

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного
забезпечення

Кваліфікаційна робота бакалавра

на тему «CRM система обліку заявок клієнтів інтернет-провайдера»

Виконав: студент групи _____ ПЗ19-1

Спеціальність _____ 121 Інженерія програмного
забезпечення

_____ Зубко Владислав Дмитрович
(прізвище та ініціали)

Керівник _____ к.т.н., доц. Фірсов О. Д.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент _____ УМСФ
(місце роботи)

_____ в.о. зав.каф. транспортних технологій
та міжнародної логістики
(посада)

_____ к.т.н., доц. Кузьменко А.І.
(науковий ступінь, вчене звання, прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка: 99 с., 22 рисунка, 4 таблиць, 39 джерел.

Об'єкт: бізнес-процес взаємодії технічної підтримки інтернет-провайдера та абонентів.

Предмет: автоматизація бізнес-процесу по обслуговуванню абонентів інтернет-провайдеру.

Мета проекту: розробка CRM системи обліку заявок клієнтів інтернет-провайдера.

Основна задача дипломного проекту, знайти спосіб у який буде автоматизована комунікація між відділом технічної підтримки, за допомогою програмно-інформаційних засобів. Результатом автоматизації є:

- зниження витрат робочого часу оператора підтримки на обробку заявки;
- зниження витрат робочого часу техніка, що займається усуненням помилок у абонента;
- мінімізація помилок в обробці заявок викликаной «людським фактором».

У першому розділі наведено відомості про об'єкт дослідження, обґрунтовано доцільність, викрито ряд проблем, потреба в вирішенні яких обумовлює пошук програмного рішення для автоматизації об'єкта дослідження. Також на основі знайдених проблем у об'єкті дослідження, формується постановка задачі, яку буде вирішено у наступних розділах.

У другому розділі проведено огляд існуючих рішень на проблеми з першого розділу та специфікації вимог. Проведено аналіз доцільності використання існуючих рішень виходячи з проблем з першого розділу, технічних вимог, вимог до безпеки особистих даних абонентів (згідно GDPR [1]), та можливості подальшого вдосконалення системи, під розглянутий бізнес-процес, враховуючи його можливі зміни, ускладнення, збільшення бази абонентів, збільшення кількості операторів, можливе додавання нових акторів у процесі, тощо.

У третьому розділі проведено практичну реалізацію розробленої CRM системи, та надана документація для користувачів, та технічні відомості для

розробників.

Актуальність розробки CRM системи полягає у зменшенні людської похибки, зменшення часу оператора та техніка по обробці заявки абонента, а як наслідок можливе скорочення робочого штату та зменшення фонду оплати праці підприємства. Також автоматизація робить адміністрування заявок стандартизовано, пришвидшує пошук потрібної заявки, дає змогу вести облік, аналітику, та згодом, завдяки цифровізації даних та подальшої розробки системи може виводити графіки ефективності операторів чи техніків, вести облік їх KPI.

Список ключових слів: АВТОМАТИЗАЦІЯ, ВЕБ-ДОДАТОК, БАЗА ДАНИХ, CRM, MYSQL.

ABSTRACT

Explanatory note: 99 pages, 22 figures, 4 tables, 39 references.

Object: Business process of interaction between technical support of an internet service provider and subscribers.

Subject: Automation of the customer service business process for the internet service provider.

The main objective of the diploma project is to develop a CRM system for tracking customer requests for the internet service provider.

The main task of the diploma project is to find a way to automate the communication between the technical support department using software tools.

The result of automation includes:

Reducing the working time spent by support operators on request processing;

Reducing the working time spent by technicians on troubleshooting for subscribers.

Minimizing errors in request processing caused by human factors.

The first chapter provides information about the research object, justifies the relevance, highlights a number of problems, and emphasizes the need to find a software solution for automating the research object. Based on the identified problems within the research object, the chapter formulates the task that will be addressed in the subsequent chapters.

The second chapter provides an overview of existing solutions to the problems identified in the first chapter and specifications of requirements. It analyzes the feasibility of using existing solutions based on the problems identified in the first chapter, technical requirements, requirements for personal data security (according to GDPR [1]), and the potential for further system improvement considering the discussed business process, taking into account possible changes, complexities, an increase in the subscriber base, an increase in the number of operators, and the possible addition of new actors to the process, etc.

The third chapter focuses on the practical implementation of the developed CRM

system and provides documentation for users and technical information for developers.

The relevance of developing a CRM system lies in reducing human error, reducing the time spent by operators and technicians on processing subscriber requests, and consequently, the possibility of reducing the workforce and labor costs of the enterprise. Automation also standardizes request administration, speeds up the search for necessary requests, allows for tracking, analysis, and, eventually, through data digitalization and further system development, enables the generation of efficiency charts for operators or technicians and tracking their KPIs.

Keywords: AUTOMATION, WEB APPLICATION, DATABASE, CRM, MYSQL.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1 Опис завдання	13
1.2 Опис предметної області і напрямків дослідження.....	13
1.3 Аналіз і характеристика об'єкта проектування	15
1.4 Висновки до першого розділу	19
РОЗДІЛ 2. ОГЛЯД ІСНУЮЧИХ РОЗВ'ЯЗКІВ ПОСТАВЛЕНОЇ ВИЩЕ ЗАДАЧІ, ЇХ АНАЛІЗ ТА ВИБІР МЕТОДІВ ВИРІШЕННЯ	20
2.1 Огляд та аналіз існуючих аналогів	20
2.2 Обґрунтування оптимального варіанта реалізації мети кваліфікаційної роботи	31
2.2.1 Висновки аналізу існуючих аналогів	31
2.2.2 Вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу	32
2.2.3 Вибір операційної системи	32
2.2.4 Вибір мови програмування	33
2.2.5 Вибір веб-серверу	35
2.2.6 Вибір бази даних	36
2.3 Висновки до другого розділу	39
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ	40
3.1 Опис алгоритму і програмного забезпечення	40
3.2 Вибір і обґрунтування структури проектування системи та її компонентів	42
3.3 Опис використовуваного системного програмного забезпечення для розробки і експлуатації	45
3.4 Основні рішення з реалізації системи в цілому і її компонентів	46

3.5 Інструкція роботи користувача з системою	51
3.5.1 Інструкція роботи техніка КРС	51
3.5.2 Інструкція роботи оператора відділу підтримки	53
3.6 Висновки до третього розділу	56
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТОК А	65
ДОДАТОК Б	69

Перелік умовних позначень

ПЗ – програмне забезпечення

КРС – кабельно-ремонтна служба інтернет-провайдеру

CRM (Customer relationship management) – для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

ERP-система (Планування ресурсів підприємства) призначена для автоматизації обліку й керування. Зазвичай ERP-системи будуються за модульним принципом і в тому або іншому ступені охоплюють всі ключові процеси діяльності компанії.

