Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Платформа для інтеграції CRM-системи «ZOHO»

(тема)

Виконав:

студент II курсу, групи КСМзм-20-1

Кравченко Р.О.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп’ютерні системи та мережі
(повна назва освітньої програми)

Керівник: проф. Кучук Н.Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Кравченко Ростиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Платформа для інтеграції CRM-системи «ZOHO»

затверджена наказом по університету від " 25 " жовтня 2021 р. № 169 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2021 р.

3. Вхідні дані до роботи 1) існуючі CRM-системи; 2) локального сервіс Open Server; 3) мова програмування C#; 4) фреймворк ASP.NET Core

4. Перелік питань, що потрібно опрацювати у роботі
   1) аналіз предметної області;
   2) проектування середовища функціонування;
   3) проектування системних взаємозвязків;
   4) тестування;
   5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 15 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз CRM-систем, які існують	26.10.21-05.11.21	
2	Встановлення вимог до цієї платформи.	06.11.21-15.11.21	
3	Розробити платформу для інтеграції з		
	CRM-системою «ZOHO»	16.11.21-22.11.21	
4	Основні етапи технології розробки	23.11.21-29.11.21	
	веб-платформи		
5	Тестування розробки	23.11.21-27.11.21	
6	Оформлення матеріалів кваліфікаційної роботи	29.12.21-05.12.21	
7	Подання кваліфікаційної роботи керівникові та	06.12.2021	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	08.12.2021	
	вання		

Дата видачі завдання 25 жовтня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кучук Н.Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 74 с., 14 рис., 29 табл., 21 джерело

ПЛАТФОРМА, ІНТЕГРАЦІЯ, БАЗА ДАНИХ, C#, ФРЕЙМОРК, ВЕБ-ДОДАТОК.

Об’єктом дослідження є CRM-системи та інтеграція з ними.

Метою дослідження є розробка платформи для інтеграції з CRM-системою «ZOHO».

У роботі досліджені CRM-системи які існують на даний момент, основні функції які потрібні для цієї платформи, встановлені вимоги до цієї платформи, етапи технології розробки веб-платформи, обґрунтовано вибір локального сервісу, мови програмування, фреймворку та середовища програмування, проведено функціональне тестування веб-додатку.

ABSTRACT

Master’s thesis: 74 pages, 14 figures, 29 tables, 21 sources.

PLATFORM, INTEGRATION, DATABASE, C#, FRAMEWORK, WEB-APPLICATION.

The object of research is CRM systems and integration with them.
The aim of the research is to develop a platform for integration with the CRM-system "ZOHO".

The work investigates the CRM systems existing at the moment, the main functions necessary for this platform, the established requirements for this platform, the stages of the web platform development technology, the choice of a local server, programming language, framework and programming environment is justified, based on these studies, an integration platform was developed and tested.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

ІТ – інформаційні технології

СКБД – система керування базами даних

CRM – управління відносинами з клієнтами (англ., Customer Relationship Management)

EF – реальний фремворк (англ., Entity Framework)

FTP – протокол передачі файлів (англ.,File Transfer Protocol)

HTTP – протокол передачі гіпертексту (англ.,HyperText Transfer Protocol)

JSON – нотація об’єктів JavaScript (англ.,JavaScript Object Notation)

VoIP – голос через Інтернет-протокол (англ.,Voice over Internet Protocol)

WWW – всесвітня мережа (англ.,World Wide Web)

ВСТУП

Предметна область: робота присвячена розробці та дослідженню платформи для інтеграції з CRM системою.

У кваліфікаційній роботі була розроблена платформа для інтеграції з CRM системою. На даний момент схожі веб-додатки не вдалося знайти.

Ця система розроблена для того, що б зробити запит на CRM систему, використовуючи надані нам пароль і логін, завдяки цьому CRM система нам повертає дані, у вигляді JSON об’єктів, які ми перетворюємо в лист класів. Після ми зберігаємо ці дані в базу даних.

Leads – це сутності, які вже зареєстровані в програмі, але ще не відкрили свій рахунок і не робили ніяких транзакцій.

Accounts – це сутності, які представляють собою, що вже зареєстровані та внесли деякі суми на рахунок.

Refferals – це сутності, які вже старовинної використовують систему і запросили лідов і отримують відсоток від їх транзакцій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальні положення

З появою Web-технологій комп’ютери починають використовувати абсолютно нові верстви населення Землі. Можна виділити дві найбільш характерні групи, що знаходяться на різних соціальних полюсах, які були стрімко залучені в нову технологію, можливо, навіть супротив їх власного бажання. З одного боку, це були представники елітарних груп суспільства – керівники великих організацій, президенти банків, топ-менеджери, впливові державні чиновники і т.д. З іншого боку, це були представники найширших верств населення – домогосподарки, пенсіонери, діти.

При появі технології Web, комп’ютери повернулися обличчям до цих двох абсолютно протилежних категорій потенційних користувачів. Еліту об’єднувала одна риса – в силу високої відповідальності і практично стовідсотковою зайнятості "великі люди" ніколи не користувалися комп’ютером. Типовою була ситуація, коли з комп’ютером працював секретар. У якийсь момент часу вони зрозуміли, що комп’ютер їм може бути корисний, що вони можуть результативно використовувати той невеликий час, який можна виділити для роботи за комп’ютером. Вони раптом зрозуміли, що комп’ютер – не просто модна і дорога іграшка, але інструмент отримання актуальної інформації для бізнесу. При цьому їм не потрібно було витрачати скільки-небудь помітного часу, щоб освоїти технологію роботи з комп’ютером (у порівнянні з тим, як це було раніше) [1].

Спектр соціальних груп, що підключаються до мережі Інтернет і що шукають інформацію в WWW, весь час розширюється за рахунок користувачів, що не відносяться до категорії фахівців в області інформаційних технологій: лікарі, будівельники, історики, юристи, фінансисти, спортсмени, мандрівники, священнослужителі, артисти, письменники, художники. Список

можна продовжувати нескінченно. Той, хто відчув корисність і незамінність Мережі для своєї професійної діяльності або захоплень, приєднується до величезної армії споживачів інформації у «Всесвітньому Павутинні».

Web-технології повністю перевернули наші уявлення про роботу з інформацією і з комп'ютером. Виявилося, що традиційні параметри розвитку обчислювальної техніки – продуктивність, пропускна здатність, ємність запам'ятовуючих пристроїв, не враховували головного: «вузького місця» системи -– інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І тільки коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, відбувся безпрецедентний вибух інтересу до можливостей обчислювальної техніки.

З розвитком технологій гіпертекстової розмітки в Інтернеті стало з'являтися все більше сайтів, тематика яких була абсолютно різною – від сайтів великих компаній, що оповідають про успіхи компанії і її провали, до сайтів маленьких фірм, що пропонують відвідати їх офіси в межах одного міста.

Розвиток Інтернет технологій став поштовхом до появи нової гілки в Інтернеті – Інтернет форумів. Стали з'являтися сайти, і навіть цілі портали, на яких люди з усіх куточків планети можуть спілкуватися, отримувати відповіді на різні питання і, навіть, укладати ділові угоди.

Web-сайти використовуються як механізм спілкування між власниками сайту і його користувачами, а, іноді – між самими користувачами. Власники сайтів зазвичай ставлять завдання і визначають основні правила взаємодії, в той час як користувачі – це ті люди, що відвідують сайт і намагаються користуватися представленим на ньому вмістом або його можливостями. Канал зв'язку між власником сайту і його відвідувачем може змінюватися. Найчастіше власники сайтів надають користувачам інформацію для її споживання, роблячи з цього частково односторонню взаємодію.

Web-сайти можна розсортувати за кількома широкими категоріями.

Інформаційні сайти. На таких сайтах представлена інформація щодо конкретної теми або про певну організацію. Це самі поширені в мережі Internet Web-сайти.

Операційні сайти. Сайтом такого типу можна скористатися з метою виконання будь-якої операції або завдання. У цю категорію входять сайти, зайняті в електронній комерції.

Сайти спільнот. На цих сайтах представлена інформація або кошти, пов’язані із здійсненням операцій, але упор робиться на взаємодію між відвідувачами. Сайти, засновані на спільнотах, мають тенденцію до фокусування на конкретній темі або людині; вони заохочують взаємодію між схоже мислячими особистостями.

Розважальні сайти. Ці сайти створюються для ігор або якогось цікавої взаємодії, для якої можуть вживатися елементи операційного, інформаційного типів і сайтів спільнот.

Інші сайти. У цю категорію входять художні або експериментальні сайти, особисті Web-простори, такі як Web-журнали, а також сайти, які можуть не слідувати загальноприйнятим Web-угодам або не мати чітко певного економічного призначення.

Крім цього, можна згрупувати сайти на основі організацій, які підтримують або в якомусь сенсі платять за сайт. В рамках цього типу класифікації визначимо п’ять основних груп, що перераховані нижче.

Корпоративні сайти. Сайт з цієї групи створюється і підтримується організацією або індивідуумом для отримання комерційної вигоди - або безпосередньо за допомогою електронної комерції, або побічно через стимулювання придбання товарів або послуг поза Internet.

Урядові сайти. Вищим органом по відношенню до такого сайту є урядова організація, а призначенням сайту є задоволення будь-якої суспільної або правової потреби.

Освітні сайти. Сайт такого типу курирує якась освітня установа (можливо, що має відношення до урядових органів).

Філантропічні сайти. Філантропічний сайт існує з метою просування цілей некомерційної організації або благодійної діяльності приватної особи або організації.

Персональні сайти. Такий сайт існує виключно на розсуд якоїсь людини або групи людей по будь-яким причинам, зазвичай будучи плодом виплеску творчої енергії або формою самовираження особистості;.

Класифікація може виявитися складним завданням. Наприклад, освітні сайти насправді можуть потрапляти в категорію урядових. Деякі сайти з категорії персональних можуть, ймовірно, належати до групи філантропічних або комерційних – в залежності від причини, за якою людина береться за створення Інтернет ресурсу.

Сьогоднішні підприємства залежні від IT. Кожна частина бізнесу тепер потребує підключення до Інтернету для роботи, а не тільки для фільмів під час перерв; від спілкування електронною поштою, обміну миттєвими повідомленнями та VoIP, до бек-офісу ERP та CRM – не кажучи вже про важливість каналів цифрового маркетингу та електронної комерції. На CRM системах мі зупинимося більш детальніше.

CRM (Customer Relationship Management – Управління взаємовідносинами з клієнтами) це не програмний продукт і не технологія. Це навіть не набір продуктів. CRM – це спрямована на побудову стійкого бізнесу концепція і бізнес стратегія, ядром якої є "клієнто-орієнтований" підхід.

Ця стратегія заснована на використанні передових управлінських і інформаційних технологій, за допомогою яких компанія збирає інформацію про своїх клієнтів на всіх стадіях його життєвого циклу (залучення, утримання, лояльність), витягає з неї знання і використовує ці знання в інтересах свого бізнесу шляхом вибудовування взаємовигідних відносин з ними.

Результатом застосування стратегії є підвищення конкурентоспроможності компанії і збільшення прибутку, так як правильно побудовані відносини, засновані на персональному підході до кожного клієнта, дозволяють залучати нових клієнтів і допомагають утримати старих.

CRM системи стали потрібні на високо конкурентному ринку, де у фокусі стоїть клієнт. Головне завдання CRM систем – підвищення ефективності бізнес процесів, зосереджених у "фронт-офісі", спрямованих на залучення і утримання клієнтів – у маркетингу, продажах, сервісі й обслуговуванні, незалежно від каналу, через який відбувається контакт з клієнтом.

З точки зору управління бізнесом ефект від впровадження CRM виявляється в тому, що процес прийняття рішення за рахунок автоматизації переноситься на більш низький рівень і уніфікується. За рахунок цього підвищується швидкість реакції на запити, росте швидкість обороту коштів і знижуються витрати.

Нарешті, CRM включає в себе ідеологію і технології створення історії взаємин клієнта і фірми, що дозволяє більш чітко планувати бізнес і підвищувати його стійкість.

Витрати на залучення нового клієнта в середньому в п’ять разів більше, ніж на утримання існуючого.

Велика частина компаній зі списку Fortune 500 втрачає 50% своїх клієнтів кожні 5 років.

Задоволений клієнт розповість про вдалу покупку в середньому 5 своїм знайомим. Незадоволений клієнт розповість про поганий послузі або купівлі мінімум 10 знайомим.

Збільшення відсотка утримання клієнтів на 5% збільшує прибуток компанії на 50-100%.

В середньому компанія контактує 4 рази на рік з існуючим клієнтом і 6 разів на рік з потенційним.

Постачальники продуктів класу CRM обіцяють підвищення прибутковості підприємств на десятки відсотків, а рентабельність проектів - від 200 до 800 відсотків за 2-3 роки.

Базова концепція CRM (фокус не на продукті, а на клієнті + персоналізація) йде корінням в минуле.

Класичний приклад: коли супермаркетів не було, то основна маса товарів продавалася через безліч маленьких магазинчиків. Господар магазину знав усіх своїх клієнтів, які жили в по сусідству, в обличчя і по імені. Знав їх потреби, звички, смаки, фінансовий стан, факти особистого життя і т.д. Він знав хто, коли і навіщо прийде. Бізнес будувався на лояльності цих постійних покупців. Зараз це б назвали персоналізацією.

Потім настала ера споживання, виросли супермаркети, масовий продукт, масовий покупець. Все – якісно та все гарно, продається на кожному розі, але без персоналізації. Про персоналізації - забули. Адже не можна приставити до кожного покупця по продавцю.

В епоху конкуренції якість товару скрізь приблизно однаково. Норма прибутку - впала. Єдиний спосіб вижити в конкурентній боротьбі - виділитися серед інших продавців товарів і послуг, запропонувати продукт кожного клієнта персонально, з урахуванням його індивідуальних потреб і особливостей.

І тут виявилося, що на сучасному рівні розвитку комп’ютерних технологій можна "повернутися до минулого" і забезпечити персоналізацію навіть в масових продажах. Господар магазину раніше зберігав в голові інформацію про ста своїх клієнтів. База даних може зберегти і обробити сто тисяч. І запропонувати кожному саме те, до чого він звик, і що може захотіти.

