докум	Підпис	Дата		

малому підприємству «Сервіт» для швидкого розвитку, адже надають можливість суттєво знизити витрати у сфері ІТ та забезпечити гнучкість та масштабування.

Платформа AWS дозволяє: 1) швидко та безпечно розміщувати на хостингу як існуючі, так і нові програми на основі моделі SaaS; 2) вибрати операційну систему, мову програмування, платформу інтернет-додатків, бази даних та інші необхідні сервіси; 3) використання Auto Scaling и Elastic Load Balancing. Окрім того, AWS надає можливості використання служб IaaS, SaaS, PaaS.

Інфраструктура як послуга (IaaS) – це різновид служби хмарних обчислень, що надає основні ресурси для обчислень, зберігання та мережного підключення на запит з оплатою в міру використання. IaaS допомагає зменшити витрати і мінімізувати складнощі, які пов'язані з придбанням фізичних серверів та інфраструктури центрів обробки даних та управлінням ними.

Рішення SaaS працюють безперебійно, завдання візуалізації з інтенсивним процесором виконуються економічно ефективно, а архіви та резервні копії даних зберігаються надлишково. Так, SaaS надає повноцінний набір програмного забезпечення, який сплачується постачальнику хмарних служб у міру використання. SaaS дозволить «Сервіт» швидко запускати програму з мінімальними попередніми витратами.

Платформа як послуга (PaaS) – є повноцінним середовищем розробки та розгортання у хмарі з ресурсами, які дозволяють надавати будь-які програми, від простих хмарних додатків до просунутих хмарних додатків промислового класу. PaaS також дозволяє уникнути витрат і труднощів, пов'язаних з придбанням ліцензій на програмне забезпечення, базової інфраструктури додатків, програмного забезпечення проміжного рівня, контейнерів, наприклад Kubernetes, або засобів розробки та інших ресурсів, а також керування ними. Графічно структура вищеназваних хмарних сервісів зображена на рис.2.1, а щодо їх переваг та недоліків, то для зручності викладемо їх у табличному вигляді (табл.2.1).

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						26
Вим.	Аркуш	№ докум	Підпис	Дата		

AWS відповідає в необхідному обсязі забезпечення безпеки та відповідності вимогам стабільності.



Рис. 2.1 Структура хмарних сервісів

Так, в AWS Nitro System система безпеки вбудована за умовчанням на рівні чіпа, що безперервно моніторить, захищає та перевіряє апаратне забезпечення інстансу та мінімізує потенційні види атак.

Таблиця 2.1

Порівняння сервісів AWS

Сервіс модель	Переваги	Недоліки
IaaS	- можливість прямого контролю над конфігурацією програми; - високопродуктивні обчислення; - поліпшення безперервності бізнес-процесів та аварійного відновлення	Сучасна інфраструктура в основному створена на замовлення
SaaS	- програмне забезпечення оновлюється постачальником - гнучкість ітерації - використання безкоштовного клієнтського програмного забезпечення	«Чорна скринька»: відсутність видимої базової інфраструктури для гарантії ризику
PaaS	- розробка тестування та розгортання додатків повністю потужностями хмари. - скорочення часу програмування. - спрощена розробка для декількох платформ, включаючи мобільні платформи - ефективне керування життєвим циклом додатків	Група розробників має працювати в рамках платформи. Кодування відбувається онлайн.

Окрім того, AWS підтримує такі сьогоденні важливі стандарти безпеки та сертифікатів відповідності як PCI-DSS, HIPAA/HITECH, FedRAMP, GDPR, FIPS 140-2 та NIST 800-171.

MySQL обрано через наступні його властивості: швидкість, багато-поточковість, використання багатокористувачького і надійного серверу баз даних SQL (Structured Query Language). MySQL Server здатен убезпечити «Сервіт» від великих навантажень. Програмне забезпечення MySQL має подвійну ліцензію. Також перевагою MySQL є легкість в адмініструванні та не вимогливість до ресурсів.

AWS Network Load balancer для MySQL Cluster – автоматично розподіляє вхідний трафік додатків за кількома цільовими об'єктами та віртуальними пристроями в одній або декількох зонах доступності.

PHP («Hypertext Preprocessor») – препроцесор гіпертексту – є поширеною мовою загального призначення з відкритим вихідним кодом. PHP створювався спеціально для ведення веб-розробок, код якого може використовуватися безпосередньо в HTML-код. PHP є доступним для більшості операційних систем, зокрема Linux, Unix, Microsoft Windows, macOS, RISC OS тощо. Також у PHP включена підтримка більшості сучасних веб-серверів, таких як Apache, IIS та багато інших. Версія PHP обрана 7.4 через наступні особливості: строга типізація класів; передзавантаження для покращення продуктивності (значно прискорює виконання скрипту і робить код чистішим і швидшим); функції з стрілками для короткого запису анонімних функцій; інтерфейс зовнішніх функцій відкриває нові можливості для розробки розширень на PHP; оператор розпакування в масивах (для неасоціативних масивів).

Symfony Framework – є провідним фреймворком PHP для створення веб-сайтів і веб-додатків. Symfony використовує шаблон проектування проекту MVC (див. рис. 2.2). Symfony обрано як фреймворк для CRM-системи «Сервіт» через наступні переваги:

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						28
Вим.	Аркуш	№ докум	Підпис	Дата		

- він розроблений для масштабних або складних проектів, які містять величезні можливості та використовуються значною кількістю клієнтів;
- використовує багаторазові компоненти (більш надійна модульність);
- використовує Twig як механізм шаблонів за замовчуванням (Додаток В);
- використовує об'єктно-реляційне відображення (ORM) для доступу до даних через Doctrine;
- підтримує такі бази даних: Drizzle, MySQL, Oracle, PostgreSQL, SAP Sybase SQL Anywhere, SQLite, SQLServer [15].

Крім того, архітектура MVC стає в нагоді, оскільки дозволяє відокремити бізнес-логіку від рівня презентації.

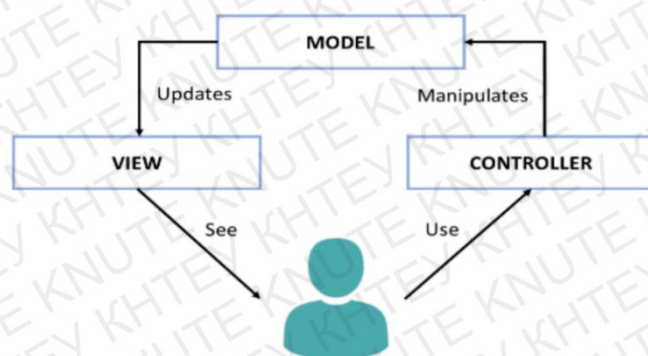


Рис. 2.2 Графічне зображення «Model View Controller»

Окрім того Symfony підтримує інструменти профілювання, такі як PHPUnit Pretty Printer і Blackfire, які полегшують тестування. Окрім цих можливостей, він пропонує пакети доктрин, які можна використовувати для створення надійних і швидких наборів тестування. Загалом, із Symfony є можливість запускати модульні тести, функціональні тести та інтеграційні тести [10].

Щодо роботи з базами даних, то Фреймворк використовує картографи даних для відображення об'єктів у базі даних. Він розділяє об'єктну модель і схему бази даних, дозволяючи змінювати стовпці бази даних без налаштувань кодової бази.

jQuery – це швидка, невелика і багатофункціональна бібліотека JavaScript. Бібліотека включає наступний функціонал: css-селектори sizzle, робота з dom-елементами (дозволяє вставляти та/або видаляти елементи DOM на HTML-

					КНТЕУ 121– 023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		29

сторінці, а також спрощує перенесення рядків), події, візуальні ефекти, доповнення на ажах (можливість адаптувати код для всіх веб-браузерів.). Отже, вона значно спрощує такі речі, як обхід і маніпуляції з документами HTML, обробку подій, анімацію та Ажах, завдяки простому у використанні API.

Bootstrap – інструмент, що дозволяє швидко створити сайт із стандартних блоків. Основні переваги Bootstrap у: кросбраузерності, швидкості створення інтерфейсу сайтів, стандартизації та адаптивності, економії часу, чистому CSS (але його сирцевий код використовує два найбільш популярні CSS-препроцесори Less та Sass), легкості та ефективності масштабується на сайті на основі єдиного базового коду: від телефонів і планшетів до настільних комп'ютерів [35].

Таким чином, на вибір програмних та апаратних засобів значний вплив вчинили наступні вимоги: кросбраузерність, стандартизація, гнучкість, здатність до масштабування, підтримка MVC, економія часу.

2.2 Класифікація завдань хмарної CRM-системи для малого підприємства «Сервіт»

Для проектування ефективної CRM-системи підприємства необхідно визначити масштаб бізнесу: призначення та функції CRM-системи. Для цього необхідно провести класифікацію її основних завдань, виокремити ієрархічні рівні та виявити зв'язки між ними. Отриману інформацію використати для з'ясування бізнес логіки підприємства «Сервіт».

Здійснімо класифікацію основних завдань хмарної CRM-системи для малого підприємства «Сервіт» за наступними критеріями.

I. Залежно від напрямку впливу

1. Внутрішні завдання – направлені на внутрішню організацію та комунікації на підприємстві.
2. Зовнішні завдання – направлені на управління відносинами з клієнтами.

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
						30
Вим.	Аркуш	№ докум	Підпис	Дата		

3. Змішані – збір та обробка інформації з метою документування, статистичної обробки, прогнозування клієнтської активності.

II. Внутрішні завдання

1. Формування задач співробітникам та отримання нагадувань про важливі події (комунікація).

2. Управління стадіями виконання замовлень, контроль виконання доручень (тел), відстеження результатів роботи (відомості) та робочим часом.

4. Виконання планів продажів менеджерами, дотримання термінів оплати і поставки (Excel 1С інтеграція).

5. Контроль страхових зобов'язань та технічного огляду спеціальної техніки.

III. Зовнішні завдання

1. Підтримка і консультування клієнтів (виконуючий роботи).

2. Партнерські програми з оренди спеціальної техніки (техніка).

IV. Змішані завдання

1. Автоматизації діловодства та документообігу (планування, звіти).

2. Збір та обробка аналітичної інформації компанії (БД клієнтів (об'єкти), ефективність роботи співробітників (звіти) тощо).

3. Забезпечення безпеки – налаштування прав доступу.