SLA (Service level agreement) або рівень сервісу – ключовий показник, який дає зрозуміти, чи добре з нашим клієнтським сервісом. Іншими словами, швидко і комфортно наші клієнти отримують відповідь оператора або доводиться не можна пробачити довго чекати відповіді.

OTRS (Open-source Ticket Request System) – відкрита систему обробки заявок. OTRS дозволяє організаціям, які займаються технічною підтримкою будь-яких проектів, спільно працювати над вирішенням проблем користувачів.

ITS (Issue tracking system) – Система відстеження проблем (також ITS, система заявок на усунення несправностей, служба підтримки, керування запитамми або система заявок на інциденти). Програма як послуга (англ. Software as a service, SaaS) – модель поширення програм споживачам, при якій постачальник розробляє веб програму, розміщує її й керує нею (самостійно або через третіх осіб) з метою використання її замовниками через інтернет.^{[1][2]} Замовники платять не за володіння програмами як такими, а за їх використання (через API, що доступне через веб і яке часто використовують веб служби). SaaS також відомий як «програмне забезпечення на вимогу».

WMS – Система управління складом

HRMS – Система управління персоналом

БД – база даних.

СКБД – система керування базами даних.

Ключові показники ефективності (key performance indicators, KPI) – фінансова та нефінансова система оцінки, яка допомагає організації визначити ступінь досягнення стратегічних цілей працівником.

XSS (Cross Site Scripting, міжсайтовий скриптинг) – тип вразливості інтерактивних інформаційних систем у вебi. XSS виникає, коли на сторінки, які були згенеровані сервером, з якоїсь причини потрапляють користувацькі скрипти. Специфіка подібних атак полягає в тому, що замість безпосередньої атаки сервера зловмисники використовують вразливий сервер для атаки на користувача.

ORM (Object-relational mapping, Об'єктно-реляційна проекція) – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних».

MVC (Модель–вид–контролер, Model-view-controller) – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

LNMP – аббревіатура набору вільного ПЗ з відкритим кодом, в який входять ОС Linux, вебсервер Nginx, СКБД MySQL та інтерпретатор PHP.

ВСТУП

У сучасному світі вже існує велика кількість рішень, що виконує функції зв'язків з клієнтами, поліпшенню комунікації з ними, ведення статистики тощо. На перший погляд здається що всі можливі бізнес-процеси давно автоматизовані, і якщо поглянути на існуючі рішення можна швидко знайти оптимальну систему, яка закриває всі потреби підприємства. Є великі системи схожі на швейцарський ніж, є системи для малих задач. Але при більш близькому розгляді не всі вони оптимально реалізують бізнес-процес конкретного підприємства. Це не дуже дивно, бо майже всі з сучасних систем автоматизації бізнесу працюють по системи оплати SaaS, що передбачає залучення максимальної кількості підприємств, що користуються системами для рентабельності самої системи автоматизації. Цей факт робить розробників системи заручником найрозповсюдженіших бізнес-процесів. Наразі маємо таку ситуацію. Розробникам великих систем автоматизації треба робити свої системи так, щоб ними користувалися максимальна кількість підприємств. Тому вони часто не мають змогу пропрацювати всі можливі варіанти розгалуження бізнес процесів, і реалізують типові бізнес процеси, наприклад продаж товарів без умов залучення декількох фахівців, чи системи для продажу інформаційних умов за підпискою, без детального налаштувань складових тарифів. Звичайно є виключення, наприклад Sales Force, Microsoft Dynamics CRM, SAP та багато інших. Вони мають іншу проблему – складність користування, та впровадження. Велика кількість налаштувань, дуже ускладнює роботу, та дає привід персоналу саботувати перехід на ці системи.

Також важливою складовою вибору системи автоматизації є безпека клієнтських даних. Багато підприємств обирають розробку свого ПЗ замість користування вже готової, а розглядаючи нашу предметну область це набуває ще і економічного сенсу.

Економіка інтернет-провайдера складається з таких доходів і розходів.

Доходи це абонентські платежі клієнтів щомісяця, а до витратної частини

ВХОДЯТЬ:

- оренда каналу зв'язку;
- витрати матеріалу на підключення абоненту;
- витрати на ремонт та модернізацію мережі;
- фонд оплати праці для робітників компанії.

Тобто ми можемо зробити висновок, що значну частину витрат складають зарплати працівників, зокрема, операторів техпідтримки, та техніків, тому задача зі зменшення кількості працівників веде до збільшення прибутку, а придбання сторонньої системи автоматизації може нівелювати вигоди від використання системи автоматизації. Також у будь-якої з систем автоматизації що є на ринку відсутня гнучкість, та можливість допрацювати необхідний функціонал, що може бути потрібен при зміненню оптимізованого бізнес процесу.

Розробка CRM система обліку заявок клієнтів інтернет-провайдера дозволить вирішити ці протиріччя найбільш гнучким та оптимальним способом.

Метою цієї кваліфікаційної роботи є розробка інформаційної системи обліку заявок клієнтів інтернет-провайдера, з простим і зрозумілим інтерфейсом, котра буде мінімізувати можливість помилок у роботі персоналу, та пришвидшить обробку заявок.

Актуальність розробки полягає у необхідності підприємства інформатизувати існуючі дані та можливості здійснювати швидку обробку заявок, пошук, облік ефективності роботи робітників інтернет-провайдера.

Очікується, що розроблений веб-додаток буде мати такі характеристики, як інтуїтивно зрозумілий інтерфейс, можливість швидкого пошуку інформації, розподілений доступ для різних працівників, та можливість модернізації.