1.2 Аналіз існуючих CRM систем та їх порівняння

За допомогою Zoho CRM (рисунок 1.1) можна отримати панорамне уявлення про свій бізнес, відстежувати основні можливості продажів і маркетингу, а також підвищувати конверсію запитів в продажу. Маючи більше 150 000 клієнтів по всьому світу, Zoho CRM є одним з найпопулярніших інструментів в цьому списку [3].

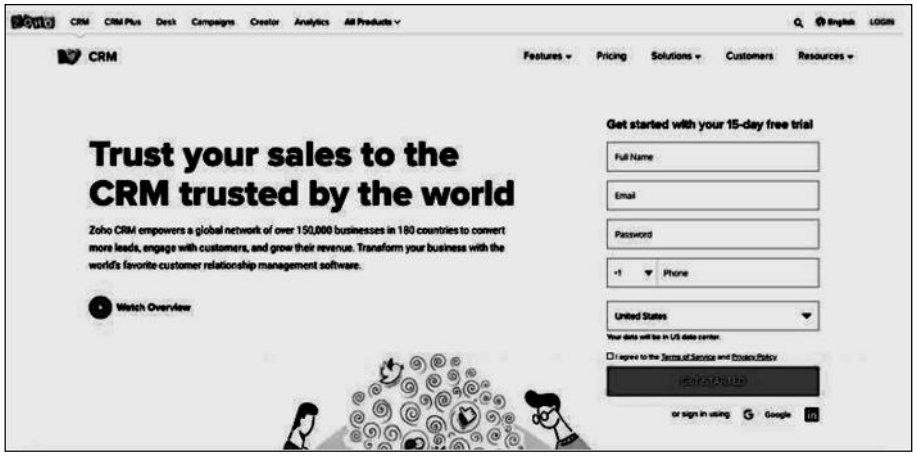

Рисунок 1.1 – Зовнішній вигляд Zoho CRM.

Ключові характеристики:

Advanced CRM Analytics (Просунута CRM - аналітика) – інтегрує ваші дані і виводить виконувані аналітичні дані.

AI-Powered Sales Assistant (Помічник з продажу на основі ІІ-інтелекту) – Zoho допомагає відділам продажів і представникам приймати рішення на основі профілю клієнта і виявляти аномалії процесів.

Performance Management (Управління продуктивністю) – за допомогою таких функцій, як Гейміфікація (ділова гра), звітність, прогнозування продажів і управління територіями продажів, ви можете максимально ефективно використовувати свої витрати на малий бізнес.

Sales Enablement (Можливість продажу) – допомагає вам генерувати цінові пропозиції та отримувати доступ до сценаріїв продажів.

Process management (Керування процесами) – дозволяє команді планувати кожен крок процесу продажів за допомогою будівника продажів.

Плюси:

- багатоканальна підтримка клієнтів по телефону, чату, електронної пошти та соціальних мереж;
- забезпечує високий рівень автоматизації;
- настроюваність під запити користувача;
- понад 100 сторонніх інтеграцій, включаючи LinkedIn, Zapier,

інтеграції пошти з такими поштовими клієнтами, як Google і Outlook, і багато іншого;

- високомасштабованість;
- 15-денний безкоштовний пробний період;
- пропонує мобільну версію (для платних планів);
- забезпечує безпеку даних, і відповідність нормативним вимогам;
- пропонується багатомовних інтерфейс;
- має безкоштовний тарифний план для 3 користувачів;
- забезпечує цілісне управління календарем;
- пропонує імпорт / експорт даних.

Функція прогнозу продажів допомагає в збагаченні даних, в аналізі налаштувань електронної пошти та в багатьох інших аспектах.

HubSpot CRM (рисунок 1.2) – це повнофункціональний маркетинговий пакет, а його програмне забезпечення CRM є найпопулярнішим пропозицією.

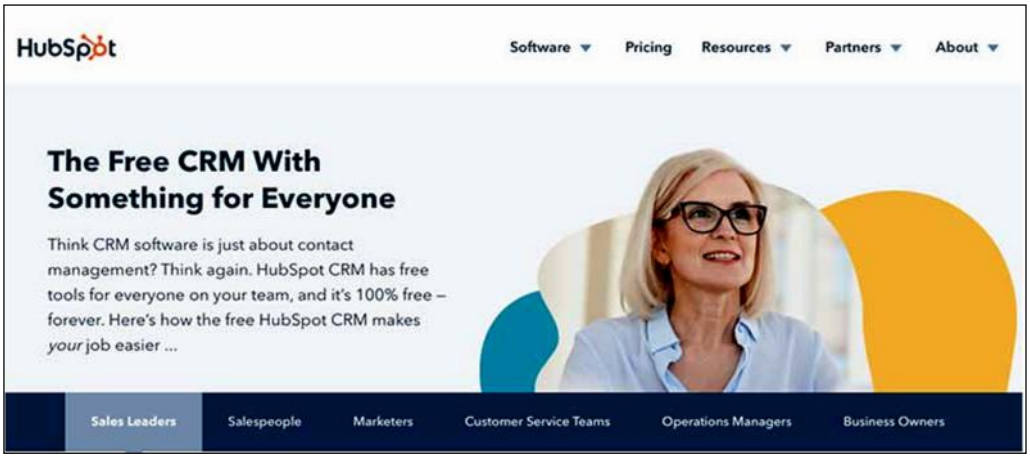

Рисунок 1.2 – Зовнішній вигляд HubSpot CRM

HubSpot CRM відмінно підходить для фахівців з продажу завдяки своїм відмінним функцій і інструментів економії часу. Це також хороший варіант для маркетингових команд з можливістю захоплення, відстеження і зростання лидов в базі даних. Вони пропонують 100% безкоштовну версію свого програмного забезпечення, з можливістю оновлення та отримання доступу до

всього набору маркетингових інструментів.

Ключові характеристики:

- email and lead nurturing (Електронна пошта і виховання лідерів) – ви можете відправляти безкоштовні персоналізовані електронні листи і відстежувати свої результати за допомогою вбудованих аналітичних звітів;
- contact organization and management (організація і управління контактами) – ви можете підключити свій поштовий ящик до цього розділу CRM і автоматично синхронізувати і організовувати свої контакти в одному місці;
- over 300 integrations (більше 300 інтеграцій з іншим ПО) – легко підключайте існуючий технологічний стек (пристрої магазинного типу) і отримуйте більше функціональності від інструментів, за які ви вже заплатили;
- live chat and chatbots (живий чат і чат-боти) – допоможіть клієнтам частіше користуватися запитами в службу підтримки, інструментами підвищення продуктивності і можливостями миттєвого чату;
- sales pipeline data (дані конвеєра продажів) – лідери продажів можуть переглядати весь свій конвеєр продажів з інформацією про діяльність з продажу і індивідуальної продуктивності.

Плюси:

- безкоштовний план, який ви можете оновити в будь-який час;
- інтегровано з наборами продажів і маркетингу, які включають в себе живий чат, цільові сторінки, маркетинг по електронній пошті, управління рекламою, управління документами і багато іншого;
- безкоштовний маркетинг по електронній пошті і сегментація списків;
- командна електронна пошта і входять розмови для доступу до всіх повідомлень в одному місці;
- відмінний конструктор форм для збору інформації про лідах.

Facebook, Instagram, LinkedIn та інші інструменти управління рекламою для відстеження рентабельності інвестицій в платних кампаніях Google, Facebook, Instagram, LinkedIn та інших соціальних мережах. Freshsales

(рисунок 1.3) – це заснована на штучному інтелекті CRM для оцінки лидов, активності електронної пошти, захоплення електронної пошти і так далі. Одна з його найбільш помітних особливостей полягає в тому, що він забезпечує панорамний огляд вашого бізнесу [5]:

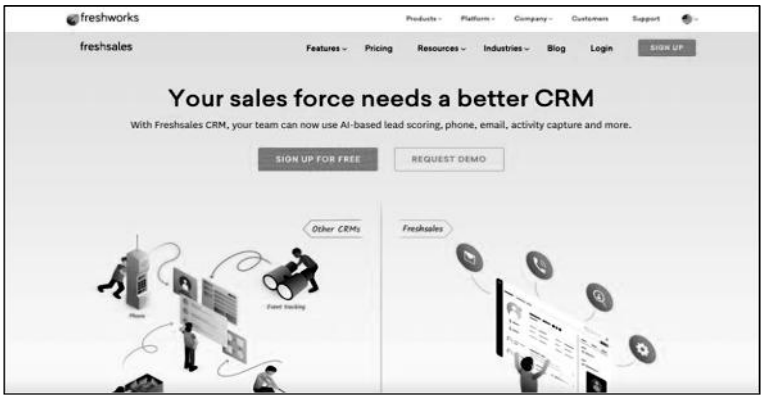

Рисунок 1.3 – Зовнішній вигляд Freshsales

Ключові характеристики:

- lead management (управління лідамі) – надає рішення для перетворення потенційних лідів в клієнтів;

- deal management (управління операціями) – дає вам повну картину того, де угода поміщається в воронку продажів. крім того, це допоможе вам керувати і організовувати кожен крок в цій воронці і максимізувати ваші шанси на конверсію;

- tracking and lead scoring (відстеження і оцінка лидов) – за допомогою вбудованої функції штучного інтелекту ви можете ранжувати і оцінювати ліди, а також визначати ті з них, які з найбільшою ймовірністю здійснять покупку;

- auto lead assignment (автоматичне призначення лидов) – crm дозволяє вручну відібраним лідам автоматично вибирати продавців;

- smartforms (web-to-lead) (розумні форми (від web інформації до ліду) - коли людина заповнює веб-форму на вашому сайті, він автоматично стає контактом і додається в вашу базу даних.

Плюси:

- забезпечує високий рівень автоматизації і інтелектуальні робочі процеси;

- широкий спектр інтеграцій, включаючи mailchimp, zapier, calendar, piesync, segment і багато іншого;

- restful api допоможуть вам читати, змінювати, додавати і видаляти дані з довідкової служби. [10]

Сегментація лидов на основі їх поведінки:

- розширена оцінка лидов, заснована на характеристиках і поведінці;
- розсилка персоналізованих вітальних і голосових повідомлень;
- сумісно з gdpr (регламентом захисту персональних даних);
- 21-денний безкоштовний пробний період;
- відмінна підтримка клієнтів по телефону та електронною поштою;
- відомий, як один з кращих мобільних crm з мобільним додатком;
- відстеження подій (відстежує попередню комунікацію, щоб спланувати майбутній підхід);
- дуже гнучкий у налаштуванні.

Salesforce (рисунок 1.4) є одним з найвідоміших імен у цьому списку і має більше мільйона користувачів по всьому світу. Це хмарне CRM-програмне забезпечення, яке обслуговує всі галузі бізнесу, включаючи продажі, сервіс, маркетинг, аналітику і багато іншого.

Ключові характеристики:

- opportunity management (управління можливостями) – ви можете закрити більше угод, визначивши ідеальні можливості за допомогою модуля "управління можливостями";

- contact management (управління контактами) – за допомогою цього модуля ви можете відстежувати такі параметри, як історія клієнтів, комунікації і згадки в соціальних мережах;

- sales performance management (управління ефективністю продажів) – встановлюйте цілі продажів і оновлюйте цільові показники.

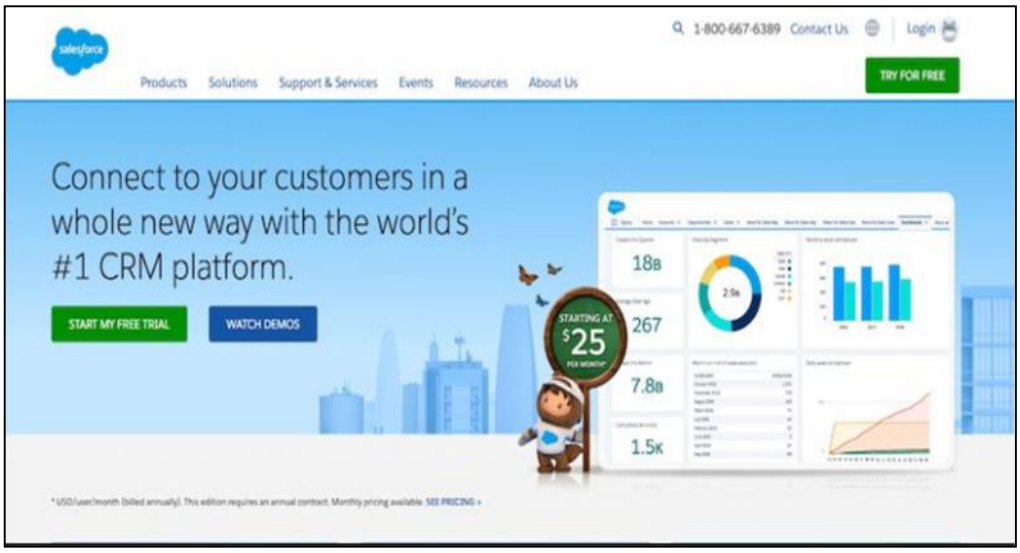

Рисунок 1.4 – Зовнішній вигляд SalesForce

Плюси:

- 24/7 (цілодобова) підтримка клієнтів доступна по телефону, електронній пошті і зверненнями;
- високорівнева автоматизація і персоналізація;
- 30-денний безкоштовний пробний період;
- хмарний хостинг (швидше і дешевше);
- сумісність з linux, windows, mac, android;
- відкритий арі (тому дуже легко настроюється);
- пропонує управління всією сферою підприємства, crm-систему продажів і управління соціальними мережами;
- багатомовна і мультивалютна підтримка клієнтів;
- широкий вибір шаблонів електронної пошти на вибір;
- вбудований ейнштейн веде підрахунок лидов;
- управління лідамі за допомогою автоматизації інформацією відділу продажів і бізнес-аналітики;
- сумісний з усіма пристроями, включаючи мобільні телефони;
- надає партнерські спільноти;
- вбудований конструктор процесів і інструменти генерації лідів;

- велика кількість сторонніх інтеграцій (leadexec, ambassador, zuora, zendesk і ін.);
- масова розсилка по електронній пошті допоможе вам заощадити час і зусилля[12].