4. Автоматизація керування технологічним процесом (планування).

5. Рішення завдань централізованого збору й обробки інформації (відомості).

Функції CRM включають продажі, маркетинг, підтримку споживачів. В системі збирається, зберігається і аналізується інформація щодо клієнтів, орендарів та орендодавців, виконавців робіт, водіїв, а також про внутрішні бізнес-процеси компанії (відомості). Призначення CRM-системи для «Сервіт» – автоматизація бізнес процесів для ефективного обслуговування, ведення та підтримки клієнта. Проте для ефективного виконання такого призначення

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		31

достатню увагу треба звернути й на внутрішні комунікації в компанії, адже їх ефективність та оперативність є запорукою безперервного функціонування компанії та виконання нею взятих на себе зобов'язань. Останнє формує гудвіл – той нематеріальний актив, який безпосередньо справляє вплив на поведінку споживача.

Оскільки «Сервіт» є малим підприємством, то і користувачами CRM-системи є обмежена кількість співробітників з відмінними правами доступу до CRM-системи. Тому, внутрішня комунікація «Сервіт» направлена не стільки на узгодженість бізнес-процесів, скільки на виконання посадових функцій користувачів.

2.3 Проектування архітектури хмарної CRM-системи малого підприємства «Сервіт»

Завданням даного підпункту є побудова архітектури хмарної CRM-системи для малого підприємства «Сервіт» на основі розробленої загальної схеми у п. 1.3 розділу I. та вимог до апаратної та програмної частини CRM-системи у п. п. 2.1, 2.2. Для виконання поставленого завдання буде використані наступні дані:

1. Ресурс підтримки СУБД – платформа AWS.
2. Функції резервної хмари для цілей CRM-системи «Сервіт» буде виконувати сервіс Daily snapshot (backup, резервна хмара).
3. СУБД для системи, що розробляється обрано MySQL.
5. Розподілювач навантаження – Load balancer for My SQL Cluster.
6. Запити CRUD – є стандартизованими, визначені через запити до MySQL.
7. Бізнес логіка визначена через бізнес процеси, що класифіковані у п. 2.2, а також містить балансувальник навантаження Elastic Load Balancing (ELB).
8. jQuery Bootstrap.
9. Запити користувача і відповідь на них визначимо через Amazon Route 53 – DNS сервіс, який віддає IP адресу за доменним ім'ям. Зареєстровані домени

					КНТЕУ 121– 023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		32

Amazon Route 53 налаштовані на автоматичне оновлення.

Представлена на рис. 2.3 архітектура хмарної CRM-системи відображає основні компоненти та виконувані ними сервіси інформаційної системи з урахуванням вимог гнучкості, кросбраузерності, стабільності та безпеки.

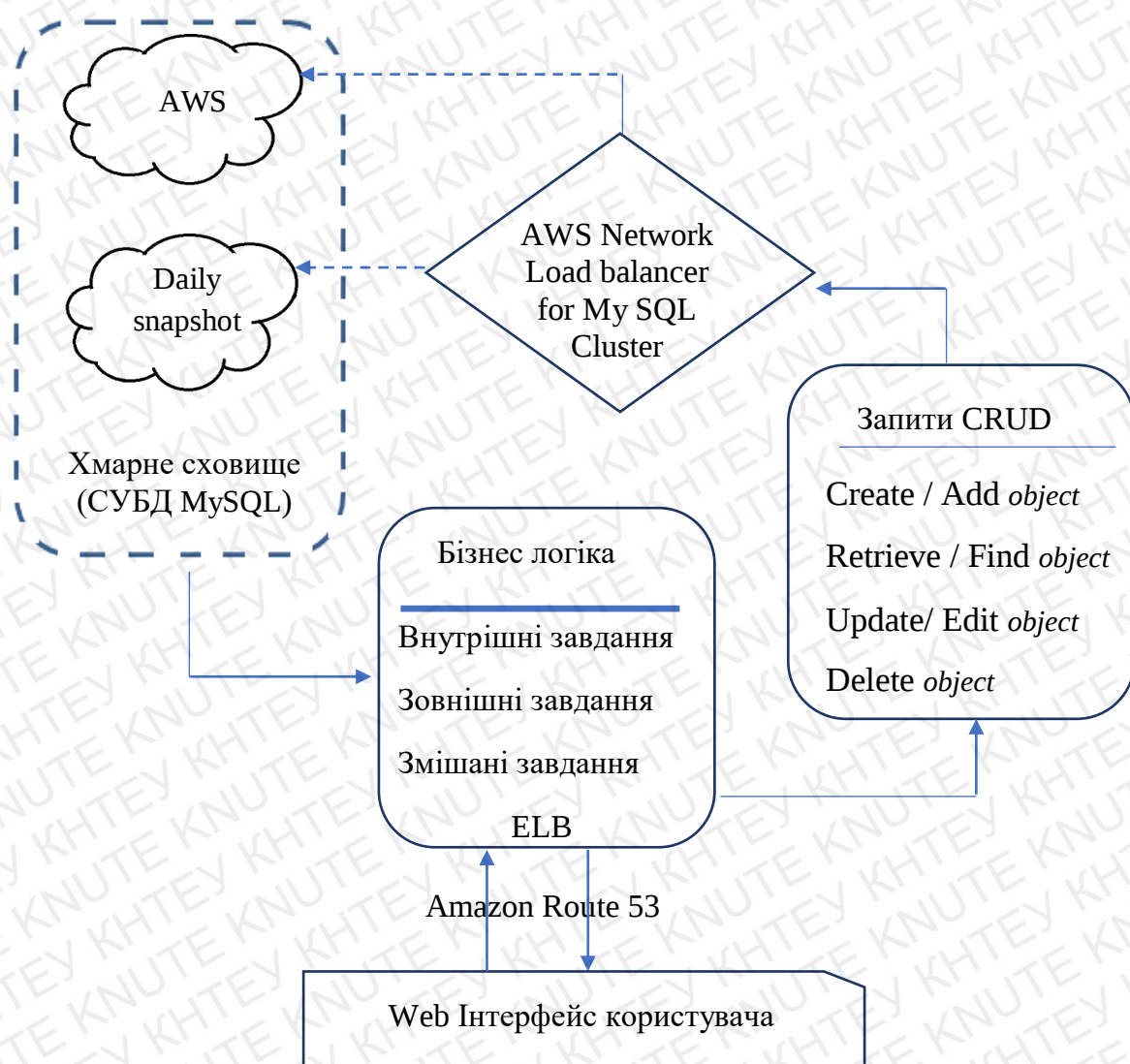


Рис. 2.3 Архітектура хмарної CRM-системи малого підприємства «Сервіт»

2.4 Висновки до розділу 2

Відповідно до поставлених завдань в даному розділі проаналізовані бізнес-процеси малого підприємства «Сервіт», на основі яких визначені функціональні особливості інформаційної системи. З'ясовані вимоги до хмарної CRM-системи:

					КНТЕУ 121–023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		33

здатність системи до масштабування, інтеграції, сумісності, мультикористувацького та багатозадачного режиму, стандартизації, модульності, стабільності та безпеки.

Обрано наступні апаратні та програмні можливості для забезпечення можливостей гнучкості CRM-системи «Сервіт»: для інтеграції – інструменти хмарного сервісу та сховища, для сумісності – будуть використані програмні засобами, які підтримують протоколи доступу TCP та UDP; для мультикористувацького та багатозадачного режиму – передбачена система, що дозволяє аутентифікувати та авторизувати користувача на використання CRM-системи з визначеними політиками, а багатозадачність підтримуватиметься як апаратними засобами (кілька серверів) так і програмними, стандартизації, модульності; для стандартизації – програмна частина хмарної CRM-системи буде стандартизована за принципами SOLID, для обміну даними – JSON, для роботи з БД – ORM Doctrine; для модульності – прийнято рішення про реалізацію у Three-tier архітектурі в трьох незалежних рівнях, які є змінними на вимогу.

Здійснено класифікацію основних завдань хмарної CRM-системи для малого підприємства «Сервіт», яка розкрила основні завдання малого підприємства «Сервіт», що вплинуло на формування бізнес логіки CRM-системи. Наведена класифікація вплинула на вибір апаратних та програмних засобів для реалізації програмної частини проекту.

Побудована та схематично зображена архітектура хмарної CRM-системи для малого підприємства «Сервіт». Представлена схема відображає основні компоненти та виконувані ними сервіси інформаційної системи з урахуванням вимог гнучкості, кросбраузерності, стабільності та безпеки.

					<i>КНТЕУ 121–023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		34

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ХМАРНОЇ CRM-СИСТЕМИ

Даний розділ присвячений визначенню основних програмних модулів хмарної CRM-системи та їх структури відповідно до потреб бази даних малого підприємства «Сервіт». Особливої уваги потребує аналіз запитів як до окремих сутностей бази даних так і до системи в цілому. Оскільки, окрім технічних інструментів забезпечення безпеки CRM-системи важливим є її правовий захист, то доцільним вбачається розглянути правові заходи захисту хмарної CRM-системи як цілісного об'єкту правового захисту, так і окремих компонентів інформаційної системи.

3.1 Розробка баз даних, знань, системи управління операціями хмарної CRM-системи малого підприємства «Сервіт»

Для досягнення цілей даного підрозділу використані наступні методи наукового пізнання: метод критики – для виявлення проблеми безпеки, з'ясування підходів до вдосконалення інструментів охорони та захисту CRM-системи, а також недоліків ORM-інструменту; теоретико-прогностичний метод – для розробки пропозицій щодо вдосконалення функціонування хмарної CRM-системи; системно-структурний метод – для з'ясування структури і послідовності виконання SQL запитів до CRM-системи. Перш ніж розпочати розробку бази даних для підприємства «Сервіт» розглянемо інструментарій та етапи звернення до СУБД для отримання даних з неї, а також для модифікації самої бази даних.

					<i>КНТЕУ 121–023-01.МР</i>			
					Проектування хмарної CRM-системи для малого підприємства	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		РЗ	35	53
Зав. каф.		Криворучко О.В.		21.06.21	Програмна реалізація архітектури хмарної CRM-системи	Факультет інформаційних технологій, 2 курс, 2м група		
Керівник		Харченко О.А.		21.06.21				
Гарант		Токар В.В.		21.06.21				
Розроб		Гоголін В.В.		21.06.21				

З'ясуємо переваги ORM (Object-Relational-Mapping) та ті проблеми, які можуть виникнути під час їх використання. Також, визначимо склад та структуру бази даних малого підприємства «Сервіт», побудуємо загальну модель бази даних, використовуючи її окремі сутності. Для з'ясування зав'язків між об'єктами розробимо схему сутність-зв'язок.