У процесі роботи було проведено огляд та аналіз існуючих рішень та технологій на яких можна розробляти цю CRM, що дало змогу обґрунтовано обрати стек технологій, що буде найбільш доцільний до вирішення даної задачі. Було розроблено інтерфейс додатку, звичний для користувачів за допомогою css бібліотеки twitter bootstrap з використанням об'єктно-орієнтованої мови програмування Javascript. Бекенд виконано на фреймворку laravel з використанням

у якості сховища даних СКБД MySQL.

Ця кваліфікаційна робота детально описує процес розробки Web-додатку «CRM система обліку заявок клієнтів інтернет провайдера», а також надає рекомендації з його використання, різними типами користувачів, та технічний опис структури веб-додатку, для спрощення модернізації кодової бази.

Результати роботи сприятимуть поліпшенню та пришвидшенню доступу до актуальної інформації для робітників провайдера.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис завдання

Завдання полягає у створенні зручного інтерфейсу для обліку заявок клієнтів інтернет-провайдера, де працівник провайдера зможе бачити яку саме заявку слід обробити, помітити заявку як оброблену, чи створити нову заявку. Звідси витікає такі вимоги до системи:

- мобільність – доступність з будь-якої точки;
- можливість роботи з будь-яких пристроїв(комп'ютер, ноутбук, планшет, телефон);
- захист від витоку особистих даних клієнтів;
- актуальність статусів заявок;
- можливість одночасної роботи багатьох працівників з однією базою заявок;
- можливість розширення функціоналу, наприклад відстежування показників роботи відділів.

1.2 Опис предметної області і напрямків дослідження

У компанії працює 2 вида працівників, що обробляють заявки, це оператори відділу технічної підтримки, та технічні спеціалісти (монтажники).

У звернення в компанію є 2 випадка, це діючий клієнт, або який бажає стати їм, та планує підключитись. При відповідному випадку ми обробляємо запит клієнта по різному.

Якщо це нове підключення, то перевіряємо технічну можливість підключення за даною адресою. Якщо технічна можливість є, тоді перевіряємо чи

була адреса раніше зареєстрована адреса, чи ні. І вже у відповідності результату надаємо умови клієнту. Якщо клієнт згоден з умовами, тоді формується заявка на підключення, по якій у порядку черги займаються відповідні спеціалісти(техніки, монтажники), використовуючи одну систему.

У зверненнях від діючих клієнтів можуть питання різного характеру. Питання технічного або фінансового характеру. Якщо питання пов'язано с приводу балансу або тарифу чи додаткових опцій, ми переглядаємо карточку клієнта і можемо побачити рахунок клієнта, його особистий рахунок та тарифний план. Коли питання клієнта стосується технічної сторони підключення, не працює передача зв'язку або якась інша складність, то спершу оператор намагається вирішити цю проблему по телефону, проконсультувавши абонента про можливі несправності безпосередньо на комп'ютері абонента, або у приміщенні(відійшов кабель, вимкнули роутер, тощо). Якщо проблему не вийшло вирішити у телефонному режимі то оператор оформляє заявку на виклик КРС до абонента для усунення проблеми. Фахівці кабельно-ремонтної служби перевіряють обладнання по будинку, обладнання у абонента, налаштування його мережевої карти, тощо.

У центрі бізнес-процесу у нас є заявка, яка має статус і може його змінювати. Послідовна зміна статусу заявкою і є моделлю нашого бізнес-процесу, який ми будемо інформатизувати. Кожен статус заявки відповідає діям які вона потребує зробити собою. Життєвий цикл заявки обмежений статусами які нам треба відслідковувати. Типова заявка проходить через такі статуси:

- створена;
- потребує дзвінка;
- потребує візиту майстра КРС;
- оброблена

Статуси «створена» і «оброблена» присутні завжди, а інші статуси залежать від змісту звернення абонента, та працівника який за неї відповідає і обробляє її.

1.3 Аналіз і характеристика об'єкта проектування

Вивчивши предметну область ми можемо сказати що у нас в системі повинно бути 2 ролі:

- оператор технічної підтримки;
- працівник відділу КРС.

Кожна з ролей повинна мати свій логін і пароль, і свій інтерфейс, де буде бачити лише потрібну інформацію для роботи, і не бачити зайвих даних.

Кожна з ролей повинна мати змогу заявку, але по різному. Роль КРС повинна мати змогу лише змінити статус і додати коментар до заявки, а роль оператора технічної підтримки повинна мати змогу створити заявку, змінити статус на будь який.

У таблиці 1.3.1 розглянемо можливості кожної з ролей за паттерном CRUD (Create Read Update Delete).

Таблиця 1.3.1 – Можливості кожної з ролей за паттерном CRUD (Create Read Update Delete)

операція	оператор	КРС
створити	+	-
читати	+	+
змінити	+	+
видалити	-	-

Такий аналіз нам дещо дав, але нам треба більш детально визначити операцію зміни, бо вона дуже обширна, тому що включає в себе як зміну статусу і коментаря к заявці, так і неочевидні речі на наприклад дату редагування заявки. Тому щоб продовжити аналіз нам спершу треба виділити сутності нашої системи. Спершу розглянемо бізнес процес, який ми досліджуємо у вигляді UML діаграми

прецедентів (use case) та зображено на рисунку 1.1.