Недоліки:

- не може використовуватися компаніями, які хочуть тільки локальне розгортання;
- підтримка живого чату недоступна.

1.3 Формування вимог до працездатності платформи

Розроблюване програмне забезпечення відноситься до сервісів, які дають змогу отримати статистику через ендпоінти які надає CRM система "ZOHO". Користування сервісом не повинно обмежуватися геолокацією користувача і різновидами доменами сайтів.

Сервіс має працювати таким чином: користувач може користуватися цією системою якщо адміністратор зареєстрував цього користувача у сервісі, підтвердити свій телефон вказаний при реєстрації (для того щоб запобігти використання «фейкових» облікових записів). Після підтвердження треба успішно пройти процедуру авторизації, далі користувач потрапляє на головну сторінку системі, де має змогу відстежувати статистику для свого облікового запису, або підконтрольних йому користувачів. Також, для кожного доданого користувача повинні бути: налаштування розділів інформації до яких у користувача дозволи.

Розроблюваний сервіс буде виконувати наступні функції:

- реєстрація користувача;
- авторизація користувача;
- редагування даних користувача;
- додавання або видалення розділ який має право бачити користувач.

1.4 Постановка завдання

На основі проведеного аналізу предметної області, порівняльного аналізу існуючих CRM систем та сформованих вимог до працездатності платформи, що розробляється, визначено наступне:

- на сьогодні серед бізнес-структур неухильно зростає популярність CRM систем;
- лідируючі позиції серед малого бізнесу займає CRM-система «ZOHO»;
- основним недоліком CRM-системи «ZOHO» є відсутність простого і дешевого застосунку інтеграції користувачів та адміністратора із хмарним сервісом даної системи.

Дані висновки дозволили сформувати завдання і мету дослідження.

Завданням дослідження є організація зручного інтерфейсу між користувачами або адміністратором із хмарним сервісом CRM-системи «ZOHO»

Предметом дослідження є засоби інтеграції користувачів та адміністратора з CPM-системами.

Об’єктом дослідження є процес отримання, аналізу та зберігання інформації, сформованої CRM-системою «ZOHO».

Метою дослідження є розроблення програмної платформа для інтеграції із CRM-системою «ZOHO», зручної для користувачів та адміністратора і дешевою відносно аналогічних платформ.

Для досягнення мети необхідно вирішити такі часткові задачі:

- виконати проектування середовища функціонування, зокрема локального серверу, технології розробки, засобу реалізації бази даних та середовища програмування;
- виконати проектування системних взаємозв’язків, зокрема структурних елементів бази даних та розробити діаграму прецедентів;
- розробити відповідне програмне забезпечення;
- здійснити глибоке тестування розробленої платформи.

2 ПРОЕКТУВАННЯ СЕРЕДОВИЩА ФУНКЦІОНУВАННЯ

На даний момент в світі існує безліч різних технологій, для того щоб створити серверну частину веб-платформи, але в кожної є свої мінуси та плюси. Виділимо важливі критерії при виборі технологій:

- гнучкість рішення;
- наявність великої спільноти;
- відмово стійкість рішення;
- вимоги до навантажень;
- вимоги до безпеки;
- кросплатформеність;
- важкість проекту;
- швидкість розробки;
- вимоги до розширення проекту
- доступні інструменти розробки;
- наявність готових рішень.

2.1 Вибір локального серверу

Локальний сервер – програма, яка створює на персональному комп’ютері середовище повноцінного веб хостингу. Тобто на вашому домашньому комп’ютері створюється міні-хостинг, на якому будуть успішно функціонувати всі серверні движки, скрипти, CMS (WP, Joomla та інші). Вам навіть не потрібно буде підключатися до Інтернету – у вас буде свій міні-інтернет з одним або декількома сайтами. Так що за допомогою локального сервера можна успішно займатися веб-розробкою і потім переносити свої скрипти на реальний веб хостинг в інтернеті. Велика частина сайтів сучасного інтернету динамічні і працюють на технології ASP.NET. Але браузери розуміють тільки HTML і CSS, а C# який використовують для програмування

на ASP.NET - немає. Тому що C# це серверний мова програмування і сервер якраз перетворює і обробляє C#-код (або результат його виконання) в вид, зрозумілий браузеру. І такі обробники стоять на кожному сервері / хостингу в інтернеті (без них нікуди), але не на вашому домашньому комп'ютері [8].

Потреба в локальних серверах постійно росла при розробці динамічних сайтів на C#, Perl та іншими мовами програмування. Спочатку це було обумовлено поганим і дорогим Інтернетом, потім люди зрозуміли необхідність тестування скриптів в спеціальному середовищі, та й взагалі виріс аматорський і професійний інтерес до програмування.

Для повноцінної імітації веб-сервера і вирішення всіх вищеозначених завдань і був створений локальний сервер.

Зазвичай ці проблеми вирішувалися, та й до сих пір вирішуються, засобами FTP-клієнта. Файл завантажується з веб-хостингу, редагується та завантажується назад.

Це як мінімум незручно – треба бути постійно підключеним до стабільного Інтернету, потрібно чекати поки завантажиться назад (файл адже може бути великим, їх може бути кілька), потрібно постійно редагувати файли коли «щось йде не так». Локальний сервер, налаштований ідентично якби він був би на хостингу веб-сервера, спрощує цей процес. На веб-хостинг завантажується тільки підсумкова, фінальна версія веб-додатку.

Локальний сервер IIS. IIS (Internet Information Service) – програмне забезпечення для розгортання веб-сервера [9]. Розшифровується як «джентельменський набір веб-розробника» – набір дистрибутивів і програмного забеспечення для веб-розробки на локальному ПК. Денвер є одним з найстаріших локальних серверів широко відомих в рунеті, одним з основних переваг якого в момент появи була можливість роботи з USB-флеш-накопичувача.

Зараз вже є і інші локальні сервера, які не поступаються за функціоналом.

IIS – пропріетарний набір серверів для декількох служб Інтернету від компанії Microsoft. IIS поширюється з Windows NT. Основним компонентом IIS є веб-сервер, який дозволяє розміщувати в Інтернеті сайти. IIS підтримує протоколи HTTP, HTTPS, FTP, POP3, SMTP, NNTP.

Відмінність комплексу Winginx:

- швидкий і простий запуск локального сервера nginx на ОС Windows;
- зручна локальна розробка сайтів і сервісів на Node.js і PHP;
- мультипроектна система для розробки, що має універсальні і гнучкі налаштування, легко оновлюються компоненти;
- середовище для ведення проекту: можна створювати завдання і враховувати час на їх виконання;
- середовище для локального тестування і запуску веб-додатків, сайтів і браузерних сервісів.

Зовнішній вигляд серверу представлено на рисунку 2.1.

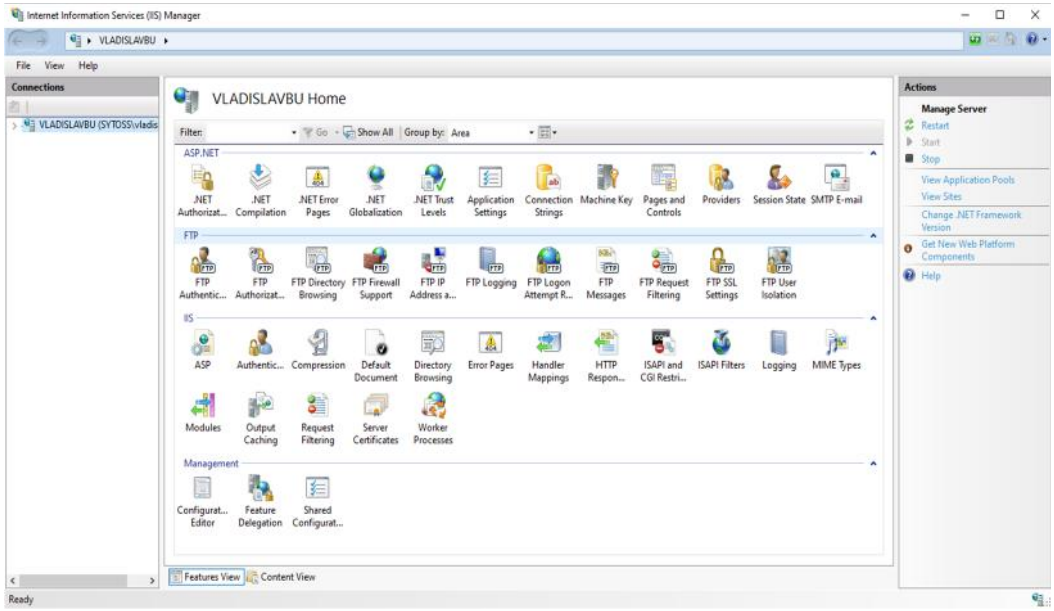

Рисунок 2.1 – Зовнішній вигляд IIS

Особливості Winginx в порівнянні з іншими локальними серверами: єдиний центр управління сервером і оновленнями компонентів, одночасна мультипроектного робота з декількома сайтами (в т.ч. використовуючи різні

версії PHP), управління завданнями і проектами, облік часу на виконання завдань, завантаження безкоштовних CMS з магазина Winginx і їх установка «в 1 клік». Winginx має вбудований серверний менеджер і центр поновлення. Winginx не навантажує локальний комп'ютер, непомітно працюючи в треї.

Локальний сервер Open Server – програмне середовище, що створює портативну локальну серверну платформу [11]. Open Server створений спеціально для веб-розробників і враховує всі отримані рекомендації і побажання по роботі середовища. Завдяки цьому, Open Server широко використовується в різних країнах світу для тестування, налагодження і розробки з нуля різних веб-проектів і створення повнофункціональних веб-серверів в локальних корпоративних і домашніх мережах. Зовнішній вигляд зображений на рисунку 2.2.

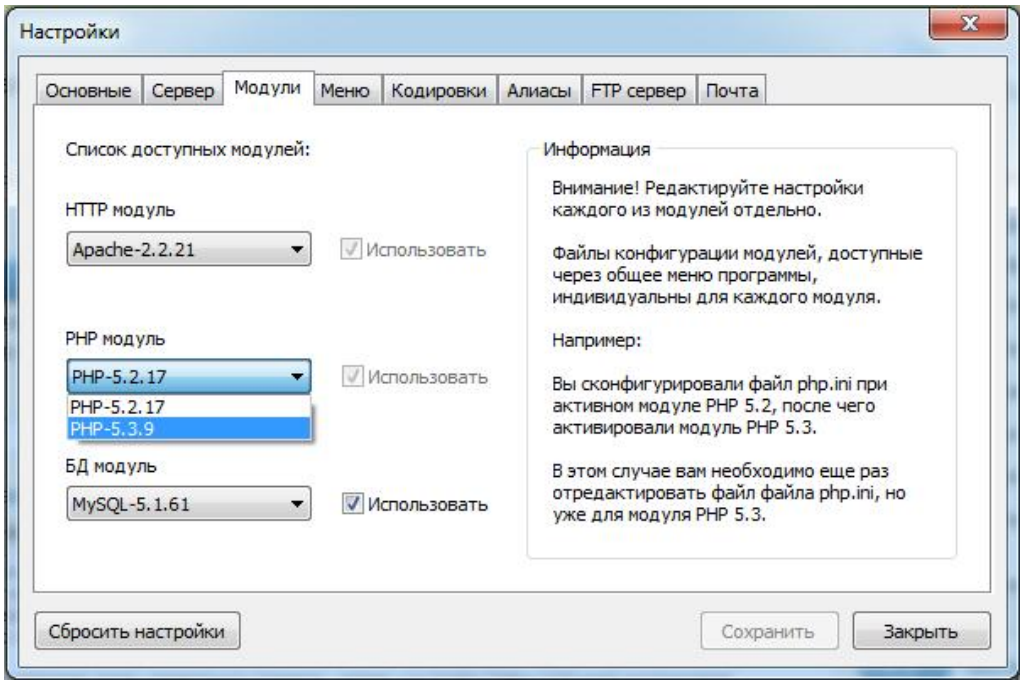

Рисунок 2.2 – Зовнішній вигляд OpenServer

Особливості комплексу OpenServer:

- не вимагає установки (портативність);
- можливість роботи з USB накопичувача;

- одночасна робота з Denwer, Xampp і т.д .;
- робота на локальному / мережевому / зовнішньому IP адресу;
- підтримка SSL без всяких додаткових налаштувань;
- створення домену шляхом створення звичайної папки;
- підтримка кириличних доменів;
- підтримка аліасів (доменних покажчиків);
- захист сервера від зовнішнього доступу;
- punycode конвертер доменних імен;
- пакет з понад 40 портативних програм;
- планувальник завдань (cron);
- створення локального піддомена без втрати видимості основного домену в мережі Інтернет;

В ролі локального серверу був вибраний саме OpenServer.

2.2 Вибір технології розробки

Сучасні технології створення і підтримки веб-сайтів орієнтовані на платформи, що дозволяють ефективно управляти інформаційним наповнюванням і даними, які надходять від відвідувачів сайту. Як правило, такі рішення базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP або використовують готові потужні засоби (наприклад, фреймворки) для створення корпоративних сайтів, орієнтованих на впровадження зазначених технологій.

Створення веб-сторінок фрагментами серверного коду є технологією ASP, ASP.NET (Active Server Pages). Це розроблення Microsoft комерційно доступна технологія, за допомогою якої веб-майстер може самостійно формувати динамічно поновлювані веб-сторінки. Характерною особливістю такої технології є можливість відділення функціональної частини розробки від процесів створення дизайну. ASP-сторінки можуть містити HTML-текст, змішаний зі сценаріями як JavaScript і VBScript. В процесі обробки запиту

нової сторінки його виконує сервер і динамічно генерує браузеру потік HTML-тексту відображається на екрані монітора. ASP-технологія Microsoft отримала подальший розвиток в технологіях JSP, PHP та ін.