3.1.1 Мова запитів SQL як основа обробки даних в БД

SQL є декларативною мовою запитів і підтримується Американським національним інститутом стандартів (ANSI). Запит має етапи логічної та фізичної обробки. Етапи логічної обробки визначають порядок, у якому логічно обробляються різні оператори. Нижче на рис. 3.1 наведено основний синтаксис запиту разом із логічним номером порядку, в якому він буде виконуватися:

```
SELECT t.*
FROM Entities\Plan t
WHERE (t.date BETWEEN :start AND :end)
AND t._object IN (9,13,7,33,45,6,59,34)
```

Рис. 3.1 Синтаксис SQL запиту

У наведеному вище запиті використано оператори в такій послідовності: перший — оператор FROM, тоді як оператор, який записаний в першому рядку SELECT, обробляється майже останніми. На кожному кроці створюється тимчасова віртуальна таблиця (доступна лише для SQL Server), яка використовується як вхідні дані для наступного кроку. Жодна з проміжних віртуальних таблиць не доступна зовнішньому запиту, лише віртуальна таблиця, згенерована на останньому етапі, повертається абоненту.

Етапи фізичної обробки являють собою механізм фізичної обробки запиту в БД. Розуміння етапів фізичної обробки запитів SQL допомагає розробникам робити правильний вибір під час розробки схеми бази даних і написання запитів [14].

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		36

Кожного разу, коли SQL Server отримує запит для виконання, він виконує два основних кроки для повернення результату запиту. Першим кроком є компіляція запиту, яка генерує план виконання запиту реляційним механізмом SQL Server, а другий крок – виконання плану виконання запиту механізмом зберігання даних SQL Server.

Необхідно зауважити, що під час фізичної обробки запиту SQL Server отримує допомогу від двох компонентів нижче: 1) реляційний механізм SQL Server – відповідає за генерацію плану виконання запиту; 2) механізм зберігання даних (сховище) SQL Server – отримує план виконання запиту як вхідні дані та виконує дії для повернення бажаного результату.

В СУБД запит аналізується, а потім обробляється оптимізатором запитів, який створює план виконання. Коли будь-який запит досягає SQL Server, перше місце, куди він потрапляє, – це реляційний механізм. Тут процес компіляції запиту відбувається в три етапи: парсінг (аналіз), прив'язка та оптимізація.

Парсінг (аналіз) – це процес перевірки синтаксису – чи правильно написаний запит чи ні. Він не перевіряє, чи існує стовпець, який використовується в речення WHERE, у будь-якій зазначеній таблиці в пропозиції FROM. Результатом цього процесу є дерево розбору або дерево послідовностей.

Зв'язування – це процес, який виконується алгебризатором. Він перевіряє правильність семантики запиту; наприклад, чи є вказані дві таблиці, об'єднані в пропозиції FROM, дійсно таблицями чи ні. Алгебризатор створює дерево процесора запитів, яке працює як вхідні дані для оптимізатора запитів [14].

Оптимізація є останнім кроком у процесі компіляції та генерує план виконання запиту. Оптимізатор завжди намагається знайти спосіб обробки/виконання запиту за мінімальний час; він порівнює кілька варіантів із застосуванням усіх можливих перестановок і комбінацій, щоб створити найефективніший план виконання запиту за розумний час.

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		37

Якщо поданий запит простий, наприклад `SELECT * FROM <<ім'я_таблиці>>`, оптимізатор не витратить час на оптимізацію плану, і оптимізація на основі фактичних витрат у таких випадках не потрібна.

Оптимізатор розглядає доступну статистику за допомогою дерева процесора запитів (дерево, створене процесом зв'язування) і евристично вибирає найкращий план.

3.1.2 Object-Relational-Mapping як рішення для відображення бізнес об'єктів

Сьогодні ORM поширені повсюдно: кожна платформа має принаймні один ORM, іноді з «офіційним» API. Запити, що генеруються фреймворками об'єктно-реляційного відображення (ORM), призначені для подолання труднощів ORIM (об'єктно орієнтований імпеданс). Варто зазначити, що невідповідність об'єктно-реляційного імпедансу (ORIM) виникає, коли об'єктно-орієнтована розробка додатків – ієрархічна парадигма – зустрічається з рівнем реляційної бази даних, парадигмою на основі набору.

З точки зору реалізації, ORIM означає подолання невідповідності між викликом методу в програмі та створенням мови структурованих запитів (SQL), що вимагається для цього методу. Використання вбудованого SQL може вирішити цю потребу, але цей метод нетерпимий до змін схеми і може призвести до недоліків безпеки. Потреба розглядати таблиці як об'єкти в інтерфейсі програми-бази даних натомість породила рішення – використання класу інструментів, відомого як фреймворки об'єктно-реляційного відображення (ORM). Вони призначені для подолання розриву між викликами об'єктно-орієнтованих методів і створенням SQL. Фреймворки ORM відрізняються за особливостями, але зазвичай зберігають внутрішню модель даних і використовують бази правил і евристики для створення SQL з цієї моделі даних у відповідь на запити програми [7, с. 23].

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		38

Варто зауважити, що можливі проблеми із запитами, згенерованими ORM. Моделі (в узгодженні Model-View-Controller (MVC)) дуже тісно пов'язані зі схемами баз даних. Тому, при зміні схем є загроза несумісності моделей. Інша пов'язана проблема – спадкування. Наприклад, якщо класу надано можливості створення, оновлення та вставки, підкласи можуть успадковувати від цього класу, що може надати дозвіл на прямий доступ до бази даних, зменшуючи єдність у додатках.

Виклик методів, наданих ORM, а не виконання збережених процедур або виконання вбудованих запитів видається зручним та надає переваги сумісності з об'єктно-орієнтованими мовами програмування та вписується в сучасний дух розробки додатків на основі безперервної інтеграції [7]. Однак, одним важливим недоліком такого підходу є накладні витрати на обслуговування, пов'язані з утриманням концептуальної або логічної моделі даних і фізичної моделі даних в синхронності, що може проявлятися у труднощах підтримання базового коду[4, с. 40-41]. Приклад опису сутності засобами ORM Doctrine до моменту створення об'єкту: «use \Doctrine\ORM\Mapping as ORM;». Загалом він є псевдонімом, який переназначає складний шлях як просте позначення ORM.

Таким чином, використання ORM значно поліпшує роботу з базою даних через наявні інструменти доступу до неї та здатність відображати сутності в об'єктно-орієнтованому вигляді.

3.2 Визначення складу та структури бази даних

Загальна модель системи, яка піддана аналізу, являє собою ієрархічну структуру, яка складається з декількох рівнів абстракції з обмеженою кількістю компонентів на кожному рівні. Одним з головних принципів структурного системного аналізу є виділення на кожному рівні абстракції найсуттєвіших компонентів або елементів системи. Загальна схема бази даних зображена на рис. 3.2. Розглянемо більш детально кожен з сутностей БД «Сервіт» для розуміння

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		39

структури БД. Для наочності використаємо схематичне представлення сутностей. Кожна з сутностей має поле «id», але для зручності воно не буде відображено на схемах об'єктів.

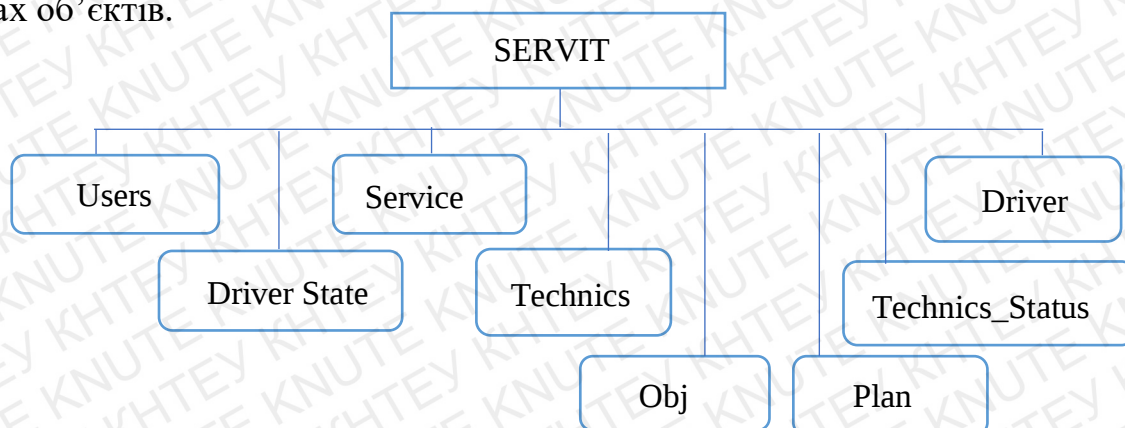


Рис. 3.2 Загальна схема бази даних CRM-системи «Сервіт»

Перший об'єкт – «Users» – являє собою сховище аутентифікаційних даних користувачів хмарної CRM-системи «Сервіт» (рис. 3.3). Для забезпечення надійного захисту даних від несанкціонованого доступу було прийняте рішення зберігати паролі на сторонньому ресурсі – бастіон Хост, а не в базі даних.

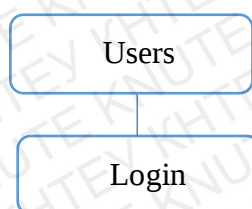


Рис. 3.3 Схема об'єкта «Users»

Наступним об'єктом для розгляду є «Service» (рис. 3.4). Полями його є «Code» – код послуги, що надається підприємством «Сервіт»; «Name» – назва послуги в інформаційній системі; «Created» – час та дата створення послуги в БД (час, з якого вона є доступною для замовлення).