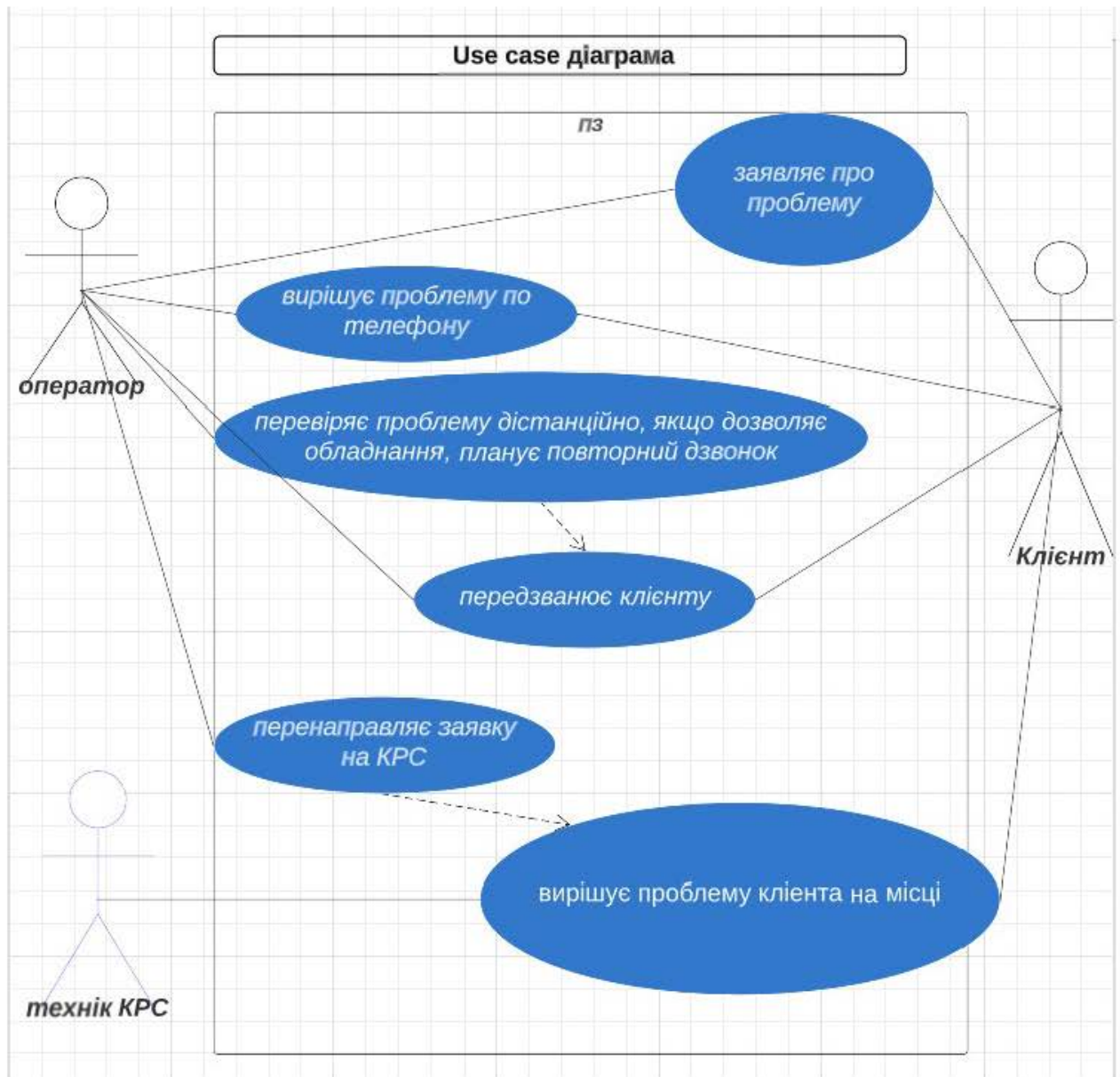


Рис. 1.1 UML діаграма прецедентів

По діаграмі ми можемо зробити висновок, що ключовою сутністю системи є заявка, а акторами – оператор, технік КРС і клієнт. Із діаграми витікає що система повинна відображати потоки комунікації між акторами.

Для комунікації з клієнту достатньо телефона, тому клієнту доступу до системи не потрібно, а для оператора і техніка ми зробимо інтерфейси, для

виконання дій зображених на діаграмі. Для комунікації оператор-клієнт, оператор-технік КРС, технік КРС-клієнт, нам потрібні додаткові дані про звернення і про самого клієнта, які ми будемо зберігати у 2 сутностях заявка і клієнт, де в сутності заявка будуть міститись дані про заявку, а в сутності клієнт дані про клієнта, причому ці сутності повинні бути з'єднані відношенням 1 до багатьох, де одному клієнту відповідає багато заявок.

Також ми можемо виділити основні сутності системи, вони знадобляться нам для проектування, спочатку ER-діаграми, а потім і бази даних, а згодом і структуру програмних класів.

Основні сутності:

- користувач(оператор, технік КРС);
- заявка;
- клієнт.

Сутність користувач має такі властивості:

- id;
- ПІБ;
- роль (оператор чи технік КРС);
- дані для входу

Сутність заявка має такі властивості:

- id;
- клієнт;
- статус;
- коментар;
- дата і час створення звернення;
- дата і час останньої зміни звернення;

Оскільки ми формалізували що таке заявка, ми можемо більш детально розкрити права на редагування заявки (таблиця 1.3.2).

З таблиці видно що кожен з працівників бачить тільки ті статуси заявок з якими він працює. Це допомагає швидше орієнтуватися у списках заявок.

Таблиця 1.3.2 – Видимість заявок

Статус	Оператор	Технік КРС
створена	+	-
потребує дзвінка	+	-
потребує візиту майстра КРС	-	+
оброблена	-	-

Також для зручності треба організувати вивід на головному екрані кожної з ролей тільки актуальні заявки для праці – для оператора у статусі потребує дзвінка, а для технік КРС у статусі потребує візиту майстра КРС.

Також треба обмежити можливість змінювати статус заявки на невірний, щоб виключити людську помилку. Так майстер КРС не може перевести у статус “створена”, чи у статус “потребує дзвінка”.

Отже ми формалізували бізнес процес зміни заявки, і маємо готову задачу для реалізації або пошуку вже реалізованого рішення, що задовольняє потреби бізнесу.

Вертаючись до проектування даних, треба зазначити зв'язки між сутностями:

- Оператор-абонент – багато до багатьох;
- Оператор КРС-абонент – багато до багатьох.

Зазвичай такий вид зв'язку потребує додаткової сутності, яка буде виступати зв'язком між працівником і абонентом. У нас у ролі такої сутності виступає сама заявка.

І водночас заявка є одиницею комунікації між клієнтом і провайдером, представником, якого залежно від суті заявки є або оператор відділу техпідтримки, або технік КРС.

1.4 Висновки до першого розділу

Наша система повинна реалізовувати сутності та їх відношення між собою згідно нашого опису задачі. Також на потрібна можливість мати особистий кабінет для кожної ролі користувачів(оператор та технік КРС) з розділенням прав, з обмеженням інформації та можливості робити тільки ті дії які обумовлені посадовими інструкціями працівників в залежності від ролі.