Технологія JSP (Java Server Pages) – це технологія створення серверних сторінок Java. Специфікація JSP є розширенням Java Servlet API для генерації динамічних веб-сторінок на веб-сервері. Така крос-платформа є альтернативою технології ASP корпорації Microsoft.

Специфікація Sun під назвою JSF (Java Server Faces) реалізує технологію JSP, що описує правила створення веб-додатків зі зручним для користувача інтерфейсом і орієнтована на розробку серверних компонентів створення інтерфейсу.

Однією з перших технологій створення веб-додатків, які виконують-ються сервером, була Common Gateway Interface (CGI) технологія.

Вона дозволила розробку і виконання серверних додатків, обіг яких відбувається за допомогою зазначеного в URL імені (і параметрів). Залежно від обраного протоколу вхідної інформації таких веб-додатків вважають безпосередньо код HTTP-заголовка або запит пошукової системи. CGI-додатки – це консольні додатки, які генерують HTML-код, переданий браузера. Подібні додатки можуть являти собою код на скриптових мовах, який інтерпретується на сервері. Крім того, CGI- застосування представляють робочий файл, який можна створити за допомогою будь-якого засобу розробки, генерує консольні додатки для операційної системи, під управлінням якої працює веб-сервер.

Серед інших популярних технологій, що реалізують створення веб сторінок з фрагментами коду, що виконується на сервері, виділимо некомерційну, вільно поширювану технологію PHP (Personal HomePages). Ця технологія заснована на використанні CGI-додатків, інтерпретують впроваджений в HTML-сторінку код скриптовій мовою. Головною особливістю мови PHP є її практичність. PHP надає програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Вона

відрізняється винятковою гнучкістю до потреб розробника. хоча PHP традиційно рекомендують використовувати в поєднанні з HTML-кодом, проте PHP з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови програмування.

Результати порівняння технологій розроблення і впровадження веб-ресурсів зведені в таблиці 2.1. Розглянуті технології забезпечують одночасну функціональність, ефективний супровід процесів створення сайтів і їх наповнення інформаційними ресурсами.

Таблиця 2.1 – Порівняння сучасних технологій розробки сайту

	PHP	JSP	ASP.NET
Кросплатформеність	+	+	-
Продуктивність	+/-	+/-	+
Простота використання	+	+/-	+/-
Доступні програмні бібліотеки	+	+	+
Поділ дизайну і логіки	+	+/-	+
Безкоштовність	+	+	-

Отже, в даній кваліфікаційній роботі буде використовуватися мова програмування C#, однак робити веб-сервіс на чистому C#, дуже довго, и не зручно, тому треба вибрати сучасний фреймворк, який би ідеально підійшов для цього. Тому, порівняємо найпопулярніші C# фреймворки.

Фреймворк – це програмне забезпечення, що полегшує розробку і об’єднання різних компонентів великого програмного проекту. На відміну від бібліотеки функцій, фреймворк накладає обмеження на структуру і логіку програмного продукту [12].

Веб-фреймворк призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів і ресурсів. Веб-фреймворки містять логіку обробки HTTP-запитів, спрощують доступ до баз даних та інше. Фреймворк є

надбудовою над мовою програмування і дозволяє конструювати програми з сторонніх модулів, легко їх розширювати і модифікувати.

З одного боку, фреймворк вводить обмеження на структуру файлів, стиль оформлення коду, правила з розділення логіки і має функції, які можуть зовсім не використовуватися в готовому рішенні. З іншого боку, фреймворки скорочують час проектування і розробки додатків, виключають дублювання коду, а також спрощують супровід проектів.

Крім того, фреймворки супроводжуються документацією, навколо фреймворків утворюються співтовариства, що містять приклади і базу знань.

При виборі фреймворку для розробки необхідно враховувати патерни, які використовує фреймворк, наявність документації та розмір спільноти навколо фреймворка, а також можливість розширення функціоналу проектованої системи за рахунок власних і сторонніх розширень.

Під час вибору фреймворка був обраний ASP.NET Core. ASP.NET Core – це MVC-фреймворк. Цей вибір був зроблений виходячи з ряду причин відрізняючи його від попередніх версій ASP.NET, а саме:

- новий легкий і модульний конвеєр HTTP-запитів;
- можливість розгортати додаток як на IIS, так і в рамках свого власного процесу;
- використання платформи .NET Core і її функціональності;
- поширення пакетів платформи через NuGet;
- інтегрована підтримка для створення та використання пакетів NuGet;
- єдиний стек веб-розробки, що поєднує Web UI і Web API;
- конфігурація для спрощеного використання в хмарі;
- вбудована підтримка для впровадження залежностей;
- можливість розширення.

ASP.NET Core спочатку спроектований дуже ретельно для відповідності всім вимогам при розробці серйозних веб-додатків. ASP.NET Core не є ні побічним продуктом будь-якого проекту, ні складанням сторонніх рішень. Він є продуктом компанії Microsoft спільно з спільнотою і має велику

продуктивність в порівнянні з ASP.NET. Має модульну структуру і сумісна з такими операційними системами як Windows, Linux і macOS.

Незважаючи на те, що це новий фреймворк, побудований на новому веб-стеку, він має високий ступінь сумісності концепцій з ASP.NET. Додатки ASP.NET Core підтримують паралельне управління версіями, при якому різні програми, що працюють на одному комп'ютері, можуть орієнтуватися на різні версії ASP.NET Core. Це було неможливо в попередніх версіях ASP.NET. Статичну архітектуру фреймворку можна розглянути на рисунку 2.3.

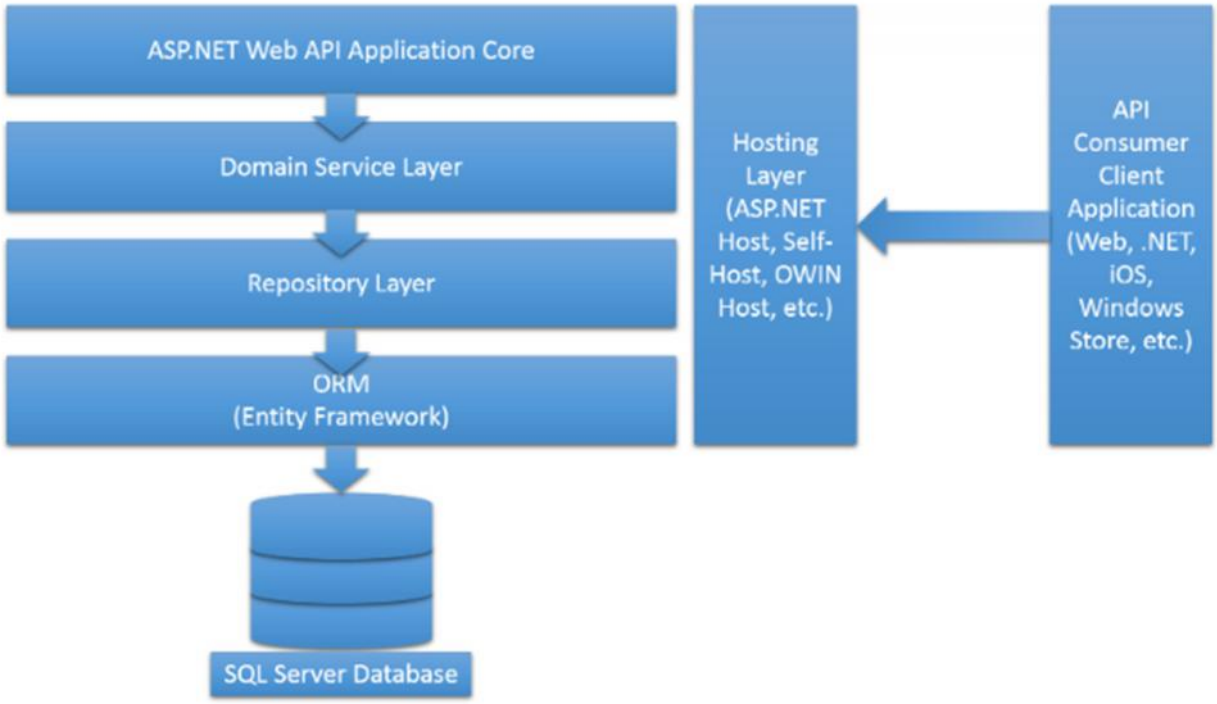

Рисунок 2.3 – Статична архітектура фреймворку

Особливості:

Висока продуктивність;

- інтерфейси DAO і ActiveRecord для роботи з базами даних (PDO);
- підтримка інтернаціоналізації;
- кешування сторінок та окремих фрагментів;
- перехоплення і обробка помилок;

- введення і валідація форм;
- аутентифікація і авторизація;
- використання AJAX і інтеграція з jQuery;
- генерація базового PHP-коду для CRUD-операцій (скаффолдінг);
- підтримка тем оформлення для їх легкої зміни;
- можливість підключення сторонніх бібліотек;
- міграції бази даних;
- автоматичне тестування;
- підтримка REST.

2.3 Вибір засобу реалізації бази даних

Робота з базами даних є однією з головних складових процесу програмування сайту динамічного типу. Бази даних для сайтів використовують з метою зберігання різнопланової інформації. База даних представляє з себе певний набір взаємопов’язаних таблиць. Розміри таблиць в базах різні, а їх кількість – довільна. Бази даних на сервері накопичують необхідну для роботи сайту інформацію статистичного характеру.

До найпопулярніших мережевих баз даних відносять SQL, MySQL, Oracle Database та інші. Вибір потрібної системи управління базою даних (СКБД) обумовлюється вимогами до інформаційних характеристик і функціональних можливостей системи.

Однією з найпоширеніших систем управління базами даних в наш час вважається MySQL, яка є альтернативою комерційним системам [13].

MySQL – вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона

використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування [16]. MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GP [14].

Об’єктно-реляційна система управління базами даних компанії Oracle (Oracle Database) орієнтована під операційні системи Windows, Unix, Linux і MacOS. Oracle Database, на відміну від MySQL, має більш широку сферу застосування. СКБД Oracle широко відома як в нашій країні, так і в світі. На ній базується безліч сучасних інформаційних систем [15].

SQL (Structured Query Language – мова структурованих запитів) – декларативна мова програмування засобів взаємодії користувача з базами даних, реалізує процеси формування запитів, оновлення і керування базами даних, створення схеми бази даних і її модифікації, систему контролю доступу до інформаційних ресурсів. SQL може формувати інтерактивні запити або, будучи вбудованою в додатки, виступати в якості інструкцій для керування даними. Крім того, стандарт SQL містить функції визначення процесів зміни, перевірки і захисту даних [16]. У таблиці 2.3 представлено порівняння сучасних СКБД.

Проаналізувавши базові характеристики розглянутих СКБД, можна зробити висновок про зручність використання MySQL для створення динамічних веб-сторінок, оскільки сервер MySQL по всім параметрам виявився найкращий, тому в даній каліфікаційній роботі будемо використовувати саме його.

Таблиця 2.3 – Порівняльна характеристика СКБД

	SQL	Oracle	MySQL
Надійність	+	+	+
Швидкість	–	+	+
Простота	–	–	+
Зручність користування	+/–	+	+
Безкоштовність	+/–	–	+

2.4 Вибір середовища програмування

Visual Studio 2019 Comunity – лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів (рисунок 2.4).

Дані продукти дозволяють розробляти як консольні додатки, так і ігри та програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight [17].

Інтегроване середовище розробки Visual Studio – це стартовий майданчик для написання, налагодження і складання коду, а також подальшої публікації додатків. Інтегроване середовище розробки (IDE) являє собою багатофункціональну програму, яку можна використовувати для різних аспектів розробки програмного забезпечення. Крім стандартного редактора і відладчика, які існують в більшості середовищ IDE, Visual Studio включає в себе компілятори, засоби автозавершения коду, графічні конструктори і багато інших функцій для спрощення процесу розробки.

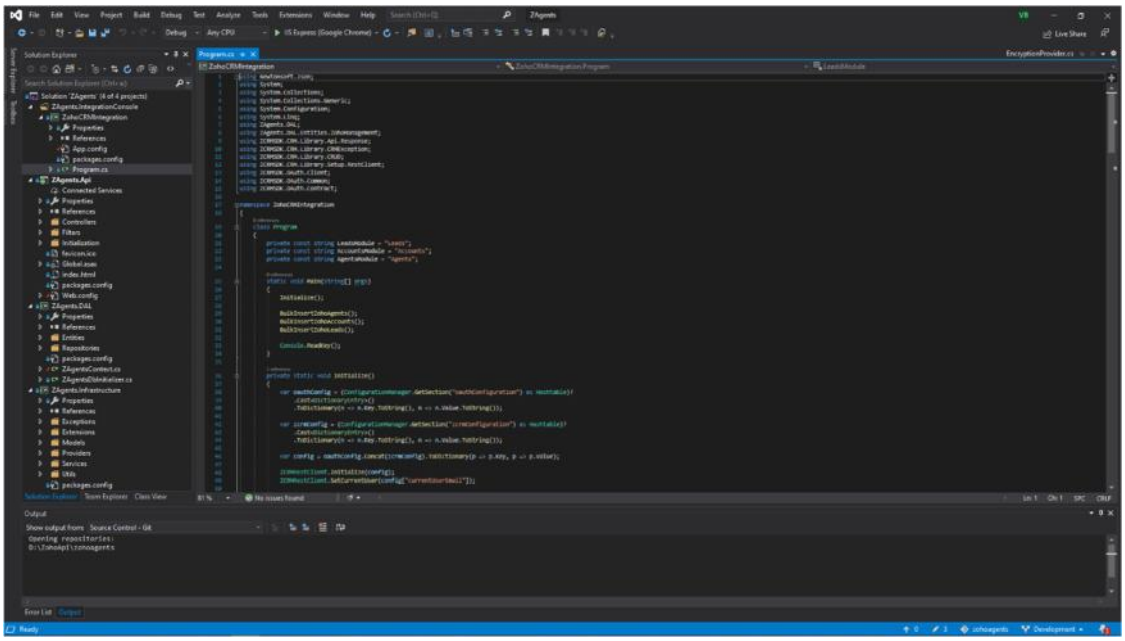

Рисунок 2.4 – Зовнішній вигляд Visual Studio 2019

Особливості. Основна задача IDE – спростити розробку програми з допомогою C# на різних платформах. Тому серед можливостей, які властиві будь-якому середовищу розробки, є і специфічні.