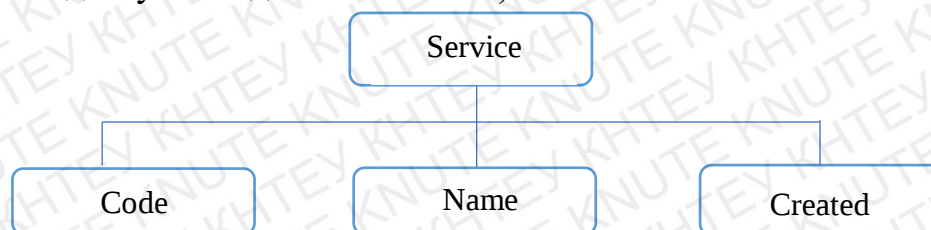


Рис. 3.4 Схема об'єкта «Service»

					КНТЕУ 121– 023-01.МР	Аркуш
						40
Вим.	Аркуш	№ докум	Підпис	Дата		

Об'єкт «Technics_status» містить перелік статусів техніки, яка знаходиться в розпорядженні підприємства (рис. 3.5). Даний об'єкт має лише одне поле «Name», який містить позначення статусу техніки.

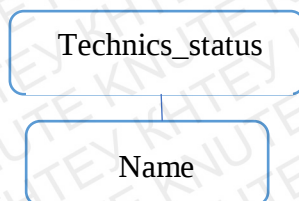


Рис. 3.5 Схема об'єкта «Technics_status»

Об'єкт «Driver_state» містить перелік стану водіїв, а єдине поле «Name» – позначає назву такого стану (рис. 3.6).

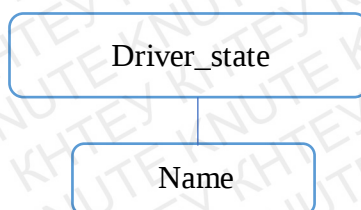


Рис. 3.6 Схема об'єкта «Driver_state»

Об'єкт «Driver» – інформація про водія: поле «Name» - П.І.Б. водія, «Phone» – номер телефону водія, поле «_State» є полем-зовнішнім ключем для зв'язку з сутністю «Driver_state», поле «Created» – містить інформацію про дату прийняття на роботу водія (рис.3.7).

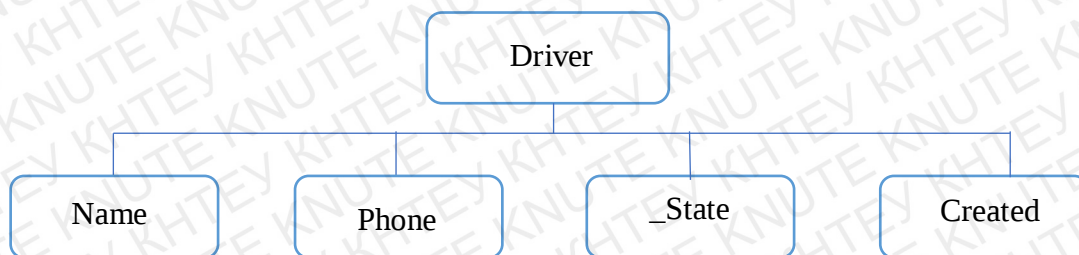


Рис. 3.7 Схема об'єкта «Driver»

Об'єкт «Obj» містить інформацію про об'єкти, на якому ведеться робота з надання послуги. Поле «Name» – назва об'єкту, поле «Address» – адреса об'єкту, на якому надається послуги, поле «Phone» – номер телефону клієнта, поле «Created» – дата додавання об'єкту у систему (рис 3.8).

					КНТЕУ 121– 023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		41

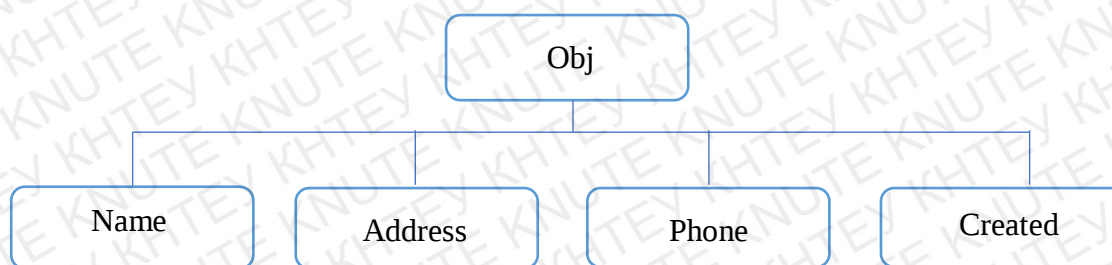


Рис. 3.8 Схема об'єкта «Driver»

Об'єкт – «Technics» – містить інформацію про наявні транспортні засоби на підприємстві та доступні для оренди. Поле «Name» – найменування транспортного засобу, поле «Lastinspection» – останній проведений технічний огляд транспортного засобу, поле «Cost» – ціна оренди за годину, «_driver» – є полем-зовнішнім ключем для зв'язку з сутністю «Driver» та вказує на те, за яким водієм закріплено транспортний засіб; «Endofinsurance» – останній день дії договору страхування транспортного засобу; «Created» – дата реєстрації транспортного засобу на підприємстві (рис. 3.9).

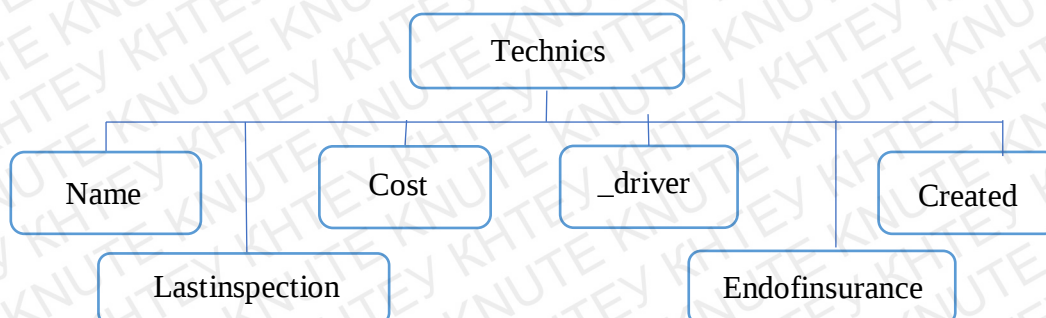


Рис. 3.9 Схема об'єкта «Technics»

Об'єкт «Plan» містить данні про всі вищеназвані об'єкти, задіяні в наданні послуги на зазначену дату (рис. 3.10).

Для розподілу повноважень між користувачами системи важливо визначити політики доступу до CRM-системи. Побудуємо метасистеми для визначення їх можливостей впливу на систему та обсягів повноважень на використання даних. Користувачами системи є: директор, відділ операторів, бухгалтерський відділ.

					КНТЕУ 121– 023-01.MP	Аркуш
						42
Вим.	Аркуш	№ докум	Підпис	Дата		

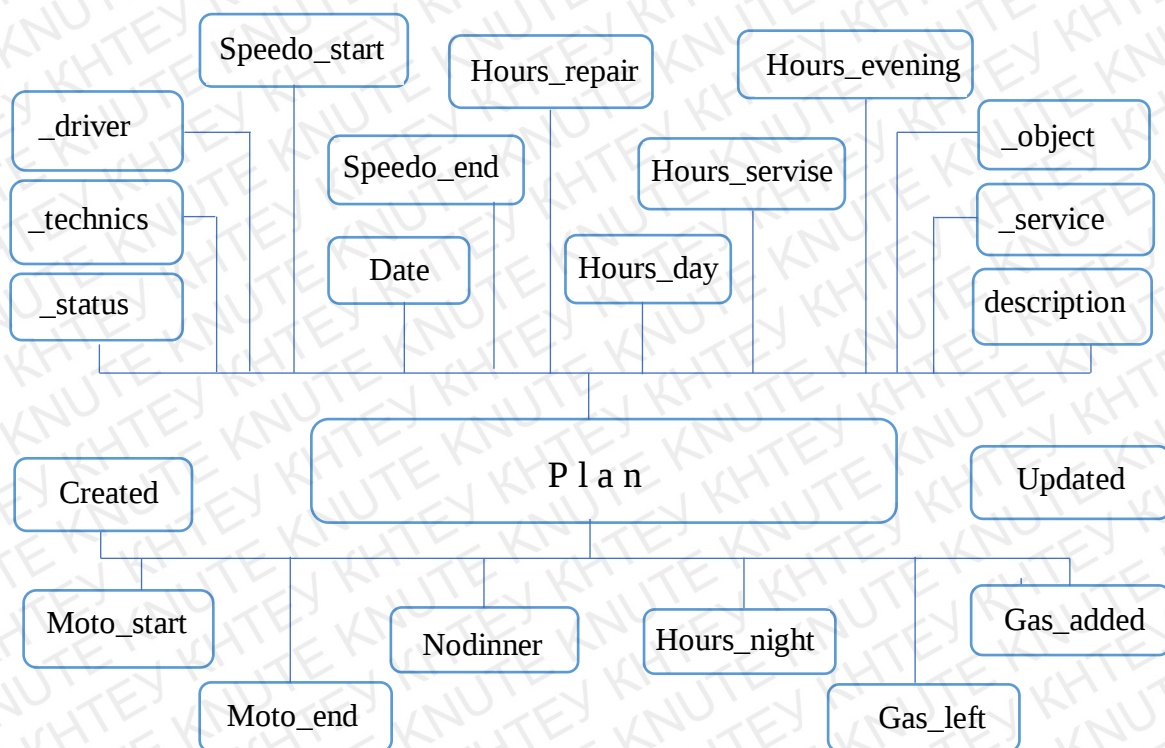


Рис. 3.10 Схема об'єкта «Technics»

Для директора важливим є доступ до даних інформаційної системи для моніторингу за операційною діяльністю підприємства та контролю виконання взятих на себе зобов'язань. Отже, для користувача Директор передбачається функції читання даних з будь-якого ресурсу бази даних «Сервіт». Подібні політики доступу є у відділу бухгалтерії для формування звітності підприємства. Побудуємо метасистему процедур «Читання даних» та «Введення даних» (рис. 3.11).