Система повинна контролювати дії працівників, виключаючи дії, що не входять до посадових інструкцій та виключати можливість помилок.

З економічних вимог до системи ми можемо сформулювати максимально низьку вартість розробки, підтримки, обслуговування, а також легку заміну спеціалістів для розробки ПЗ(у випадку коли система буде розроблятися с нуля).

З технічних вимог ми можемо виділити одночасність редагування даних декількома користувачами, безпеку особистих даних клієнтів та користувачів, та багатоплатформність(можливість роботи на багатьох пристроях та з любого місця).

Виходячи з цього оптимальним рішенням є веб-додаток. А який саме веб-додаток, і його структуру ми визначимо у наступному розділі оглянувши існуючі варіанти вирішення цієї задачі.

РОЗДІЛ 2.

ОГЛЯД ІСНУЮЧИХ РОЗВ'ЯЗКІВ ПОСТАВЛЕНОЇ ВИЩЕ ЗАДАЧІ, ЇХ АНАЛІЗ ТА ВИБІР МЕТОДІВ ВИРІШЕННЯ

2.1 Огляд та аналіз існуючих аналогів

Оскільки ми у висновках попереднього розділу визначились з тим ж наше рішення буде у сфері веб (що забезпечує доступність на всіх видах пристроїв через браузер), тепер нам треба конкретизувати у якому вигляді буде це рішення.

У доступі є багато різних варіантів використання, що застосовуються різними компаніями для фіксації комунікацій з клієнтами, від дуже простих до дуже складних. Можно спостерігати закономірність між тим наскільки легко буде ввести в експлуатації працівників і кількості можливостей, що дають готові рішення.

Найбільш очевидним і простим варіантом вирішення задачі, що буде максимально легким для навчання працівників є просто файл який можна редагувати онлайн декількома працівниками одночасно. Таке рішення надає багато сервісів, найбільш гнучким і зрозумілим з них є Google Sheets, що фактично є аналогом програм Microsoft Excel в онлайн доступі. До переваг цього рішення можна віднести той факт, що майже 100% людей колись працювали з Excel і тому не буде проблем з інтеграцією цього рішення у робочий процес. У нашому випадку такий файл буде мати вигляд зображений на рисунку 2.1.1.

	A	B	C	D	E	F	G
1	номер заявки	Статус заявки	Ім'я абонента	контактний телефон	коментар до заявки	Дата оформлення заявки	
2	1	потребує візита майстра КРС	Олександр	380933423423	Пошкоджено кабель, треба виклик майстра КРС	25.02.2023	
3	2	створена	Василь	380933423423	потребує уточнень	28.02.2023	
4	3	оброблена	Олена	380933423423	все виконано	03.03.2023	
5	4	потребує візита майстра КРС	Владіслав	380933423423	пошкоджено обладнання	06.03.2023	
6	5	оброблена	Сергій	380933423423	все виконано	09.03.2023	
7	6	створена	Марія	380933423423	чекаємо повторного дзінка клієнта	12.03.2023	
8	7	потребує візита майстра КРС	Олексій	380933423423	проблеми з кабелем у квартирі абонента	15.03.2023	
9	8	оброблена	Олександр	380933423423	все виконано	18.03.2023	
10	9	потребує візита майстра КРС	Іван	380933423423	Пошкоджено кабель, треба виклик майстра КРС	21.03.2023	
11	10	потребує дзвінка	Юлія	380933423423	клієнт не вдома, треба передзвонити	24.03.2023	
12	11	оброблена	Степан	380933423423	все виконано	27.03.2023	
13	12	потребує візита майстра КРС	Елізавета	380933423423	Пошкоджено кабель, треба виклик майстра КРС	30.03.2023	
14	13	потребує візита майстра КРС	Марина	380933423423	нема світла на обладнанні	02.04.2023	
15	14	створена	Федор	380933423423	перетелефонувати коли клієнт буде вдома	05.04.2023	
16	15	оброблена	Сергій	380933423423	все виконано	08.04.2023	
17	16	потребує дзвінка	Олена	380933423423	треба перевірити доступність обладнання та перетелефонувати	11.04.2023	
18							

Рис. 2.1.1 Приклад рішення у Google Sheets

Такий спосіб вирішення задачі разом із очевидними перевагами має декілька суттєвих недоліків. Важко шукати потрібну заявку коли заявок буде хоча б 100, працівникам буде важко визначити де саме їх заявка, і яку з них треба обробляти. Також складно швидко знайти яка з заявок найбільш пріоритетна і потребує першочергової уваги. Немає сортування, за статусом, датою, номером заявки. Технічна можливість є, але сортуючи заявки по статусу чи по даті, ми одразу відсортуємо для всіх користувачів документу, що може відволікти їх та ускладнити їм роботу. Також майже відсутня можливість модернізації такої системи у

напрямку розширення функціоналу, відокремленою інтерфейсів різних працівників, розділення ролей. Немає контролю за порядком зміни статусів.

З вищезазначеного видно, що цей варіант не підходить для нашої задачі, хоча він може використовуватися як проміжний доки не буде обраний оптимальний, або на дуже крайній випадок. Тому треба шукати більш складний варіант рішення нашої задачі.

До більш складних варіантів відноситься низка веб рішень для бізнесу, яких дуже багато під різні задачі та під різні бізнес процеси. Тому перш ніж вибирати саме рішення, треба визначитись з видом рішення яке нам потрібно. Умовно є такі види систем для реалізації бізнес-процесів, як CRM, ERP, BPMS, ITS, тощо. Слід зазначити, що не одне із рішень на ринку не можна повністю віднести до однієї з систем, бо зазвичай бізнес складно формалізувати, і кожне з рішень закриває потреби не одного виду, а багатьох. Часто в одній системі є ознаки одразу усіх видів, тому будь-яке готове рішення частіше всього перевантажено функціоналом. Це прямий наслідок того, що розробники конкретної системи зацікавлені у тому щоб вона підходила максимальній кількості бізнесів.