Редактор коду C#. Visual Studio надає багатий і інтелектуальний редактор коду для C# з підсвічуванням коду, розширеною конфігурацією форматування коду, перевіркою на наявність помилок на льоту і розумним авто доповненням. Можливості редактору коду:

- підтримка усіх версій C#, включаючи генератори, співпрограми, простори імен, замикання, типажі, синтаксис коротких масивів, доступ до члена класу при інстанціюванні, розіменування масиву при виклику функції, бінарні літерали, вираження в статичних виклики тощо. Visual Studio може використовуватися як для сучасних, так і для традиційних проектів на C#;

- автодоповнення коду фіналізують класи, методи, імена змінних, ключові слова C#, а також широко використовувані імена полів і змінних залежно від їхнього типу;

- простіше клонувати репозиторій Git або відкрити існуючий проект, також легше почати новий проект;

- отладчик моментальних знімків для налагодження додатків в хмарі Azure додає підтримку служби Azure Kubernetes і набору масштабів віртуальних машин;

- для налагодження точки зупину даних для додатків .Net Core допомагають розробникам ламатися тільки при тих змінах значень, які вони шукають;

- для налагодження в Visual Studio IntelliCode передбачено завершення коду за допомогою AI;

- розробники можуть отримати доступ до розміщених репозиторіїв з служб розробника Azure через вікно «Пуск»;

- розробники можуть встановлювати розширення для інших хостів контролю версій, щоб переглядати репозиторії, що належать розробнику або організації розробника;

- оновлення завантажуються у фоновому режимі, коли хост знаходиться в режимі очікування, після завершення завантаження користувачі отримують повідомлення про те, що завантаження готова до установки;

- покращення продуктивності в степпінге і перемиканні гілок.

- Visual Studio Live Share для спільної роботи встановлюється за умовчанням. Live Share можна використовувати для парного програмування, проведення оглядів коду, презентацій або «програмування мобів» під час хакатонів.

- покращено пошук для меню, команд, параметрів і встановлюваних компонентів.

- передбачений індикатор документа для розуміння «працездатності» файлу коду, який розробники можуть запускати і настроювати за допомогою очищення коду одним клацанням миші.

- розробники можуть зберегти набір виправлень для очищення коду у вигляді профілю для запуску під час очищення.

- .Net Core проекти можуть бути налаштовані простіше з першокласними файлами проекту.

- Google Chrome можна запускати з одними аргументами, а розробники можуть налагоджувати додатка JavaScript в середовищі IDE.

- підсвічування гарячих шляхів ідентифікує виклики функцій, які використовують найвищий відсоток ЦП або які виділяють найбільшу кількість об’єктів.

C # і Visual Basic підтримують аналізатор Regex. Регулярні вирази тепер розпізнаються і на них включені мовні функції, рядки Regex розпізнаються, коли рядок передається конструктору Regex або коли рядку відразу передує коментар, що містить рядок, Language = regex. в даний час включені наступні функції мови: класифікація, зіставлення дужок, посилання на основні моменти і діагностика. Розробники можуть переглядати функції мови C # 8.0, такі як:

- посилальні типи, що допускають обнулення;
- профілювання процесора пропонується для ASP.Net;
- доданий досвід роботи з одним проектом для контейнеризації та налагодження веб-додатків ASP.Net і консольних додатків для .Net Core.

Інструменти Visual Studio Kubernetes інтегровані в робоче навантаження розробки Azure. Завдяки поліпшеній підтримці робочих просторів Open Folder за допомогою нової панелі інструментів вибору середовища Python, стало простіше працювати з середовищами Python. Visual Studio 2019 видає підтримку Windows Mobile для універсальної платформи Windows (UWP). Розробники, яким необхідно продовжити роботу над додатком UWP для мобільних пристроїв Windows 10, повинні дотримуватися Visual Studio 2017 (Microsoft офіційно відмовилася від своєї мало використовуваної платформи Windows Mobile на початку 2019 року). Функція Search Deeper була змінена на список, що розкривається для швидкого вибору глибини початкового і подальшого пошуку. Для налаштування стилю коду можна застосувати з командного рядка за допомогою глобального інструменту формату dotnet або шаблон проекту VSIX для експериментів.

Для C ++ розробники можуть відкривати кеши CMake, згенеровані зовнішніми інструментами, такими як CMakeGUI або налаштовані системи

мета-збірки. Для C ++ поліпшений аналіз за допомогою / Qspectre, що надає допомогу щодо пом’якшення наслідків для уразливості Spectre Variant 1. Для C # продуктивність була покращена та для створення веб-додатків ASP.Net пропонуються візуальні поліпшення.Visual Studio 2019 автоматично завантажує оновлення у фоновому режимі, коли комп’ютер розробника знаходиться в режимі очікування, що дозволяє продовжувати використання до моменту установки, розробникам потрібно тільки почекати під час фактичної установки. Функція попереднього перегляду для кожного монітора включена за замовчуванням для користувачів, які відповідають системним вимогам .Net Framework 4.8 і Windows 10 April 2018 Update. Вікна інструментів, такі як Toolbox, Breakpoints і Call Stack, тепер повинні чітко відображатися на моніторах з різним масштабом і конфігураціями відображення, синя тема була оновлена за рахунок зменшення яскравості і контрастності. Функція справності документа була візуально оновлена, і розробники відразу побачили помилки або попередження в документі. Очищення коду має власний елемент управління для швидкого доступу до функцій та для розробки на C ++ розробники мають доступ до свіжої версії компілятора Microsoft Visual C ++ і набору інструментів для бібліотек (MSVC).

3 ПРОЕКТУВАННЯ СИСТЕМНИХ ВЗАЄМОЗВЯЗКІВ

3.1 Проектування бази даних

Концептуальне проектування. Мета етапу концептуального проектування – створення концептуальної моделі даних виходячи з уявлень користувачів про предметну область. Для її досягнення детально розглянемо ряд послідовних процедур, що виконується.

Визначення сутностей та їх документування. Для ідентифікації сутностей визначаються об’єкти, які існують незалежно від інших. Такі об’єкти є сутностями. Кожній сутності присвоюється осмислене ім’я зрозуміле користувачам. Імена та опису сутностей заносяться в словник даних. Якщо можливо, то встановлюється очікувана кількість примірників кожної сутності.

Визначення зв’язків між сутностями і їх документування. Визначаються тільки ті зв’язки між сутностями, які необхідні для задоволення вимог до проекту бази даних. Встановлюється тип кожної з них. Виявляється клас приналежності сутностей. Зв’язках присвоюються осмислені імена, виражені дієсловами. Розгорнутий опис кожної зв’язку із зазначенням її типу і класу приналежності сутностей, що беруть участь в зв’язку, заноситься в словник даних.

Створення ER-моделі предметної області. Для уявлення сутності і зв’язків між ними використовуються ER-діаграми. На їх основі створює-ся єдиний наочний образ моделюється предметної області – ER-модель предметної області.

Визначення атрибутів і їх документування. Виявляються всі атрибути, що описують сутність створеної ER-моделі. Кожному атрибуту при-привласнювати осмислене ім’я, зрозуміле користувачам. Про кожному атрибуті в словник даних збожеволіють такі відомості:

- ім’я атрибута і його опис;

- тип і розмірність значень;
- значення, яке приймається для атрибута за замовчуванням (якщо є);
- чи може атрибут мати Null-значення;
- чи є атрибут складовим, і якщо це так, то з яких простих атрибутів він складається;
- чи є атрибут розрахунковим, і якщо це так, то як обчислюються його значення.

Визначення значень атрибутів і їх документування. Для кожного атрибута сутності, що бере участь в ER-моделі, визначається набір допустимих значень і йому присвоюється ім’я.

Визначення первинних ключів для сутностей та їх документування. На цьому кроці керуються визначенням первинного ключа – як атрибуту або набору атрибутів сутності, що дозволяє унікальним чином ідентифікувати її екземпляри. Відомості про первинні ключи поміщаються в словник даних.

Обговорення концептуальної моделі даних з кінцевими користувачами. Концептуальна модель даних представляється ER-моделлю з супровідною документацією, що містить опис розробленої моделі даних. Якщо будуть виявлені невідповідності предметної області, то в модель вносяться зміни до тих пір, поки користувачі не підтвердять, що запропонована ним модель адекватно відображає їх особисті уявлення.

Логічне проектування БД. Логічне проектування баз даних - це процес створення моделі, що використовується в підприємстві інформації, на основі вибраної моделі організації даних, але без урахування типу цільової СКБД та інших фізичних аспектів реалізації. Мета полягає в створенні логічної моделі даних для досліджуваної частини підприємства. Логічна модель даних враховує особливості вибраної моделі організації даних у цільовій СКБД. Наприклад, визначення вимог підтримки цілісності даних та їх документування.

Логічна модель даних також грає важливу роль на етапі експлуатації та супроводу вже готової системи. При правильно організованому супроводі

підтримується в актуальному стані моделі даних, що дозволяє точно та наочно представляти будь-які внесені в базу даних зміни, а також оцінити їх вплив на прикладні програми та використання даних, вже наявних в базі.

Опис сутностей:

- сутність «users» містить інформацію про користувачів;
- сутність «invites» містить інформацію про користувачів яких додали до системи;
- сутність «zohoagents» містить інформацію яку нам надає CRM про агентів ZOHO CRM;
- сутність «zohoaccounts» містить інформацію яку нам надає CRM про акаунти ZOHO CRM;
- сутність «loginfos» містить інформації про користувачів які робили вхід до веб-платформи;
- сутність «roleusers» містить данні права користувачів та їх ролі.
- сутність «zoholeads» містить інформацію яку нам надає CRM про лідів ZOHO CRM;
- сутність «agentpermissions» містить права, які надані користувачу;
- сутність «baseagentpermissions» містить права, які є за замовчуванням;
- сутність «oauthtokens» містить дані про токени які надаються користувачу під час входу до платформи, та які потрібно оновлювати для того щоб проводити дії на платформі.

Фізичне проектування БД. Фізичне проектування баз даних – це процес підготовки описання реалізації баз даних на вторинних запам’ятовуючих пристроях.

На цьому етапі розглядаються основні відносини, організація файлів та індексів, призначених для забезпечення ефективного доступу до даних, а також всі пов’язані з цим обмеження цілісності та засоби захисту.

При виконанні етапу, розробник приймає рішення про способи реалізації розроблюваної бази даних.

Приступаючи до фізичного проектування баз даних, перш за все необхідно вибрати конкретну цільву СКБД.

Тому фізичне проектування нерозривно пов’язане з конкретною СКБД. Між логічним та фізичним проектуванням існує постійний зворотній зв’язок, так як рішення, прийняті на етапі фізичного проектування з метою підвищення продуктивності системи, здатні впливати на структуру логічної моделі даних.

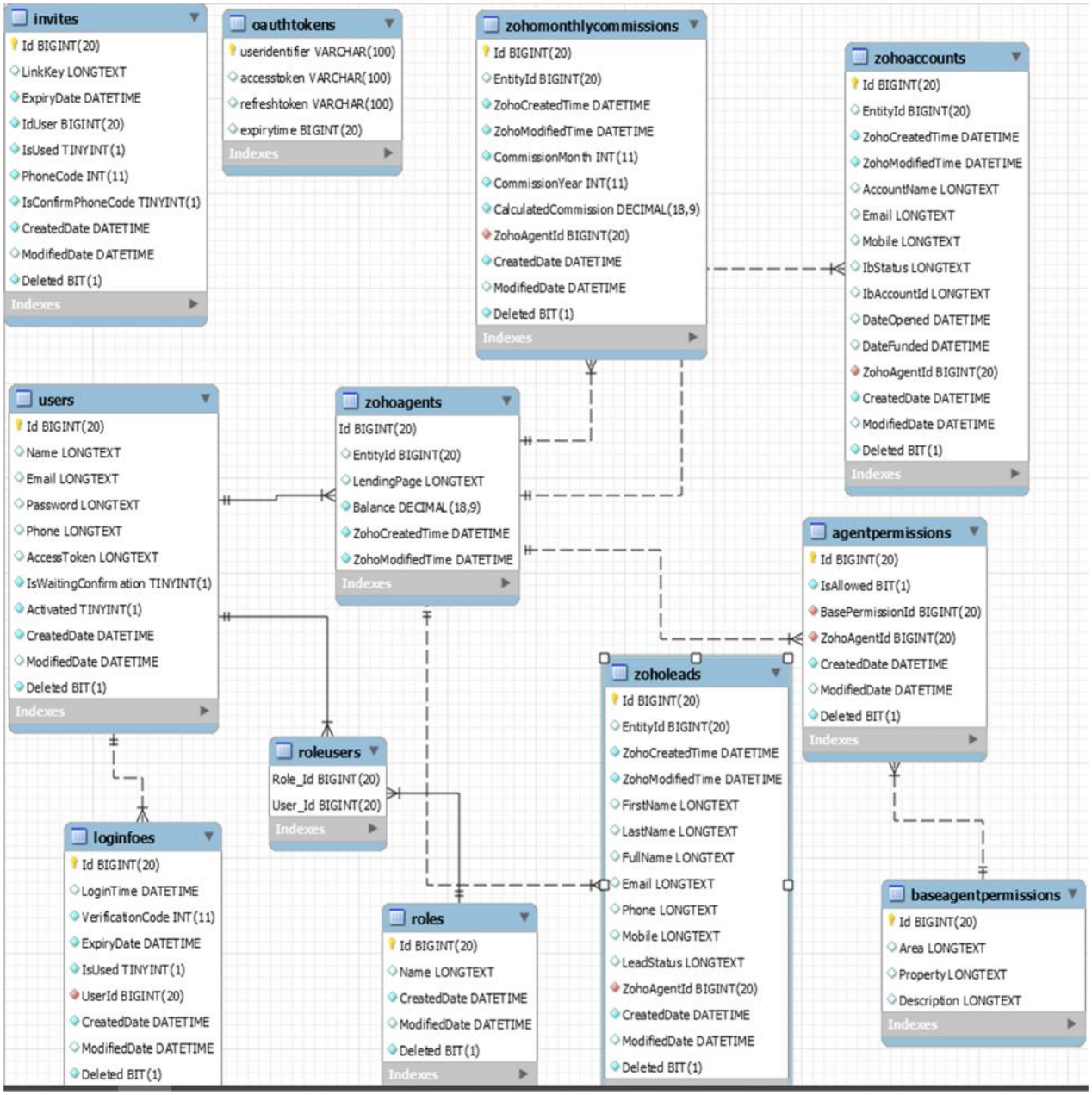

Рисунок 3.1 – Фізична модель бази даних

Як правило, основним завданням фізичного проектування баз даних є опис способу фізичної реалізації логічного проекту бази даних. На рисунку 3.1 представлена фізична модель бази даних.