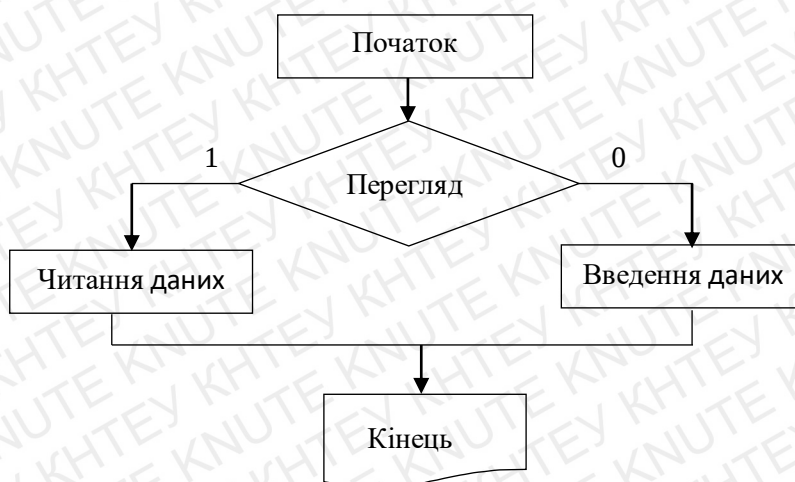


Рис. 3.11 Метасистема процедур клієнтської сторони

Оператор хмарної CRM-системи як адміністратор має повний доступ до бази даних та наділений повноваженням для її модифікації. Метасистема сторони адміністратора системи виконує процедури: модифікація бази даних та обробка інформації. Вона представлена на рис. 3.12

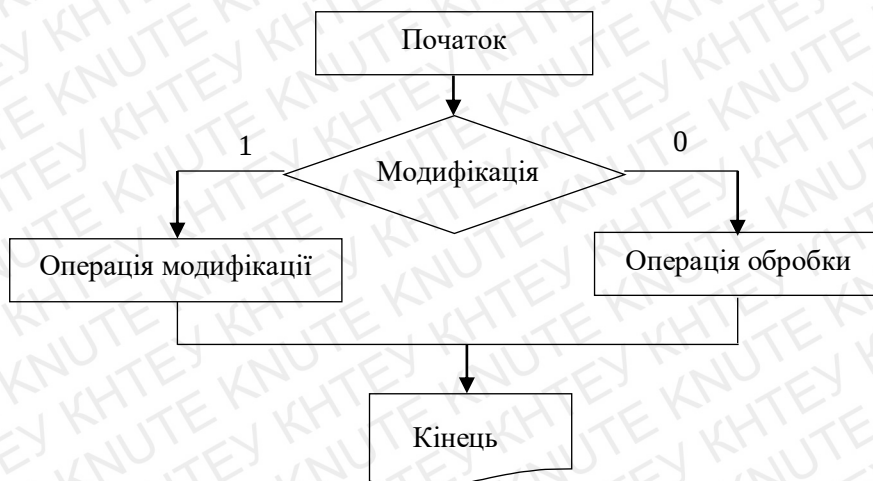


Рис. 3.12 Метасистема процедур сторони адміністратора

На рис. 3.13 наведена діаграма «Сутність-зв'язок», структура якої була запропонована Пітером Ченом, який поставив задачу проектування моделей та опублікував свою працю о 1975 р. [8]. Вона становить графічне зображення, за допомогою якого можна описувати об'єкти логічних моделей даних та відношення між об'єктами.

Оскільки базовими елементами моделі „Сутність-зв'язок” є сутності, атрибути та зв'язки, то для побудови моделі буде використано наявні в CRM-системі «Сервіт» вищеназвані елементи. Тип зв'язку – це осмислена асоціація між типами сутностей. Типу сутності та типу зв'язку відповідає певний набір атрибутів[26, с.9-10].

Отже, за допомогою діаграми «Сутність-зв'язок» здійснена побудова формалізованої структури бази даних хмарної CRM-системи для малого підприємства «Сервіт». Вона стане у нагоді при формуванні CRUD запитів до системи, а також при побудові бізнес-правил блоку «Бізнес логіка».

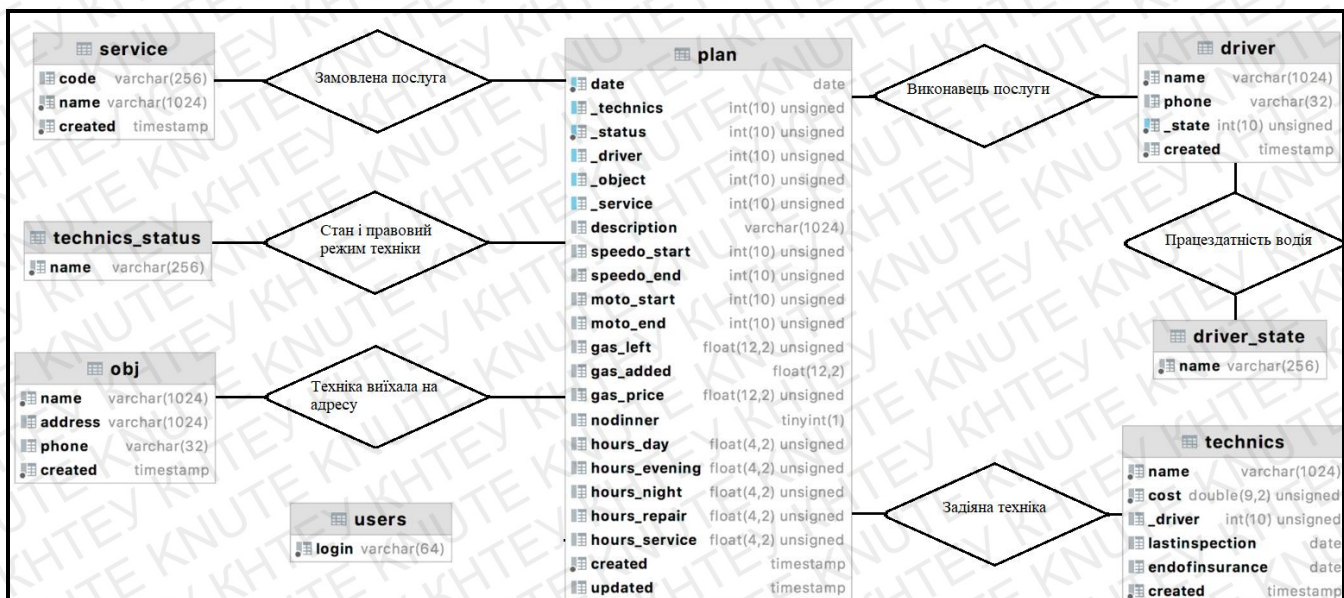


Рис. 3.13 Діаграма «Сутність-зв'язок»

Змодельовані у вигляді алгоритмічних процедур метасистеми операцій з даними надають уявлення про розгалужену систему політик доступу до даних CRM-системи, а графічно-логічне зображення об'єктів стане основою для створення бази даних.

3.3 Програмна реалізація хмарної CRM-системи та опис програмних модулів

Даний підрозділ присвячений розгляду програмних модулів, які використані для програмної реалізації хмарної CRM-системи малого підприємства «Сервіт». Основу структури програмної частини проекту становить розроблена у п. 2.3 Архітектура хмарної CRM-системи малого підприємства «Сервіт».

Як зазначалося вище для програмної реалізації хмарної CRM-системи обрані: фреймворк Symfony 5, мови програмування: PHP 7.4, jQuery Bootstrap, MySQL 8, сервіси AWS.

Для програмної реалізації хмарної CRM-системи було створено модулі, які відповідно до виконуваних ними функцій згруповані у теки.

Тека Controllers містить модулі, які приймають та опрацьовують запити користувача. До теки Controllers віднесено наступні програмні модулі:

					КНТЕУ 121–023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		45

- Admin.php – опрацьовує запити користувача з правами адміністратора;
- DefaultController.php – опрацьовує запити, які не опрацьовані іншими контролерами;
- Endpoints.php – опрацьовує Ajax запити;
- routes.yml – є конфігураційним файлом, який описує правила та послідовність завантаження контролерів;
- Test.php – опрацьовує запити покликані зручно опрацьовувати нові функції в процесі розробки або модифікації системи.

До теки Entities входить опис сутностей бази даних відповідно: AbstractEntity.php, Driver.php, DriverState.php, Entity.php, EntityEnterface.php, Obj.php, Plan.php, Service.php, Technics.php, TechnicsStatus.php, Test.php (див. Додаток В).

Приклад використання ORM для опису сутності наводився вище: « use Doctrine\ORM\Mapping as ORM;». Оператор use Doctrine\ORM\Mapping AS ORM робить усі класи в цьому просторі імен доступними як @ORM\ClassName[36]. Порядок здійснення опису сутності наведемо на прикладі таблиці «Driver», що має поля: id, name є наступним: « @ORM\Table(name="driver_state")» – оголошення про опис сутності «Driver_state»

Опис класу сутності має наступний вигляд:

```
class DriverState extends AbstractEntity
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue
     * @var int
     */
    ...
}
```

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		46

Для доступу до полей сутності та операцій над ними використовують функції Get(), Save(), Remove().

Отже, використання ORM для опису сутностей значно поліпшує представлення їх як об'єкти, а виклик методів та генерація запитів є зручним інструментом для роботи з базою даних.

3.4 Захист CRM-системи як об'єкта права інтелектуальної власності

Новостворений програмний продукт вимагає правової охорони, тому виникає питання який правовий режим у новоствореної CRM-системи? Для з'ясування можливостей правового захисту CRM-системи необхідно з'ясувати її правову природу. У Розділі 1 було з'ясовано, що поняття CRM-системи охоплює функціонал, який надає можливості збору, зберігання й аналізу інформації про споживачів, постачальників, партнерів, а також забезпечення безпеки та цілісності даних. З цього визначення можна виокремити такі ознаки CRM-системи: інструмент управління взаємовідносинами з клієнтами, ресурс для накопичення та збереження даних, функціонал статистичної обробки інформації та її впорядкування, засіб підтримки зв'язку та обміну даними з партнерами.

CRM-система є продуктом творчої інтелектуальної діяльності архітекторів програмного забезпечення, розробників та дизайнерів. Складність такого об'єкта визначається його структурою: безпосередньо програмна складова, графічний інтерфейс, база даних, інформація. Кожен елемент CRM-системи здатен мати свої особливості правової охорони відповідно до його правового режиму. Творця об'єктів авторського права законодавець іменує автором (ст. 7 Закону України «Про авторське право і суміжні права» від 23.12.1993р.)[3].