ITS(Issue tracking system) – це системи що керують списками проблем та підтримують їх. Системи відстеження проблем зазвичай використовуються в умовах спільної роботи, особливо у великих або розподілених колективах, але також можуть використовуватися окремими особами як частина режиму управління часом або особистої продуктивності. Ці системи часто включають розподіл ресурсів, облік часу, управління пріоритетами та робочий процес нагляду на додаток до впровадження централізованого реєстру проблем.

CRM(Customer relationship management) – система управління зв'язками з клієнтами у широкому сенсі. Основний акцент у таких системах робиться на відстежуванні історії взаємодії з клієнтами(їх купівлями, зверненнями за консультаціями, скаргами, тощо).

ERP(enterprise resource planning) – це системи, що забезпечують облік та звітність відносно використання ресурсів підприємства. До ресурсів підприємства можна віднести:

- робочий час працівників;
- розхідні матеріали необхідні для роботи підприємства;
- товари підприємства;
- фінанси;
- майно на балансі підприємства;
- витрати на енергоносії;
- загалом будь, що витрачається і потребує обліку.

Окремим випадком ERP систем є WMS(warehouse management system) – система управління складом, товарами, розхідниками, паливом, тощо.

Також до ERP систем з деякою поправкою(якщо розглядати працівників як ресурс), можна віднести HRMS(human resource management system) – систему управління працівниками. Ці системи направлені на облік робочого часу працівників, штатного розкладу, відділів, навичок та іншого що пов'язано безпосередньо з людиною, як із робочою одиницею.

Аналізуючи вимоги до нашого ПЗ, які були у висновках попереднього розділу ми бачимо, що нам найбільш підходить типи систем CRM і ITS. На перший погляд нам буде достатньо тикет-системи(ITS), але зважаючи на необхідність подальшого розширення функціоналу краще щоб у наявності у підприємства була CRM з можливістю відстежувати всю історію комунікації з кожним клієнтом.

Оскільки ми розглядаємо існуючі CRM-системи доцільно звернутися до списку найбільш розповсюджених систем. Розповсюдженість системи дасть нам можливість зменшити вірогідність видалення системи, та ліквідації наших даних. Список розповсюджених CRM систем візьмемо з наукового центру досліджень [research.com](https://www.research.com). Найактуальніший їх звіт датовано 2 червня 2023 року [3]. У звіті наведено 15 різноманітних CRM систем, що на думку потребують уваги. Але слід зазначити, що в основному це CRM які орієнтовані на англомовний бізнес та його реалії(ціні, бізнес-процеси, законодавство, тощо), тому для повного огляду існуючих рішень ми повинні також скористуватися і рейтингами для українського сегменту бізнесу [4,5].

Наразі отримали доволі широкий вибір, з вітчизняних та зарубіжних CRM

систем. Тепер наша задача відкинути зайві.

Salesforce Sales Cloud – це одна з систем CRM, яка забезпечує комунікацію з клієнтами, продажах і маркетингу. Платформа призначена для обслуговування компаній, які є бізнес-клієнтом (B2C) і бізнес-бізнесом (B2B).

Це дуже велика и гнучка система, але у нашому випадку вона має 2 істотні недоліки:

- велику вартість;
- складність введення в експлуатацію

Тож не зважаючи на багатий функціонал, ми вимушені її відкинути.

HubSpot Sales Hub – це інтелектуальна система електронної комерції, яка дозволяє команді продажів продавати більше за менший час, а також створювати та автоматизувати процеси продажів, які є специфічними для компанії.

Зважаючи на те що вона націлена більше на продажі, маркетинг, та супровід клієнта, що вже купив, ми відкидаємо її, бо у нашій задачі не входить реалізація бізнес-процесу продажу.

ActiveCampaign – це програмне забезпечення CRM, яке ідеально підходить для малого та середнього бізнесу, який прагне розвиватися за рахунок покращення відносин із клієнтами. Завдяки інтегрованим CRM і функціям автоматизації продажів ActiveCampaign пов'язує свої можливості електронної пошти та обміну повідомленнями зі зручною та потужною платформою автоматизації маркетингу. ActiveCampaign дозволяє компаніям створювати власні маркетингові робочі процеси та автоматизовані подальші послідовності, одночасно відстежуючи поведінку в кількох маркетингових каналах.

Тут теж акцент на маркетингу та продажу товарів, тож це не наш випадок.

Zoho CRM – це веб-система CRM, яка допомагає компаніям залучати, підтримувати та задовольняти клієнтів, щоб розвиватися. Керування потенційними клієнтами та контактами, управління продажами та функції контролю закупівель Zoho CRM є його основними можливостями. Завдяки використанню різних платформ Zoho CRM може відстежувати продажі, автоматизувати щоденні бізнес-операції та залучати клієнтів.

Zoho є дуже великою системою, що частково покриває функції ERP та HRM з гнучкою системою оплати. Про цю CRM було би варто замислюватись, коли б ми намагались автоматизувати всю роботу провайдера включаючи цикл продажу, за залучення нових клієнтів, а для нашої задачі, ця система занадто перевантажена функціоналом.

Monday sales CRM надає командам продажів повний контроль над своїми продажами, дозволяє їм керувати контактами та обліковими записами, а також оптимізує післяпродажні процеси та можливість продажу. Основний акцент на продажах для малого і середнього бізнесу, тому знову нам не підходить

LeadSquared Sales + Mobile CRM — це платформа для автоматизації продажів і мобільного управління взаємовідносинами з клієнтами, яка допомагає підприємствам швидше збільшувати дохід. Він організовує та оптимізує процеси продажів, управління потенційними клієнтами та CRM у внутрішніх і польових відділах продажів. Тут уже нам підійде частина CRM там, де управління існуючими клієнтами, але ключовим аспектом цієї системи є дохід, в той час як у нас на підприємстві ключовим показником є SLA, тому в подальшому ми не зможемо вести облік KPI наших працівників.

SuiteDash – це хмарне програмне забезпечення клієнтського порталу, яке об'єднує всі бізнес-можливості та інструменти. Завдяки широкому функціоналу та доступній ціні він створений для малого та середнього бізнесу. SuiteDash надає користувачам зручну альтернативу використанню кількох систем, що може бути дорогим, трудомістким і заплутаним. Умовно кажучи SuiteDash це “швейцарський ніж”, а нам з нього буде потрібно лише декілька функцій, і вивчивши цю систему можна сказати що вони не реалізовані.