Результатом фізичного проектування логічної схеми вище на мові SQL являються приклади скриптів, наведені на лістингах 3.1 та 3.2.

Цілісність бази даних. Під цілістю бази даних розуміється узгодженість (несуперечність) даних. Звичайно, СКБД не може контролювати правильність кожного окремого значення, введеного в базу даних. Цілісність баз даних може бути порушена внаслідок збою обладнання, помилок користувача або програмної помилки. В системах з багатьма користувачами цілість може бути порушена при одночасному зверненні до одного й того самого фрагмента даних.

Цілісність забезпечується шляхом задання обмежень. В залежності від джерела можна виділити інструментальні обмеження, структурні обмеження та бізнес-правила.

До інструментальних обмежень відносять перевірку правильності даних при введенні.

Наприклад, поле числа не може містити текстові символи, а поле дати має містити допустиме значення дати. Середні рівні реалізації інструментальних обмежень вбудовані в СКБД.

До структурних обмеженням відносять унікальність первинного ключа та унікальність можливих ключових слів. В відношенні будь-яких двох кортежах не можуть мати одне і те ж значення атрибута (або агрегату атрибутів), об’явленої як первинний або можливий ключ. Крім того, в первинному або в можливому ключі не може бути компонент з нульовим знанням. Система управління повинна відхиляти будь-яку спробу (при введенні або оновленні) вставляти в базу даних кортеж, ключ якого або пустий (нуль), або має значення, вже введене в базу даних.

Бізнес-правила представляють собою умови, що дані відповідають-предметній області. Бізнес-правила можна розділити на елементарні та

розширені. Елементарні правила обмежують значення конкретного атрибута або агрегату атрибутів через обмеження. Розширені правила виражаються у вигляді деякої залежності між атрибутами.

Лістинг 3.1 – Скрипт створення таблиці «users»

```
CREATE TABLE `users` (
  `Id` bigint(20) NOT NULL AUTO_INCREMENT,
  `Name` longtext,
  `Email` longtext,
  `Password` longtext,
  `Phone` longtext,
  `AccessToken` longtext,
  `IsWaitingConfirmation` tinyint(1) NOT NULL,
  `Activated` tinyint(1) NOT NULL,
  `CreatedDate` datetime NOT NULL,
  `ModifiedDate` datetime DEFAULT NULL,
  `Deleted` bit(1) NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=InnoDB AUTO_INCREMENT=370 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci,
```