Щодо програмної складової та бази даних, то ці частини CRM-системи набувають правового режиму об'єктів інтелектуальної власності, а саме – авторського права відповідно до п. 3. ч. 1. ст. 8 Закону[3]. У ст. 18 Закону [3] зокрема вказано, що комп'ютерні програми охороняються як літературні твори та

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		47

незалежно від способу чи форми їх вираження. З назви ст. 19 Закону [3] та її змісту випливає, що бази даних прирівнюються до таких об'єктів як збірники та інші складені твори, а творчою працею є підбір, розташування та упорядкування даних. Законодавець іменує творця бази даних (або іншого складеного твору) і автором, і упорядником збірника, а ст. 8 Закону[3] називає авторами осіб, що створили об'єкти авторського права, зокрема баз даних.

Стосовно графічного інтерфейсу, то об'єктом охорони буде:

1) ескіз такого інтерфейсу – створений художником малюнок, є об'єктом авторського права відповідно до п. 8 ч. 1 ст. 8 Закону[3]. В даному випадку творець такого малюнку є його автором, а малюнок охороняється як твір образотворчого мистецтва;

2) візуалізація web-дизайнером у програмі вже розробленого художником ескізу інтерфейсу. Нажаль, творча праця web-дизайнера не знайшла свого визначення та захисту у Законі[3], тому, якщо його праця буде мати творчу складову програмна візуалізація ескізу підпадає під охорону як об'єкт авторського права – комп'ютерна програма. Проте, якщо взяти до уваги, що web-дизайнер використовує об'єкт авторського права – твір образотворчого мистецтва і розміщує його в коді комп'ютерної програми, то можна розглядати такі дії як розташування та упорядкування, що відповідно до ст. 19 Закону[3] є діями з створення такого об'єкту авторського права як складений твір.

Варте уваги і положення ст. 19 Закону[3], де законодавцем зазначено, що охорона баз даних не поширюється на самі дані чи інформацію і не зачіпає будь-яке авторське право, що відноситься до самих даних чи інформації, які містяться у базі даних.

Таким чином, можна зробити висновок, що Закон «України про авторське право і суміжні права» (1993) має прогалини та певні колізії, які потребують уточнення, зокрема щодо правового статусу авторів та упорядників, правового режиму окремих об'єктів цифрового середовища, не зайвим буде уточнення щодо

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		48

строків та умов охорони об'єктів, що існують у віртуальному середовищі, оскільки програмне забезпечення морально застаріває і потребує оновлення, зміни, але якщо код захищати на рівні літературного твору, то забезпечення такого немайнового права як недоторканість твору не можливе[27].

3.5 Висновки до розділу 3

Побудована ієрархічна структура бази даних, визначені її сутності. Розглянуто кожний об'єкт та охарактеризовані його поля.

Проаналізовано запити до бази даних та до системи в цілому. З'ясовано, що процес компіляції запиту є триетапним: парсинг (аналіз), прив'язка та оптимізація. Представлено алгоритм метасистем процедур клієнтської сторони та сторони адміністратора, які характеризують політики безпеки користувачів системи. Побудована діаграма «Сутність-зв'язок», яка відображає базу даних в цілому та зв'язки між її сутностями.

Виявлено, що для представлення таблиці як об'єкта в інтерфейсі програми-бази зручним є використання класу інструментів – фреймворк об'єктно-реляційного відображення (ORM). Проте, оскільки моделі (в узгодженні Model-View-Controller (MVC)) дуже тісно пов'язані зі схемами баз даних існує загроза несумісності моделей при зміні таких схем.

Виконані тестові завдання та результати надали змоги стверджувати, що програмно реалізована хмарна CRM-система відповідає тим вимогам, що ставляться до сучасних хмарних CRM-систем. Виявлені недоліки не викликали занепокоєння щодо стабільності роботи системи чи загрози цілісності даних. Вказані недоліки були усунені.

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		49

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У магістерській роботі здійснено теоретичне узагальнення та запропоновано нове вирішення наукового завдання, що виявилось у розв'язанні теоретичних та практичних питань побудови архітектури хмарної CRM-системи для малого підприємства. Сформовано найбільш вагомні висновки, зокрема:

Запропоновано використання наступної дефініції: хмарна CRM-система – розташована в хмарних сервісах інформаційна система управління відносинами з клієнтами, функціонал якої надає можливості збору, зберігання й аналізу інформації про споживачів, постачальників, партнерів, а також забезпечення безпеки та цілісності даних.

Побудована типова (універсальна для будь-якої хмарної CRM-системи) хмарна CRM-система на основі її основних модулів, що логічно пов'язані інформаційними потоками: інтерфейс користувача, бізнес логіка, запити CRUD, розподільовач навантаження та хмарне сховище.

Обґрунтовано, що для ефективного провадження бізнесової діяльності CRM-система має бути гнучкою та здатною до масштабування, що досягається завдяки таким параметрам як інтеграція, сумісність, мультикористувацький та багатозадачний режим, стандартизація, модульність.

Розроблений проект архітектури хмарної CRM-системи для малого підприємства «Сервіт».

Програмно реалізована та протестована хмарна CRM-система для малого підприємства «Сервіт».

					<i>КНТЕУ 121– 023-01.МР</i>			
					<i>Проектування хмарної CRM-системи для малого підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ВП</i>	<i>50</i>	<i>53</i>
<i>Зав. каф.</i>		<i>Криворучко</i>		<i>25.10.21</i>	<i>Висновки та пропозиції</i>	<i>Факультет інформаційних технологій, 2 курс, 2м група</i>		
<i>Керівник</i>		<i>Харченко О.А.</i>		<i>25.10.21</i>				
<i>Гарант</i>		<i>Токар В.В.</i>		<i>25.10.21</i>				
<i>Розроб</i>		<i>Гоголін В.В.</i>		<i>25.10.21</i>				

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про інформацію : Закон України від 02.10.1992 № 2657-XII, URL: <https://zakon.rada.gov.ua/laws/show/2657-12#Text>
2. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI, URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>
3. Про авторське право і суміжні права : Закон України від 23.12.1993 № 3792-XII, URL: <https://zakon.rada.gov.ua/laws/show/3792-12#Text>
4. An, Y., Hu, X., & Song, I. (2010). Maintaining mappings between conceptual models and relational schemas. *Journal of Database Management*, 21(3), 36–68.
5. AWS Vs Azure Vs Google: Cloud Services Comparison, URL: <https://www.whizlabs.com/blog/aws-vs-azure-vs-google>
6. Byrd, T.A and Turner, D.E. Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct. *Journal of Management Information Systems*. 17(1), 2000, p.p.167 – 208.
7. Colley D., Stanier C., Asaduzzaman Md. Investigating the Effects of Object-Relational Impedance Mismatch on the Efficiency of Object-Relational Mapping Frameworks. *Journal of Database Management* , 31(4), 2020, 23p.
8. Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM) People, process and technology. *Business Process Management Journal*, 9(5), 672–688
9. Chen P.P. The entity-relationship model – towards a unified view of data. *ACM Transactions on Database Systems*. March 1976. Vol. 1, No. 1. P. 9-36.
10. Dhaduk H. Laravel vs. Symfony: Similarities & Differences of Two PHP Frameworks. 22 Oct, 2021, URL: <https://www.simform.com/blog/laravel-vs-symfony/>
11. Duncan, N.B. Capturing flexibility of information technology infrastructure: a

					<i>КНТЕУ 121– 023-01.МР</i>			
Зм	Аркуш	№ докум.	Підпис	Дата	Проектування хмарної CRM-системи для малого підприємства	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко		27.02.21		СВД	51	53
Керівник		Харченко О.А.		27.02.21				
Гарант		Токар В.В.		27.02.21	Список використаних джерел	Факультет інформаційних технологій, 2 курс, 2м група		
Розроб		Гоголін В.В.		27.02.21				

- study of resource characteristics and their measure. *Journal of Management Information Systems*, 12, 2, 1995, p.p. 37–57.
12. Keen, P. Shaping the Future: Business Design through Information Technology. Boston: *Harvard Business School Press*, 1991, 264c.
 13. King, S. F., & Burgess, T. F. (2007). Understanding success and failure in customer relationship management. *IMMt*, 37(4), 421–431
 14. Kumar A. Understanding a SQL Server Query Execution Plan. 30 Jul, 2013, URL: <https://www.developer.com/database/understanding-a-sql-server-query-execution-plan/>
 15. Laravel vs Symfony in 2021 – Which PHP Framework Choose For Your Project? 16 Aug 2019, URL: <https://asperbrothers.com/blog/laravel-vs-symfony/>
 16. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A., “Cloud computing -The business perspective”. *DSS*, 2011, Vol. 51, No. 1, 176-189.
 17. O'Reilly K. Paper D. Customer Relationship Management (CRM): A Dichotomy of Online and Offline Activities. Encyclopedia of E-Business Development and Management in the Global Economy Volume II / In Lee, editor. Business science reference. 2010. 1458p.
 18. Payne, A. and Frow, P. (2005). A strategic Framework for Customer Relationship Management. *Journal of Marketing*. 69(4), 167-176.
 19. Prahalad, C. K. and Ramaswamy, V. The Future of Competition: Co-Creating Unique Value with Customers. Boston, Massachusetts: *Harvard Business School Press*. 2004. 257 p.
 20. Reddy S., Sarkar A. Amazon EC2 Cookbook. Packt Publishing, 2015, 195p.
 21. Reinartz, W.; Krafft, M.; Hoyer, W.D. The customer relationship management process: Its measurement and impact on performance. *J. Market. Res.* 2004, 41, 293–305.
 22. Shalender K., Kumar R., Yadav RK. Effective CRM Adoption and Implementation: The Critical Role of Flexibility. *Journal of Research in Management*, Volume 11 (1), 2019, 55 p.
 23. Van der Aalst, W.M.P., Pesic, M. & Schonenberg, H. Declarative workflows: Balancing between flexibility and support . *Comp. Sci. Res. Dev.* 23, 99–113