EngageBay All-in-One Suite поєднує продажі, маркетинг і обслуговування клієнтів на одній платформі. Ця CRM – це повністю веб-програмне забезпечення, яке дозволяє користувачам автоматизувати ключові бізнес-процеси та процедури для розвитку свого бізнесу. EngageBay надає користувачам всі інструменти, необхідні для ефективного охоплення поточних і потенційних клієнтів, збільшення трафіку та продажів.

Engagebay пропонує користувачам інструменти для всього, від створення цільових сторінок до керування потенційними клієнтами. У ньому є модулі для електронного маркетингу, CRM, продажів, чату та автоматизації маркетингу. Він також може допомогти користувачам керувати внутрішніми процесами, надаючи такі інструменти, як планувальник, керування контактами, конвеєр угод і завдання, які можна призначати. Крім того, процеси CRM і модулі продажів і маркетингу можуть бути автоматизовані.

Engagebay пропонує широкий спектр інструментів CRM, які включені в інформаційну панель із інтуїтивно зрозумілим інтерфейсом. Програмне забезпечення не потребує великих знань або технічних навичок у сфері продажів або маркетингових платформ. Крім того, у нього є безкоштовний план, який включає багато кількості функцій для спрощення та оптимізації процесу продажу.

Accelo – це хмарна програма, яка поєднує прості у використанні та ефективні інструменти для професійних послуг, щоб компанії могли просувати угоди, починати проекти та задовольняти клієнтів і співробітників. Він надає платформу CRM, яка без зусиль створює, розвиває та підтримує професійні відносини з клієнтами. Компанії можуть використовувати його для пошуку клієнтів, оплати, доставки клієнтської роботи, продажу та пошуку ресурсів.

Streak – це єдина система CRM, яка повністю створена в Gmail і працює в парі з усіма іншими вашими програмами Google Workspace. Користувачі можуть керувати всіма аспектами діяльності компанії за допомогою надійної CRM, інтегрованої в папку вхідних повідомлень Gmail. Крім основних функцій CRM, до всіх планів входить надійний набір інструментів електронної пошти, включаючи відстеження електронної пошти, шаблони електронної пошти, злиття та спільний доступ до електронної пошти.

Prospect CRM – управління бізнес-операціями дистриб'юторів, виробників і оптовиків. Він пропонує керування контактами для інформації про ваших клієнтів. не підходить за характером відносин з клієнтами.

NetSuite – це CRM-рішення, яке дозволяє компаніям централізовано керувати взаємодією з поточними та потенційними клієнтами, партнерами та

постачальниками. Це комплексна платформа CRM, яка поєднує в собі традиційні та розширені можливості, щоб допомогти підприємствам збільшити дохід і зростання. Він включає всі можливості традиційного рішення CRM, наприклад автоматизацію продажів (SFA), управління обслуговуванням клієнтів і автоматизацію маркетингу. Крім того, він має можливість керувати котируваннями, комісійними, прогнозами продажів і партнерськими відносинами, а також забезпечує безперебійний потік інформації протягом усього життєвого циклу клієнта.

Vtiger All-in-One CRM – може надати чудові можливості керування запасами на основі проектів і хмари, а також функції маркетингу електронною поштою. Форми веб-CRM від Vtiger допомагають зібрати релевантну контактну інформацію, визначити її пріоритетність і використати її для прийняття кращих бізнес-рішень.

Vtiger допомагає користувачам підтримувати контакти, автоматизувати повторювані завдання, відстежувати кампанії та складати обширний звіт про продажі, маркетинг і завдання підтримки, які виконуються в організації. Завдяки автоматизації всього робочого процесу спеціальні звіти можуть зосереджуватися на окремих характеристиках потенційних клієнтів за допомогою ряду фільтрів. Vtiger відстежує маркетингові кампанії з часом і допомагає визначити, що працює, а що ні. Потужна функція керування клієнтською базою Vtiger є найбільш гнучким рішенням, яке він пропонує.

Flowlu – комплексне рішення для управління бізнесом, яке поєднує в собі управління проектами, продажі, соціальну співпрацю та інструменти CRM для сучасного бізнесу. Відстежуючи діяльність співробітників і розуміючи їхні проблеми, це програмне забезпечення дозволяє керівникам підприємств бачити всі повсякденні дії, що відбуваються в компанії.

Spiro – це CRM-програмне забезпечення на основі штучного інтелекту, яке автоматизує робочі процеси продажів і навчається попутно, дозволяючи команді продажів стати більш продуктивною та ефективною, закриваючи більше угод, ніж будь-коли раніше. Користувачі можуть створювати чіткі та точні звіти про продажі

та прогнози за допомогою розумних і потужних інструментів звітності. Spigo перетворює будь-яку команду продажів на потужні підрозділи, що завершують продажі, які приносять більший, ніж будь-коли раніше, дохід своїй організації.

Переходимо до вітчизняних топів.

CRM-лінійка Creatio – хмарна CRM-система для великих та середніх компаній, яка допомагає об'єднати та прискорити процеси продажу, маркетингу та сервісу, а також операційні процеси організації. Creatio – це платформа, яка поєднує функціонал CRM та BPM (управління бізнес-процесами). Компанією Terrasoft розроблено три основні продукти: для управління продажами, маркетингом і сервісом – Sales Creatio, Marketing Creatio, Service Creatio, Studio Creatio. Також клієнтам доступні десятки готових рішень в онлайн-каталозі Marketplace. Для нас це занадто складне рішення.

Pipedrive – переможець премії «Expert's Choice Award» за 2017 рік. У великих тестах на всі популярні рішення на ринку команда експертів виявила, що Pipedrive є найефективнішою CRM-системою для управління конвєсрами продажів для постачальників та підприємств, що розвиваються, а великий набір функцій для компаній безперечно це підтверджує. Але це не відповідає нашій задачі.

LP-CRM – CRM-система для товарного бізнесу, яка створювалася з метою особистого користування. Ми створили те, чого потребували самі. Система не підходить нам із-за націленості на товарний бізнес, що не відповідає нашому опису задачі.

HugeProfit – це система управління товарним бізнесом, яка призначена для автоматизації основних фінансових процесів: обліку залишків товару, контролю продажів та наочного відображення доходів та витрат. Система не підходить нам із-за націленості на товарний бізнес, що не відповідає нашому опису задачі.