Лістинг 3.2 – Скрипт створення таблиці «zoholeads»»

```
CREATE TABLE `zoholeads` (
  `Id` bigint(20) NOT NULL AUTO_INCREMENT,
  `EntityId` bigint(20) DEFAULT NULL,
  `ZohoCreatedTime` datetime NOT NULL,
  `ZohoModifiedTime` datetime NOT NULL,
  `FirstName` longtext,
  `LastName` longtext,
  `FullName` longtext,
  `Email` longtext,
  `Phone` longtext,
  `Mobile` longtext,
  `LeadStatus` longtext,
  `ZohoAgentId` bigint(20) NOT NULL,
  `CreatedDate` datetime NOT NULL,
  `ModifiedDate` datetime DEFAULT NULL,
  `Deleted` bit(1) NOT NULL,
  PRIMARY KEY (`Id`),
  KEY `IX_ZohoAgentId` (`ZohoAgentId`),
  CONSTRAINT `FK_zoholeads_zohoagents_ZohoAgentId` FOREIGN KEY
(`ZohoAgentId`) REFERENCES `zohoagents` (`Id`)
) ENGINE=InnoDB AUTO_INCREMENT=58237 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

При проектуванні бази даних були введені такі бізнес-правила:

- в одного користувача може буде додано декілька електронних адрес і телефонів для сповіщень (при цьому є одна основна пошта, яка використовується для входу на сайт);
- тарифний план можна придбати лише на 30 днів і більше;
- користувач може додати декілька доменів для моніторингу (при цьому другий користувач має можливість також додати цей домен);
- події можуть бути двох типів (0 і 1, де 0 це подія яка вказує, що сталась якась помилка на сайті, і 1 це якщо сайт знову перейшов в робочий стан.

3.2 Діаграма прецедентів

Прецеденти (варіанти використання) – це технологія визначення функціональних вимог до системи. Робота прецедентів полягає в описі типових взаємодій між користувачами системи і самою системою і надання опису процесу її функціонування. Ключем до прецедентів є мета користувача: прецедент являє собою безліч сценаріїв, об’єднаних деякою загальною метою користувача.

Сценарій (scenario) – це послідовність кроків, що описують взаємовідносини користувача і системи.

У термінах прецеденту користувачі називаються акторами. Актор (actor) являє собою якусь роль, яку користувач грає по відношенні до системи. Актори діють в рамках прецедентів. Один актор може виконувати кілька прецедентів; і навпаки, відповідно до одним прецедентом можуть діяти кілька акторів.

Не існує стандартного способу опису вмісту прецеденту, в різних випадках застосовуються різні формати. Загальний стиль використання описаний нижче.

Вибір одного з сценаріїв в якості головного успішного сценарію (main success scenario). Спочатку потрібно описати тіло прецеденту, в якому

головний успішний сценарій представлений послідовністю нумерованих кроків. Потім потрібно вибрати інший сценарій і вставити його в вигляді розширення (extension), описуючи його в термінах змін головного успішного сценарію. Розширення можуть бути успішними - користувач досяг своєї мети, або невдалими.

У кожному прецеденті є провідний актор, який посилає системі запит на обслуговування. Провідний актор – це актор, бажання якого намагається задовольнити прецедент і який зазвичай, але не завжди, є ініціатором прецеденту. Одночасно можуть бути і інші актори, з якими система також взаємодіє під час виконання прецеденту. Вони називаються другорядними акторами.

Кожен крок у прецеденті – це елемент взаємодії актора з системою. Кожен крок повинен бути простим твердженням і повинен чітко вказувати, хто виконує цей крок. Крок повинен показувати намір актора, а не механіку його дій. Отже, в прецеденті інтерфейс актора не описується.

Прецеденти є цінний інструмент для розуміння функціональних вимог до системи. Перший варіант прецедентів повинен складатися на ранній стадії виконання проекту. Більш докладні версії прецедентів повинні з’являтися безпосередньо перед реалізацією даного прецеденту. Важливо розуміти, що прецеденти представляють погляд на систему з боку. Тому, відповідності між прецедентами і класами всередині системи може і не бути.

Незважаючи на те що в мові UML нічого не говориться про текст прецедентів, саме текстовий зміст прецедентів є основною цінністю цієї технології. Діаграма варіантів використання веб-додатка представлена на рисунку 3.2.

В результаті аналізу поставленого завдання була виявлена роль користувача і роль адміністратора – які являються фізичною особою та працюють з даним веб додатком.

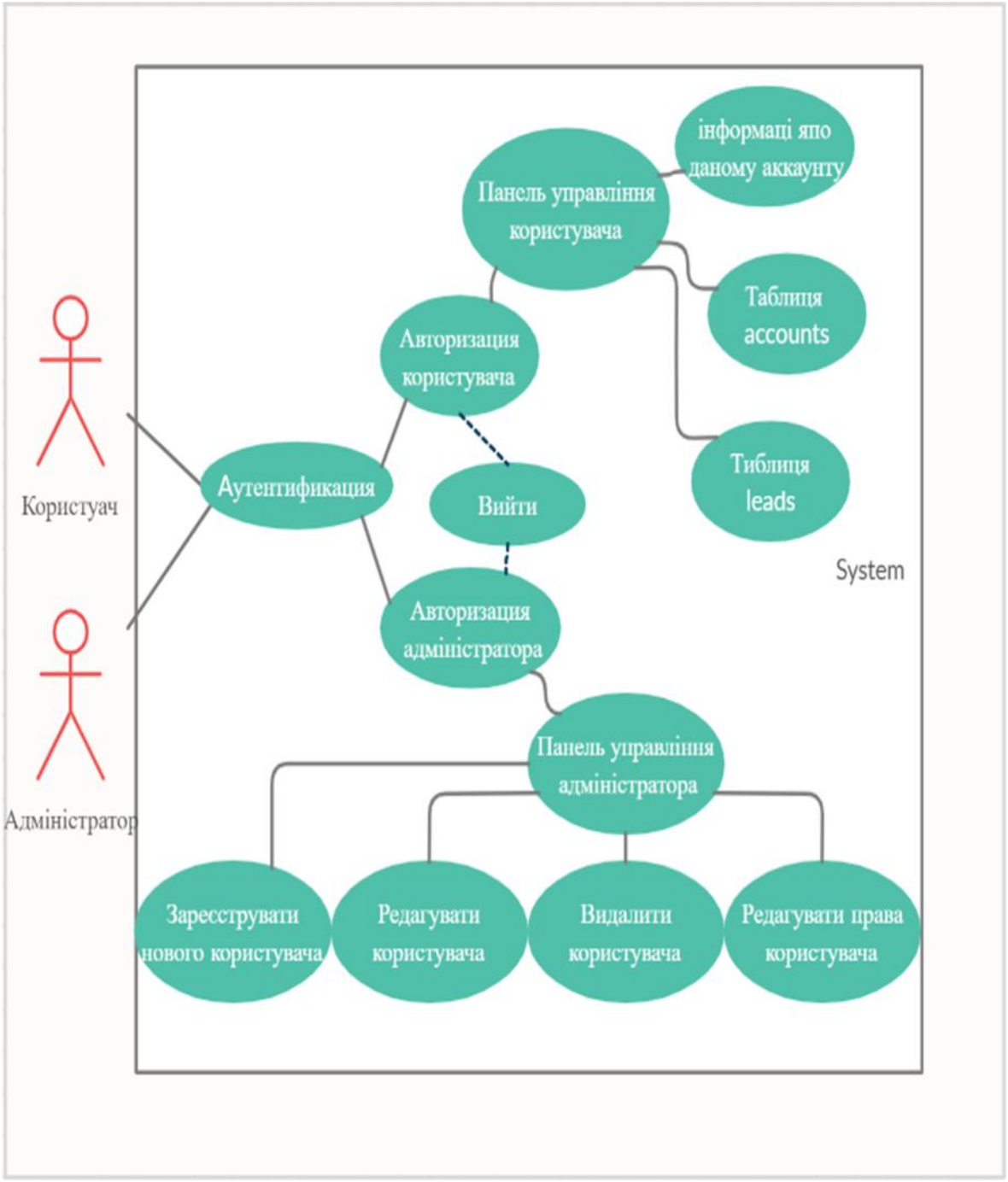

Рисунок 3.2 – Діаграма варіантів використання

Найкраще деталізувати діаграму прецедентів за допомогою графічної таблиці (специфікації), яка б показала їх вміст (таблиця 3.1 – 3.11). Деталізація прецеденту – це точне визначення кожного прецеденту.

Таблиця 3.1 – Специфікація прецеденту «Автентифікація»

Ім’я прецеденту	Автентифікація
Ідентифікатор прецеденту	ID 1
Короткий опис	Процес розпізнавання користувача системи і надання йому певних прав та повноважень
Дійсні актори	Користувач, адміністратор
Передумова	Користувач, адміністратор повинен бути зареєстрованим в системі або бути зареєстрованими іншими адміністраторами
Вихідні умови	У разі успішної автентифікації виконується вхід користувача в систему, в іншому випадку виводиться помилка автентифікації.
Порядок дій	1) Натиснути кнопку «Log in» 2) Заповнити всі обов’язкові поля (e-mail і пароль) 3) Натиснути кнопку підтвердження

Таблиця 3.2 – Специфікація прецеденту «Авторизація користувача»

Ім’я прецеденту	Авторизація користувача
Ідентифікатор прецеденту	ID 2
Короткий опис	привласнення прав користувачу на вчинення дій в системі які йому дозволені
Дійсні актори	Користувач
Передумова	Користувач повинен успішно пройти процедуру автентифікації
Вихідні умови	Попадання в систему керування
Порядок дій	-

Таблиця 3.3 – Специфікація прецеденту «Авторизація Адміністратора»

Ім’я прецеденту	Авторизація адміністратора
Ідентифікатор прецеденту	ID 3
Короткий опис	привласнення прав адміністратора на вчинення будь-яких дій в системі
Дійсні актори	Адміністратор
Передумова	Адміністратор повинен успішно пройти процедуру автентифікації
Вихідні умови	Попадання в систему керування адміністратора

Таблиця 3.4 – Специфікація прецеденту «Вийти»

Ім’я прецеденту	Вийти
Ідентифікатор прецеденту	ID 4
Короткий опис	Вихід с системи
Дійсні актори	Користувач, адміністратор
Передумова	Авторизований користувач або авторизований адміністратор
Вихідні умови	Вихід з веб додатку, завершення сесії
Порядок дій	Натиснути кнопку «Log out»

Таблиця 3.5 – Специфікація прецеденту «Панель управління користувача»

Ім’я прецеденту	Панель управління користувача
Ідентифікатор прецеденту	ID 5
Короткий опис	Панель, де можливе управління сайтами
Дійсні актори	Користувач
Передумова	Авторизований користувач
Вхідні умови (надані CRM)	1. Отримати інформацію про акаунт. 2. Отримати інформацію про його лідах. 3. Отримати інформацію про його акаунти.

Таблиця 3.6 – Специфікація прецеденту «Панель управління адміністратора»

Ім’я прецеденту	Панель управління адміністратора
Ідентифікатор прецеденту	ID 6
Короткий опис	Панель управління користувачами та їх правами.
Дійсні актори	Адміністратор
Передумова	Авторизований адміністратор
Вихідні умови	Панель, де адміністратор може управляти користувачами, реєструвати та запрошувати нових користувачів. Визначати які поля інформації будуть видні користувачам.
Порядок дій	–

Таблиця 3.7 – Специфікація прецеденту «Додавання нового адміністратора»

Ім’я прецеденту	Додавання нового адміністратора
Ідентифікатор прецеденту	ID 7
Короткий опис	Додавання нового адміністратора
Дійсні актори	Користувач, адміністратор
Передумова	Авторизація користувача і вхід в панель управління на сторінку управління сайтами
Вихідні умови	Можливість додавати, видаляти адміністратора із списку адміністраторської панелі та системи
Порядок дій	1. Натиснути кнопку «Три крапка поряд з обраним адміністратором» 2. Вибір дії 3. Зберегти

Таблиця 3.8 – Специфікація прецеденту «Надання прав користувачу»

Ім’я прецеденту	Надання прав користувачу
Ідентифікатор прецеденту	ID 8
Короткий опис	Надання прав користувачу
Дійсні актори	Адміністратор
Передумова	Авторизація адміністратора і вхід в панель управління на сторінку з користувачами
Вихідні умови	Можливість обрати користувача та надати йому доступ до тої інформації, яку бажає адміністратор, щоб бачив користувач.
Порядок дій	Обрати користувача, та натиснути «Три крапки» поряд з користувачем.

Таблиця 3.9 – Специфікація прецеденту «Пошук користувача»

Ім’я прецеденту	Пошук користувача
Ідентифікатор прецеденту	ID 9
Короткий опис	Пошук користувача
Дійсні актори	Користувач, адміністратор
Передумова	Авторизація користувача і вхід в панель управління на сторінку перегляду статистики
Вихідні умови	Можливість зробити пошук за email, телефоном або ім’ям.
Порядок дій	Натиснути поле «Search»

Таблиця 3.10 – Специфікація прецеденту «Змінити пароль»

Ім’я прецеденту	Змінити пароль
Ідентифікатор прецеденту	ID 10
Короткий опис	Змінити пароль
Дійсні актори	Користувач, адміністратор

Продовження табл. 3.10

Передумова	Авторизація користувача і вхід в панель управління на сторінку покупок
Вихідні умови	Можливість змінити свій пароль від аккаунту.
Порядок дій	Натисниту на свій профіль та натиснути кнопку «Change password»

Таблиця 3.11 – Специфікація прецеденту «Редагування профілю»

Ім’я прецеденту	Редагування профілю
Ідентифікатор прецеденту	ID 11
Короткий опис	Редагування профілю користувача
Дійсні актори	Користувач, адміністратор
Передумова	Авторизація користувача і вхід в панель управління на сторінку редагування даних
Вихідні умови	Зміна персональних даних
Порядок дій	1. Змінити всі необхідні дані 2. Натиснути кнопку «Edit»

Отже, проведена деталізація прецедентів за допомогою таблиць специфікацій дозволила сформувати функціональні вимоги до системи, що розробляється, формалізувати типові взаємодії між користувачами системи і самою системою, між адміністратором системи і системою, а також надати опис процесу її функціонування. Це дозволило сформувати платформу для інтеграції з CRM-системою «ZOHO» та перейти до її дослідження, зокрема до процесу верифікації розробленої платформи.

4 ТЕСТУВАННЯ

Процес верифікації вимог є невід’ємною частиною всього процесу розробки. Верифікація тісно пов’язана з проектуванням, розробкою і супроводом сайту.

Верифікація – це процес переконання, що завдання було виконано в повній відповідності з вимогами, які викладені в технічному задані. Паралельно з цим також фіксується нові дефекти. Верифікація дозволяє гарантувати, що програмна система реалізована без непередбаченої функціональності, відповідає пред’явленим вимогам, специфікаціям і стандартам.

Верифікація перевіряє відповідність одних створюваних в ході розробки і супроводу ПЗ артефактів іншим, раніше створеним або використовуваним в якості вихідних даних, а також відповідність цих артефактів і процесів їх розробки правил і стандартів [18]. Зокрема, верифікація перевіряє відповідність між нормами стандартів, описом вимог (технічного завдання) до ПЗ, проектними рішеннями, вихідним кодом, призначеної для користувача документації та функціонуванням самого ПЗ. Крім того, перевіряється, що вимоги, проектні рішення, документація і код оформлені відповідно до норм і стандартів, прийнятих в даній країні, галузі та організації при розробці ПЗ, а також що при їх створенні виконувалися всі зазначені в стандартах операції, в потрібній послідовності. Виявлені при верифікації помилки і дефекти є розбіжностями або протиріччями між декількома з перерахованих документів, між документами і реальною роботою програми, між нормами стандартів і реальним процесами розробки і супроводу ПЗ. При цьому прийняття рішення про те, який саме документ підлягає виправленню (може бути, і обидва) є окремим завданням [19].

Валідація перевіряє відповідність будь-яких створюваних або використовуваних під час розробки і супроводу ПЗ артефактів потреб і потреб

користувачів і замовників цього ПЗ, з урахуванням законів предметної області і обмежень контексту використання ПЗ. Ці потреби і потреби частіше за все не зафіксовані документально – при фіксації вони перетворюються в опис вимог, один з артефактів процесу розробки ПЗ [20]. Тому валідація є менш формалізованої діяльністю, ніж верифікація. Вона завжди проводиться за участю представників замовників, користувачів, бізнес-аналітиків або експертів в предметній області – тих, чия думка можна вважати бвиразом реальних потреб і потреб користувачів, замовників та інших зацікавлених осіб. Методи її виконання часто використовують специфічні техніки виявлення знань і дійсних потреб учасників.