					<i>KHTEY 121– 023-01.MP</i>	Аркуш
						52
Вим.	Аркуш	№ докум	Підпис	Дата		

- (2009). <https://doi.org/10.1007/s00450-009-0057-9>
24. Weiss, A., “Computing in the clouds, networker”, 2007, Vol. 11, No. 4, pp. 16-25.
 25. What is Social CRM?, URL: <https://www.salesforce.com/eu/learning-centre/crm/social-crm/>
 26. Буй Д.Б., Сільвейструк Л.М. Формалізація моделі «Сутність-зв'язок»: монографія. Київ: КНУ, 2011, 175 с.
 27. Гоголін В. В., Гоголіна Г. В. CRM-система як об'єкт права інтелектуальної власності. *Підготовка фахівців у сфері інтелектуальної власності: виступи учасників IV Міжнародної науково-практичної конференції* (Київ, 20 травня 2021 р.). /Відповідальний за випуск професор Р.Б. Шишка. К.: Вид-во Ліра – К. 2021. 186 с.
 28. Гоголін В. В. Класифікація CRM-систем. Software engineering комп'ютерних систем: зб. наук. ст. студ. / відп. ред. В. Я. Рассамкін. – Київ. нац торг.-екон. ун-т, 2021. с. 35-41
 29. Марковець О. В., Паздерська Р. С. Використання новітніх технологій для забезпечення комунікації у сфері менеджменту організації. Стандартизація, сертифікація, якість. №6 (118), 2019.
 30. Матросова Н. М. Профіль інформаційної системи та інтерфейс користувача як складові методичного забезпечення інформаційної систем. *Інформаційні технології і засоби навчання*. 2010. №3 (17).
 31. Мозгова Г. В., Морозов А. О., Фомін О. Д. Використання CRM-систем на українському ринку: особливості та перспективи // *Проблеми системного підходу в економіці*. 2017. Вип. 2. С. 89-94.
 32. Пейн Э. Руководство по CRM. Путь к совершенствованию менеджмента клиентов. Минск: Гревцов Паблишер, 2007. 384 с.
 33. A reliable and cost-effective way to route end users to Internet application. URL:// <https://aws.amazon.com/>
 34. Глобальні установки CSS, фундаментальні HTML елементи стилю, підсилені розширюваними класами. URL: <https://twbs.docs.org.ua/>
 35. New Release: Doctrine ORM 2.9 with Attributes, Typed Properties, more. URL:// <https://www.doctrine-project.org/2021/05/24/orm2.9.html>

					КНТЕУ 121– 023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		53

ТЕХНІЧНЕ ЗАВДАННЯ

1. Загальна інформація

1.1. Дане технічне завдання (надалі іменоване ТЗ) було розроблено відповідно до вимог ДСТУ 34.602-89 і є основним документом, що визначає вимоги та порядок проектування і розробки архітектури хмарної CRM-системи для малого підприємства «Сервіт».

1.2. Повна назва: Архітектура хмарної CRM-системи для малого підприємства.

1.3. Крайній термін: 22.01.2021 - 20.11.2021.

1.4. Допускається внесення змін в дане ТЗ за взаємною згодою сторін відповідно до ДСТУ 34.602-89.

2. Призначення розробки

Розробка призначена для використання як програмне забезпечення операційного відділу малого підприємства «Сервіт» з метою впорядкування управління взаємовідносинами з клієнтами, планування завдань, відстеження обробки замовлень та управління бізнес-процесами підприємства.

3. Характеристика об'єкта автоматизації

Мале підприємство «Сервіт»

4. Вимоги до програмного продукту

4.1. Вимоги до функціональної частини системи як програмного продукту

Хмарна CRM-система має забезпечувати такі основні функції:

- 1) доступ до даних та їх обробка засобами хмарних технологій;
- 2) визначення політик доступу до сховища даних та їх використання;
- 3) забезпечення гнучкості та масштабованості CRM-системи;
- 4) стабільність та безпека доступу до хмарного сервісу;

					<i>КНТЕУ 121– 023-01.МР</i>			
					<i>Проектування хмарної CRM-системи для малого підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ТЗ</i>	<i>54</i>	<i>53</i>
<i>Зав. каф.</i>	<i>Криворучко О.В.</i>			<i>20.03.21</i>				
<i>Керівник</i>	<i>Харченко О.А.</i>			<i>20.03.21</i>				
<i>Гарант</i>	<i>Токар В.В.</i>			<i>20.03.21</i>				
<i>Розроб</i>	<i>Гоголін В.В.</i>			<i>20.03.21</i>	<i>Технічне завдання</i>	<i>Факультет інформаційних технологій, 2 курс, 2м група</i>		

- 5) здатність інтеграції інших сервісів, підтримка експорту даних;
- 6) модульність архітектури CRM-системи з можливістю заміни програмних складових з збереженням цілісності даних та безпековою стабільністю;
- 7) програмний продукт має бути self-explaining;

Розробка має виконуватися засобами фреймворку Symfony, з використанням сервісів AWS, мова програмування Php 7.4, ORM Doctrine, база даних – MySQL, бібліотека JavaScript – jQuery, для оперативності розробки – Bootstrap.

4.2 Вимоги до клієнтської частини

Клієнтська частина має бути здатна реалізовувати наступні функціональні можливості:

- забезпечити доступ в реальному часі до системи користувача;
- забезпечити операційну здатність використання інформації для користувача в системі;
- забезпечити виконання завдань відповідно до встановлених політик, застосовуючи різні групи доступу до інформації для різних користувачів та інші безпекові інструменти.

4.3 Вимоги до серверної частини

На стороні сервера повинні бути реалізовані наступні функції:

- виконання функції сховища даних;
- забезпечення функціонування СУБД;
- вжиття заходів безпеки з використанням сервісу AWS.

4.4 Додаткові вимоги

- 1) наявність сервісного меню;
- 2) можливість формування звітності;
- 3) можливість планування робіт на обрану дату;
- 4) наявність функцій фільтрацій та сортування даних;
- 5) дизайн сторінок з використанням як базових білого та помаранчевого кольору, кнопочке меню – блакитного кольору.

					<i>КНТЕУ 121– 023-01.MP</i>	Аркуш
						55
Вим.	Аркуш	№ докум	Підпис	Дата		

5. Вимоги до проектної документації

Системна документація повинна бути виконана у відповідності до вимог ДСТУ-34 201.

Спеціалізовані елементи системи, що поставляються підрядними організаціями, повинні супроводжуватися відповідним комплектом документації, а також (коли це можливо) сертифікатами відповідності.

6. Етапи проектування

Опрацювання літератури та інших джерел за тематикою роботи.

Розробка та узгодження технічного завдання.

З'ясування вимог до хмарної CRM-системи.

Розробка проекту архітектури хмарної CRM-системи

Програмна реалізація хмарної CRM-системи.

Тестування інформаційної системи.

Підготовка матеріалів текстової частини роботи.

Підготовка матеріалів графічної частини роботи.

Оформлення технічної документації роботи.

7. Порядок контролю та приймання системи

Випробування повинні проводитися відповідно до вимог ГОСТ 34.603-92 «Інформаційна технологія. Види випробувань автоматизованих систем»

При розробці системи виконуються наступні типи тестів:

- тестування елементів системи;
- тестування системи в цілому;

При здачі в експлуатацію інтерфейс доступу повинен пройти приймальне тестування для визначення його відповідності технічному завданню.

За результатами приймальних випробувань складається акт, в якому міститься висновок про ступінь відповідності інтерфейсу доступу вимогам технічного завдання та рішення про його прийняття в експлуатацію.

					<i>КНТЕУ 121– 023-01.МР</i>	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		56

КЕРІВНИЦТВО КОРИСТУВАЧА

Призначення CRM-системи, основні можливості та вимоги

Хмарна CRM-система призначена для автоматизації роботи операційного відділу малого підприємства «Сервіт». Здатна для занесення планувальних завдань, відстеження їх виконання, формування звітності.

CRM-система виконує наступні функції:

- 1) розподіл політик доступу за логіном та паролем;
- 2) внесення даних, їх збереження, виведення та видалення з системи;
- 3) можливість використання планувальника завдань;
- 4) можливість моніторингу етапів виконання замовлень виконавцями;
- 5) сортування даних;
- 6) можливість формування звітної документації, зокрема для бухгалтерського відділу.

Інформаційна система не потребує спеціального встановлення додатку, доступ до неї відбувається засобами веб-браузера.

Технічні вимоги: будь-який технічний прилад, здатний підтримувати сучасні веб-браузери та підключення JavaScript.

Експлуатація хмарної CRM-системи

Для доступу до системи необхідно ввести Логін та персональний пароль користувача.

					КНТЕУ 121– 023-01.МР				
					Проектування хмарної CRM-системи для малого підприємства	Стадія	Аркуш	Аркушів	
Зм	Аркуш	№ докум.	Підпис	Дата		КК	57	53	
Зав. каф.		Криворучко				Керівництво користувача	Факультет інформаційних технологій, 2 курс, 2м група		
Керівник		Харченко О.А.							
Гарант		Токар В.В.							
Розроб		Гоголін В.В.							

Логін для входу:

Якщо ви забули логін, будь ласка зверніться до служби підтримки.

Персональний пароль:

Увійти

Рис. К.1 Вхід в інформаційну систему

Після отримання доступу перед користувачем розгортається головне вікно з операційним меню. Перехід до розділів системи відбувається шляхом натискання кнопок меню «План роботи», «Техніка», «Водії», «Виконроб», «Об'єкти», «Відомості».

Сервіс EMS							
Своя техніка							
#	Техніка	Водій	Телефон	ВиконРоб	Характеристика роботи	Години роботи	
1	Бобкет JCB - 260	Адімови Юрій		Червак Олег	Детярівська :	Д-8	  
2	Екс. ATLAS 1404K	підміна Москаль Сергій		Франкович	Сергій Колоса 167 :	Д-8	  
3	Екс. JCB 160W (2021)	Васинович Михайло		Червак Олег	Детярівська :	Д-8	  
4	Екс. JCB 160W (66511AA)	Бендик Олександр		Олександр 066-720-01-02	Леваневського :	Д-8	  
5	Екс. JCB 175W (Семена)	Труш Олександр		Штольневичи	Детярівська :	Д-8	  
6	Екс. JCB 3CX 2020 NEW (1)	Гончар Олег		Калин Сергій	Леваневського :	Д-4	  
7	Екс. JCB 3CX 2020 NEW (1)	Гончар Олег		Мельничук Юрій	Леваневського :	Д-4	  
8	Екс. JCB 3CX 2020 NEW (2)	Тришук Олександр		Позил Михайло	Радистів 40 (Биківня) :	Д-8	  
9	Екс. JCB 3CX Contractor (61767 AA)	Петрик Анастасій		Мельничук Юрій	Леваневського :	Д-5	  
10	Екс. JCB 3CX Contractor (61767 AA)	Петрик Анастасій		Силичук Віктор	Щербаківського 61-Е :	Д-3	  

Рис. К.2 Головне вікно CRM-системи

Кожна з кнопок навпроти запису надає можливість відслідкувати чи скорегувати дані за поточною операцією, зокрема



- надає можливість відредагувати дані в планувальнику завдань. По її натисканню відкривається вікно (див. рис К.3)

					КНТЕУ 121– 023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		58

Редагування планової роботи

Техніка:

Бобкет JCB - 260

Статус техніки:

Своя

Водій:

Виконаючий Роботи:

Червак Олег

Об'єкт:

Детярівська

Характеристика роботи:

Закрити

Виділити

Зберегти

Рис. К.3.— Редагування планової роботи

Кнопка  – надає можливість відредагувати технічні витрати (рис. К.4)

Редагування технічних витрат

Початок

Кінець

Кілометраж

Мотогодини

Бак, л

Заправлено, л

Витрачено, грн

Ціна за 1л, грн

Газ

Бензин

Дизель

Закрити

Зберегти

Рис. К.4 – Редагування технічних витрат

Кнопка  – надає можливість відредагувати часові витрати

					КНТЕУ 121– 023-01.MP	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		59

Редагування часових витрат

Без обіду

День

8

Вечір

Ніч

Ремонт

База

Сервіс

Закрити

Зберігти

Рис. К.5 – Редагування часових витрат

Звітна документація формується за допомогою меню «Відомості».

Сервіс EMS

Тільки роботи

Оберіть дату

Техніка

Водії

ВиконРоб

Об'єкти

Відомості

Початкова дата

2021-11-08

Кінцева дата

2021-11-08

Обрати період

Об'єкти в роботі

База

Берестівка

Благоустрій КТЕ

Відрядний

Гідропарк

Депутатська

Депутатська 2

Дрвляніська

Електрика 17

Жуляни (Стадіон)

Техніка

Автокран (Госа)

Автокран АА0700TX

Bobcat JCB - 260

Exc. ATLAS 1304

Exc. ATLAS 1404K

Exc. ATLAS 1504

Exc. JCB 160W (2021)

Exc. JCB 160W (66511AA)

Exc. JCB 175W

Exc. JCB 175W (Семена)

Виконуючі роботи

Анатолій (Веселова)

Ара

Арто

Батіньский Igor

Беспальчук Іван

Богущий Володимир

Вадим бетон

Вадим благоустрій

Василь МЕС Буд

Василь Михайлович

Водії

Адамович Юрій

Бабій Юра

Безпальний Олександр

Бендик Олександр

Богословець Микола

Бушула Олег

Васанович Антон

Васанович Михайло

Веретеников Михайло

Гаврилюк В.В.

Застосувати

Застосувати

Застосувати

Застосувати

Сформувати звіт на екран

Рис. К.6 Формування звітної документації

По натисканню Меню «Техніка», «Водії», «Виконуванні роботи» розгортаються дані за даним напрямком. В перелічених об'єктах наявна лише одна кнопка:



«Редагувати». По її натисканню розгортається меню редагування обраного об'єкта.

					КНТЕУ 121– 023-01.МР	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		60

Редагування техніки

×

Назва:

Екс. JCB 160W

Ціна:

0.00

Водій:

Останній ТО:

2020-08-05

Страховка:

2020-08-05

Закрити

Видалити

Зберігти

Рис. К.7 Редагування об'єкта «Техніка»

Вихід з системи здійснюється натисканням кнопки [Вийти](#) внизу вікна системи.

					КНТЕУ 121– 023-01.MP	Аркуш
Вим.	Аркуш	№ докум	Підпис	Дата		61

ДОДАТКИ

Додаток А

Лістинги

/////////# routes.yml////////

route1:

path: /

defaults: { _controller: '\Controllers\DefaultController::index' }

methods: GET

route2:

path: /404

defaults: { _controller: '\Controllers\DefaultController::notFound' }

methods: GET

route3:

path: /endpoints/{endpoint}

defaults: { _controller: '\Controllers\Endpoints::entryPoint' }

methods: POST

route4:

path: /admin

defaults: { _controller: '\Controllers\DefaultController::admin' }

methods: GET

route4.1:

path: /admin/{date<\d{4}-\d{2}-\d{2}>}

defaults: { _controller: '\Controllers\DefaultController::admin' }

methods: GET

route5:

path: /test/{date<\d{4}-\d{2}-\d{2}>}

defaults: { _controller: '\Controllers\Test::uiTest' }

methods: GET

route5.1:

path: /test

Продовження додатка А

```
defaults: { _controller: '\Controllers\Test::admin' }

methods: GET

route6:

path: /admin/technics

defaults: { _controller: '\Controllers\Admin::technics' }

methods: GET

route7:

path: /admin/drivers

defaults: { _controller: '\Controllers\Admin::drivers' }

methods: GET
```

////////Admin.php////////

```
<?php
namespace Controllers;

use \Modules\Http;
use \Modules\Helpers\TemplatesParser;
use \Modules\Helpers\PathFinder;
use \Entities\Entity;

class Admin
{
    public static function checkLogged()
    {
        if (!Http\Session::isLogged()) {
            $response = new Http\Response('Not logged');
            $response->redirect('/');
            exit(0);
        }
    }

    public static function technics($returnObject = false)
    {
        self::checkLogged();
        $data = array();
        $technic = Entity::get(Entity::ENTITY_TECHNICS);
        $technics = $technic->findAll();
        foreach ($technics as $row) {
            array_push($data, array(
```

```

'id' => $row->getId(),
'cost' => sprintf("%01.2f", $row->getCost()),
'name' => $row->getName(),
'driverId' => $row->getDriver() != " ? $row->getDriver() : '0',
'driverName' => $row->getDriverName(),
'inspection' => $row->getLastinspection()->format('Y-m-d'),
'insurance' => $row->getEndofinsurance()->format('Y-m-d')
));
}
if ($returnObject) {
    return $data;
}

$templatesParser = new TemplatesParser();
$html = $templatesParser->render('admin/technics', array('technics' => $data));
return new Http\Response($html);
}

public static function drivers($returnObject = false)
{
    self::checkLogged();
    $data = array();
    $driverState = Entity::get(Entity::ENTITY_DRIVER_STATE);
    $driverStates = $driverState->findAll();
    $dataStates = array();
    foreach($driverStates as $row){
        $dataStates[$row->getId()] = $row->getName();
    }
    $driver = Entity::get(Entity::ENTITY_DRIVER);
    $drivers = $driver->findAll();
    foreach ($drivers as $row) {
        array_push($data, array(
            'id' => $row->getId(),
            'name' => $row->getName(),
            'phone' => $row->getPhone(),
            'statusId' => $row->getState(),
            'statusName' => $row->getState() > 0 ? $dataStates[$row->getState()] : "",
        ));
    }
    if ($returnObject) {
        return $data;
    }
}

```

```

$templatesParser = new TemplatesParser();
$html = $templatesParser->render('admin/drivers', array('drivers' => $data));
return new Http\Response($html);
}

public static function objects($returnObject = false)
{
    self::checkLogged();
    $data = array();
    $object = Entity::get(Entity::ENTITY_OBJECT);
    $objects = $object->findAll();
    foreach ($objects as $row) {
        array_push($data, array(
            'id' => $row->getId(),
            'name' => $row->getName(),
            'address' => $row->getAddress(),
            'phone' => $row->getPhone(),
        ));
    }
    if ($returnObject) {
        return $data;
    }

    $templatesParser = new TemplatesParser();
    $html = $templatesParser->render('admin/objects', array('objects' => $data));

    return new Http\Response($html);
}

public static function services($returnObject = false)
{
    self::checkLogged();
    $data = array();
    $service = Entity::get(Entity::ENTITY_SERVICE);
    $services = $service->findAll();
    foreach ($services as $row) {
        array_push($data, array(
            'id' => $row->getId(),
            'code' => $row->getCode(),
            'name' => $row->getName(),
        ));
    }
    if ($returnObject) {
        return $data;
    }
}

```

```

$templatesParser = new TemplatesParser();
$html = $templatesParser->render('admin/services', array('services' => $data));
return new Http\Response($html);
}

public static function statements()
{
    self::checkLogged();
    $templatesParser = new TemplatesParser();
    $html = $templatesParser->render('admin/statements', array(
        'objects' => self::objects(true),
        'technics' => self::technics(true),
        'services' => self::services(true),
        'drivers' => self::drivers(true),
    ));
    return new Http\Response($html);
}
}

```

//////////Driver.php//////////

```

<?php
use \Doctrine\ORM\Mapping as ORM;
/**
 * @ORM\Entity
 * @ORM\Table(name="driver")
 */
class Driver extends AbstractEntity
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue
     * @var int
     */
    protected $id;

    /**
     * @ORM\Column(type="string",length=1024)
     * @var string
     */
    protected $name;

    /**
     * @ORM\Column(type="string",length=32)

```

```

* @var string
*/
protected $phone;
/**

* @ORM\Column(type="integer",length=10)
* @var int
*/
protected $_state;
/**
* @ORM\Column(type="datetime", name="created", options={"default":
"CURRENT_TIMESTAMP"})
* @var DateTime
**/
private $created;

public function getId()
{
    return $this->id;
}
public function getName()
{
    return $this->name;
}
public function setName($name)
{
    $this->name = $name;
}
public function getPhone()
{
    return $this->phone;
}
public function setPhone($phone)
{
    $this->phone = $phone;
}
public function getState()
{
    return $this->_state;
}
public function setState($_state)
{
    $this->_state = $_state;
}

```

```

    }
}

///////// TemplatesParser.php ///////////

<?php
namespace Modules\Helpers;

use \Twig\Environment;
use \Twig\Loader\FilesystemLoader;

class TemplatesParser
{
    const PARSING_ERROR_TEXT = "Template not found ";

    private $twig;

    public function __construct()
    {
        $templatesFolder = PathFinder::getViewsFile('.');
        $loader = new FilesystemLoader($templatesFolder);
        $this->twig = new Environment($loader);
    }

    public function render($template, $replaces = array())
    {
        try {
            $html = $this->twig->render($template . '.twig', $replaces);
        } catch (\Exception $e) {
            $html = self::PARSING_ERROR_TEXT . $template;
        }

        return $html;
    }

    public function addFilter(string $name, $callable = null, array $options = [])
    {
        $this->twig->addFilter(new Twig_Filter($name, $callable, $options));
    }
}

```