Тестування – це процес, який полягає в перевірці відповідності програмного продукту або сайту заявленим характеристикам і вимогам.

Далі докладно описані етапи тестування сайту.

Перший етап починається все з підготовчих робіт – тестувальник вивчає отриману документацію (аналізує функціонал за технічним завданням, вивчає кінцеві макети сайту і становить план тесту для подальшого тестування).

Функціональне тестування (другий етап) – найбільш тривалий етап перевірки ресурсу. Суть цього процесу полягає в перевірці всього описаного функціоналу:

- перевірки роботи всіх обов’язкових функцій сайту;
- тестування працездатності призначених для користувача форм;
- перевірки роботи пошуку (включаючи релевантність результатів);
- перевірки гіперпосилань, пошук неробочих посилань;
- перевірки завантаження файлів на сервер;
- перевірки працездатності лічильників на сторінках сайту;
- перегляд на відповідність вмісту сторінок сайту вихідного контенту, наданим замовником.

Тестування верстки (третій етап) – при перевірці верстки насамперед тестувальник перевіряє розташування елементів, відповідність їх позицій надавати пріоритети макетів, а також перевіряє оптимізацію зображень і

графіки. Так здійснюється перевірка валідності коду. У процесі верстки важливо зберегти коректну ієрархію об’єктів, і важливо упевнитися в її валідності за фактом завершення робіт.

Usability тестування (четвертий етап) – проводиться для оцінки зручності продукту у використанні, заснований на залученні користувачів в якості тестувальників і аналіз отриманих результатів.

Незважаючи на той факт, що опрацювання зручності використання ресурсу здійснюється в процесі складання технічного завдання, розробки макетів, бувають ситуації, коли отриманий результат не є оптимальним ним. Хоча таке і відбувається досить рідко, оптимальне рішення в даному випадку – виконати зміни в реалізований товар.

Тестування безпеки (п’ятий етап) – на даній стадії тестування спеціаліст перевіряє – чи немає у користувачів доступу до службових / закритих сторінок, а також проводить перевірку захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу.

Тестування продуктивності сайту (шостий етап) – проводиться з метою визначення швидкодії сайту або його частини під певним навантаженням. Тестування продуктивності включає в себе тестування навантаження та швидкодії.

Тестування навантаження – найпростіша форма тестування продуктивності. Тестування навантаження зазвичай проводиться для того, щоб оцінити поведінку сайту (або додатки) під заданої очікуваної навантаженням. Цією навантаженням може бути, наприклад, очікувана кількість одночасно працюючих користувачів на сайті, що здійснюють вказану кількість транзакцій за інтервал часу. Такий тип тестування зазвичай дозволяє отримати час відгуку всіх найважливіших бізнес-функцій;

Тестування швидкодії – перевірка швидкості завантаження сайту для визначення швидкості відпрацювання скриптів, завантаження зображень і контенту. Цей тест проводиться з метою оптимізації процесу завантаження сайту, а також визначення оптимальності налаштувань сервера.

В даному розділі будемо використовувати функціональне тестування веб-додатку.

Завданням функціонального тестування є перевірка відповідності програми своїм специфікаціям. При даному підході текст програми не доступний, і програма розглядається як "чорної скриньки". Найпоширенішими видами функціонального тестування є методи випадкового тестування, еквівалентної розбивки й аналізу граничних умов.

Функціональне тестування призначене для перевірки відповідності програмного забезпечення його специфікації (завданню) або еталону, тобто перевірки, чи виконує програмна система або її модулі відповідні функції та поставлені вимоги. При цьому тестувальник перевіряє виконувані функції, а не реалізацію програмного забезпечення шляхом аналізу вхідних і вихідних даних. У науковій літературі цей метод називають ще методом тестування "чорної скриньки".

Для реалізації цього методу повинні бути відомі функції програм програмного забезпечення. Результати функціонального тестування дають відповіді на запитання: як виконуються функції програм програмного забезпечення; як сприймаються вхідні дані. Функціональне тестування застосовується тільки коли програмне забезпечення вже створене, тобто на останніх етапах життєвого циклу програмного забезпечення. Якщо функціональне тестування виявить погану якість створення програмного забезпечення, то доводиться повертатися на попередні етапи розробки, що спричиняє фінансові збитки і часові витрати.Тестування в перспективі «вимоги» використовує специфікацію функціональних вимог до системи як основу для дизайну тестових випадків (Test Cases). У цьому випадку необхідно зробити список того, що буде тестуватися, а що ні, пріоритезувати вимоги на основі ризиків (якщо це не зроблено в документі з вимогами), а на основі цього пріоритезувати тестові сценарії (test cases). Це дозволить сфокусуватися і не упустити при тестуванні найбільш важливий функціонал. Приклади тестових випадків представлені в таблицях 4.1 – 4.5.

Таблиця 4.1 – Тестовий випадок №1

Ідентифікатор тестового випадку		TC-2 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити можливість виходу із системи	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту (рис. 4.1)	+
2	Користувач входить в систему	Користувач вводить email та пароль	+
3	Користувач натискає на кнопку «Log in»	Користувач ввійшов з системи	+
Результати тесту: тест виконаний успішно			

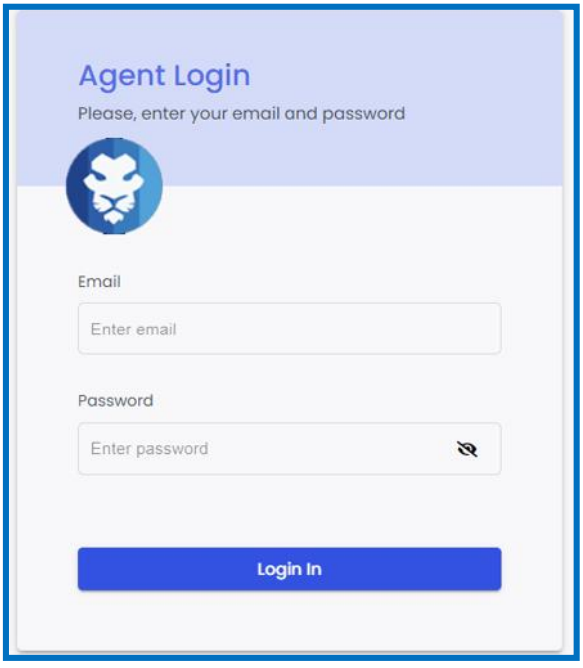

Рисунок 4.1 – Головна сторінка сайту (для користувача)

Таблиця 4.2 – Тестовий випадок №2

Ідентифікатор тестового випадку		TC-2 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити можливість виходу із системи	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту (рис. 4.1)	+
2	Користувач входить в систему	Користувач увійшов в систему (рис. 4.2)	+
3	Користувач натискає на кнопку «Log out»	Користувач вийшов з системи	+
Результати тесту: тест виконаний успішно			

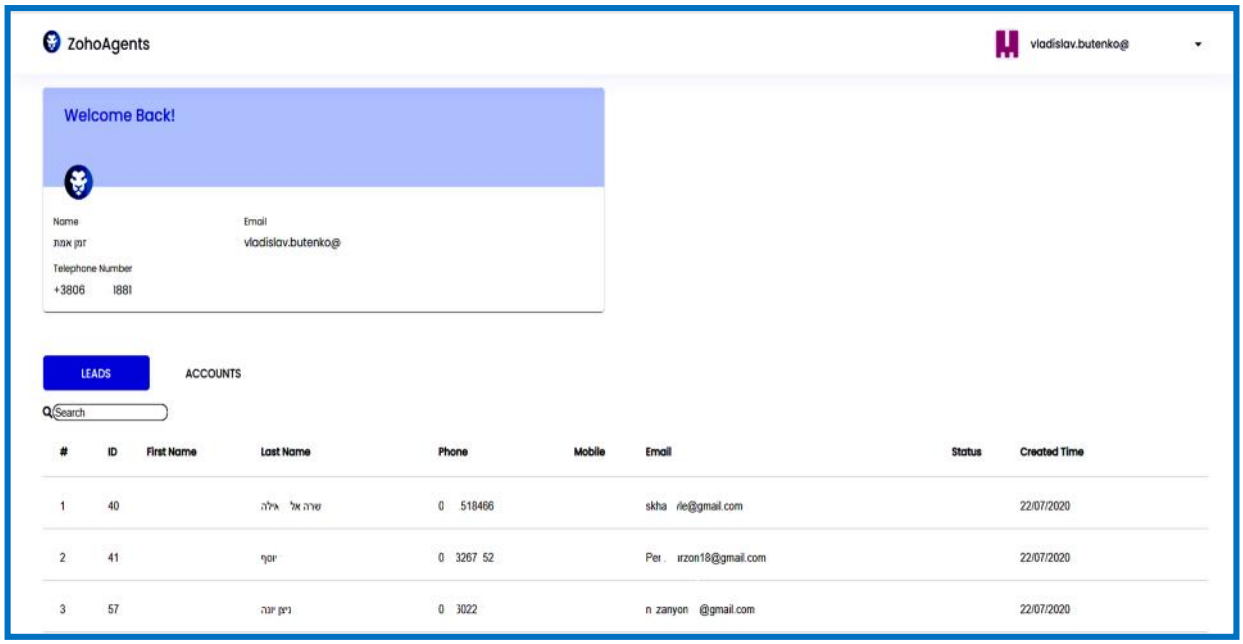

Рисунок 4.2 – Особистий кабінет користувача

Таблиця 4.3 – Тестовий випадок №3

Ідентифікатор тестового випадку		TC-3 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити можливість додати сайт в систему	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Користувач входить в систему	Користувач увійшов в систему	+
3	Користувач натискає на кнопку «Create new admin»	З’являється pop-up для додавання нового адміністратора.	+
4	Користувач вписує домен сайту у спеціальне поле для вводу і натискає кнопку «Додати»	Якщо дані введені, то список оновлюється и в ньому нові дані, додавання адміністратора (Рисунок 4.3) , якщо це не коректний e-mail або телефон, то буде помилка, що дані введені невірні (рис. 4.4)	+
Результати тесту: тест виконаний успішно			

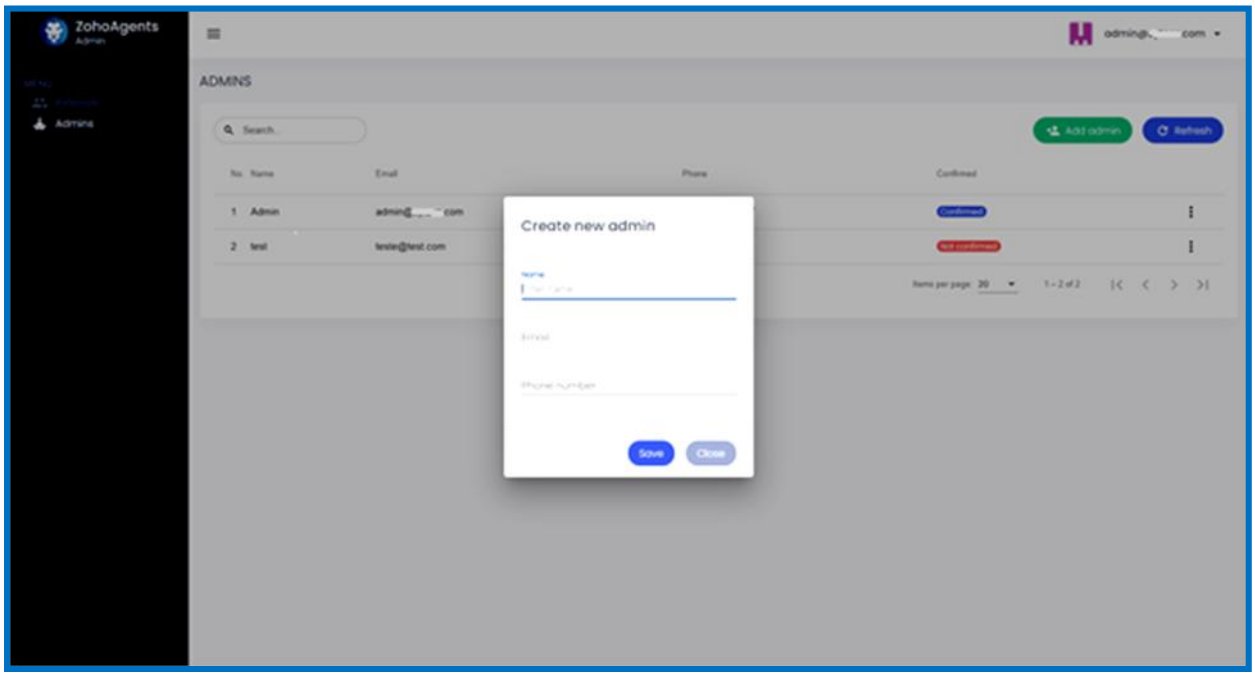

Рисунок 4.3 – Вікно для додавання нового адміністратора.

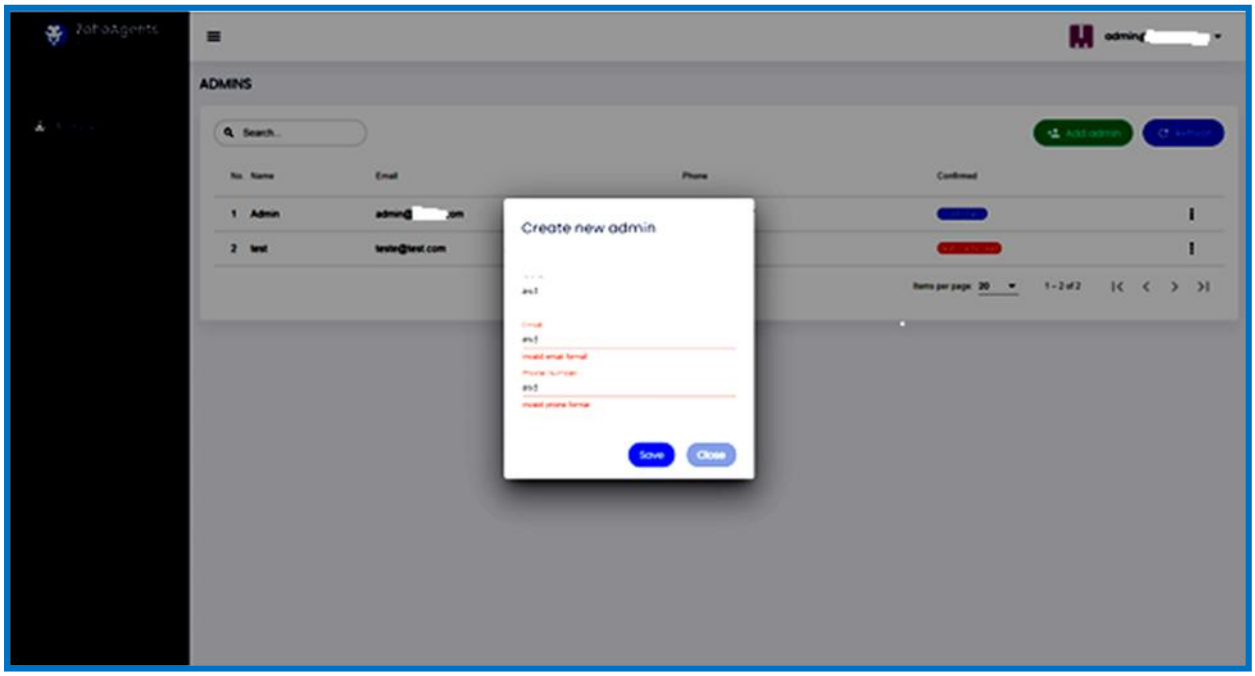

Рисунок 4.4 – Помилка додавання адміністратора

Таблиця 4.4 – Тестовий випадок №4

Ідентифікатор тестового випадку		TC-4 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити які сайти додані користувачем до системи моніторингу	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Користувач входить в систему	Користувач увійшов в систему	+
3	Користувач натискає вкладку «Мої сайти»	Потрапляє на сторінку сайтів які були додані раніше до системи (рис. 4.5)	+

Рисунок 4.5 – Реферали, яких можна зареєструвати та додати до платформи

Таблиця 4.5 – Тестовий випадок №5

Ідентифікатор тестового випадку		TC-5 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити налаштування для сайту та збереження цих налаштувань в базі даних	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Адміністратор входить в систему	Адміністратор увійшов в систему	+
3	Користувач натискає вкладку «Мої сайти»	Потрапляє на сторінку Рефералов які були запитані у CRM.	+
4	Вибирає який сайт треба налаштувати	Потрапляє на сторінку налаштування користувачів (рис. 4.6)	+
5	Оновлює необхідні дані та натискає кнопку «Зберегти»	Отримує повідомлення про успішне зберігання налаштувань	+

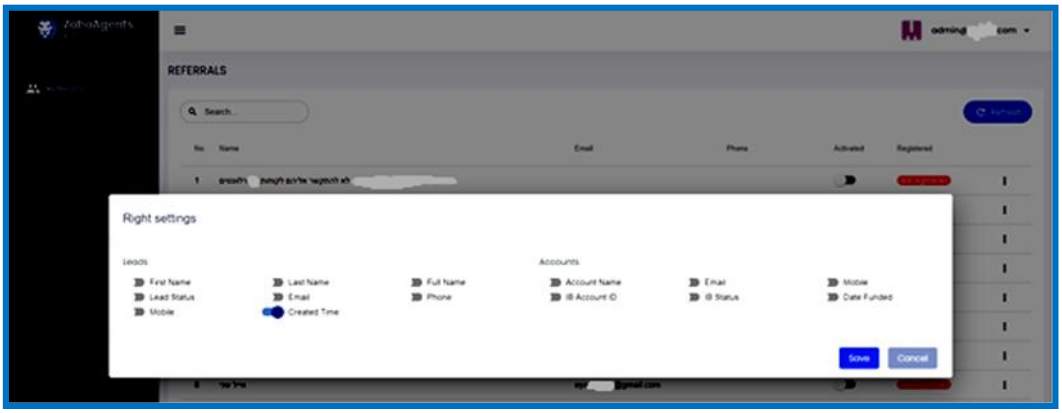

Рисунок 4.6 – Сторінка налаштувань прав користувача

Маючи вимоги до кількості відкритих багів, після кожної ітерації тестування ми порівнюємо їх з реальними даними, тим самим бачачи місця, де нам потрібно додати, для якнайшвидшого досягнення мети.

ВИСНОВКИ

За результатами виконання поставлених задач та проведеної науково-дослідної роботи сформульовано такі висновки:

1. Проведено дослідження існуючих CRM, порівняння існуючих CRM з метою виявлення недоліків і переваг кожної з них.

2. Проведено аналіз мов програмування. На основі аналізу було зроблено вибір на користь мови програмування C# як основної мови розробки, технології Swing як засіб створення графічного інтерфейсу та СУБД MySQL як засобу зберігання інформації.

3. У проектному розділі були виконані розробку бази даних, з описом усіх таблиць і їх призначення, розроблені інтерфейс, спроектовано серверну і клієнтську частину веб-сервісу.

4. Створена платформа та компоненти системи відповідають заявленим функціональним і сучасним технічним вимогам і готові до широкого використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Web Application Performance: 7 Common Problems and How to Solve Them – Режим доступу : https://stackify.com/web-application-problems/

2. Ian Molyneaux The Art of Application Performance Testing, 2nd Edition / Ian Molyneaux – O'Reilly Media, Inc. 2019.

3. Performance Testing : What is, Types, Metrics & Example – Режим доступу : https://www.guru99.com/performance-testing.html

4. Host-tracker Database Site. URL: https://www.host-tracker.com/ua

5. Official Monitis Database Site. URL: https://www.monitis.com/

6. Official Site24x7Database Site. URL: https://www.site24x7.com/

7. Official Uptimerobot Database Site. URL: https://uptimerobot.com/

8. https://blogwork.ru/chto-takoe-lokalnyj-server/

9. Official Denwer Database Site. URL: http://www.denwer.ru/

10. Official Winginx Database Site. URL: https://winginx.com/ru/

11. Platform development for integration with the "ZOHO" CRM system N. Kuchuk, R.Kravchenko Проблеми інформатизації: Матеріали дев'ятої міжнародної науково-технічної конференції. –Черкаси: ЧДТУ; Баку: ВА ЗС АР; Харків : НТУ «ХПІ»; Київ : НАУ; Харків : ДП «ПДПРОНДІАВІАПРОМ», 2021. – Т.1., С. 103.

12. Гапак О.М. Захист інформації в комп'ютерних системах: навчально-методичний посібник для студентів напряму підготовки «комп'ютерна інженерія». – Ужгород: Видавництво ПП «АУТДОР-ШАРК», 2015. – 172 с.

13. Лаврищева Е.М., Грищенко В.Н. / Складальне програмування. Основи промисловості програмних товарів / 2-вид. Доповнене та перероблене. - Київ: Наук. думка, 2019.– 372с.–ISBN 978-966-00-0848-1.

14. Нікітін С. О. Основи комп'ютерних ігор та ігрових програм : [довідник модуля] / С. О. Нікітін, Л. О. Нікітіна. – Харків : Друкарня Мадрид,

2018. – 138 с. – Режим доступу: http://repository.kpi.kharkov.ua/handle/KhPI-Press/37402.

15. Official MySQL Database Site. URL: https://www.mysql.com/

16. Official Oracle Database Site. URL: https://www.oracle.com /index.html

17. Грофф Дж., Вайнберг П. SQL: повне керівництво. К: BHV, 2019. – 608 с.

18. Official Jetbrains Database Site. URL: https://www.jetbrains.com/ru-ru/phpstorm/

19. Web Application Performance: 7 Common Problems and How to Solve Them – Режим доступу : https://stackify.com/web-application-problems/

20. IEEE 1012-2004 Standard for Software Verification and Validation. IEEE, 2020.

21. ISO/IEC 12207 Systems and software engineering - Software life cycle processes. Geneva, Switzerland: ISO, 2018.