KeerinCRM – проста та функціональна CRM-система для роботи компанії та автоматизації процесів. В системі є все від роботи з лідами до ведення фінансів, також перевагою системи є велика кількість інтеграцій. Бізнес-процеси реалізовані тут не відповідають нашому завданню.

CleverBOX:CRM – готовий набір інструментів для керування салоном краси,

клінікою, центром. Бізнес - процеси нашого напряму бізнесу не схожі на реалізовані у цій CRM.

KeyCRM – українська CRM-система із фокусом на товарний бізнес. Не підходить бо у нас не товарний бізнес.

PERFECTUM CRM+ERP – український продукт. Система може покривати всі процеси компанії. Має галузеві рішення для різних напрямів. Може бути доступна у хмарній або коробковій версіях з мобільним додатком. Модульна структура дозволяє розширювати функціонал та кастомізувати її під себе. У нас немає потреби в ERP.

Microsoft Dynamics 365 дозволяє вибудовувати бізнес-процеси згідно з ефективними моделями, виробленими в компанії (у продажах, маркетингу, обслуговуванні) або сценаріями, запропонованими бізнес-логікою системи. Це дуже крута і сучасна система с багатьма функціями і великою ціною. Не підходить з економічних міркувань.

SalesDrive – зручна українська CRM-система для онлайн-магазинів. Не підходить напрям бізнесу.

OneBox OS орієнтована на підприємців, хто потребує вести управлінський облік. Не підходить оскільки у нас основна задача не управлінській, а операційний облік.

Asteril – Досить потужна система управління бізнесом, що поєднала у собі клієнтську базу, відстеження статусів накладних, контролю фінансів, складського зберігання та спілкування з покупцями. Прискорює обробку запитів: завдяки CRM Asteril більшість заявок буде потрапляти до структури автоматично, усуваючи необхідність рутинної роботи. Дуже багато функціоналу, тому буде складно ввести у експлуатацію.

Сервіс Esports – повністю автономне хмарне програмне забезпечення для створення CRM, у простому інтерфейсі програми об'єднані проекти, люди та безліч можливостей. Використовувати його можна в будь-якому напрямку підприємництва – банківська справа, торговельна діяльність, call-центри, виробництво та електронна комерція. Сервіс Esports підійде для галузі освіти,

охорони здоров'я та навіть туризму.

За допомогою Peopleforce можна сформувати культ високої продуктивності та чітко слідувати йому, збільшуючи показники ефективності компанії. Оскільки основний акцент на управлінні, то нам не підходить.

OnePage – ця CRM-платформа орієнтована на компанії, у яких важливе місце відводиться процесу моніторингу завдань. Інтерфейс майданчика дозволяє відслідковувати статус угод, взаємодію з клієнтськими контактами, перелік дій, завдяки яким відбувся продаж.

U-ON.Travel – це CRM-система для туристичного бізнесу. Дозволяє ефективно керувати туристичною компанією та її бізнес-процесами з будь-якого комп'ютера або планшета, розташованого у будь-якій точці світу. Не підходить із-за вузької направленості.

EasyMS – легка хмарна система управління готелем, хостелом, апартаментами. Прекрасний модуль онлайн бронювання, синхронізація з онлайн каналами. Можливість зміни цін на будь-який період лише за лічені секунди.

Коммо – це хмарна CRM-система, яка допомагає бізнесу покращувати управління відносинами з клієнтами. Вона надає інструменти для автоматизації маркетингу, продажу та обслуговування клієнтів, такі як управління контактами, продажами, завданнями та проектами. Завдяки інтеграції з соціальними мережами та електронною поштою Коммо допомагає збільшувати ефективність комунікації з клієнтами та підвищувати рівень задоволеності їхніх потреб.

АмоCRM – система має оптимальний набір функцій та чудово адаптована для налагодження систематичних продажів. У сервіс інтегрована зручна телефонія (до 50 провайдерів), є корпоративний чат для обговорення робочих питань, є загальна база заявок з різних джерел. Крім того, АмоCRM дозволяє відстежувати ефективність продажів та у разі чого коригувати роботу співробітників. З недоліків можна виділити хіба виняткову націленість на процес продажу, та країна походження (Російська Федерація).

Мегаплан – вкрай зручна CRM-система для малого та середнього бізнесу. Вона має значний набір функціональних переваг:

- забезпечує ефективне управління завданнями та проектами;
- дозволяє контролювати актуальні доручення;
- веде історію взаємодії із клієнтами;
- дозволяє моніторити динаміку продажів;
- підтримує виставлення рахунків.

Немає сенсу розглядати із-за країни походження (Російська Федерація).

Бітрікс24 – цей CRM-сервіс, який користується величезною популярністю на міжнародному ринку. Він підтримує весь інструментарій, необхідний для налагодження ключових бізнес-процесів. Також російський сервіс. Єдиною значною перевагою є проста і зручна інтеграція з CMS Bitrix, але у нас немає задачі інтеграції з сайтом.

2.2 Обґрунтування оптимального варіанта реалізації мети кваліфікаційної роботи

2.2.1 Висновки аналізу існуючих аналогів

Оглянувши велику кількість різних готових CRM систем, можемо сказати що не одна з них не задовольняє наші потреби повністю.

Критерії за якими було відкинуто оглянуті системи:

- складність введення в експлуатацію;
- нереалізовані потрібні нам бізнес-процеси;
- висока вартість відносно корисності у нашому випадку;
- CRM зроблена у РФ, або дані зберігаються на їх серверах, що є порушенням вимог безпеки, та європейського законодавства щодо особистих даних користувачів, та клієнтів;
- велика кількість налаштувань, складний інтерфейс, що ускладнює можливість швидкої інтеграції у робочий процес підприємства.

Оскільки всі з оглянутих систем мають свої недоліки, і не задовольняють нашим критеріям CRM системи перелічених у першому розділі, робимо висновок, що треба реалізовувати систему с нуля своїми силами.

2.2.2 Вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу.

Будь-яке програмне забезпечення складається з декількох частин, що в поєднанні дають набір інструментів для реалізації і розробки ПЗ. Таку сукупність інструментів називають стеком технологій. Звичайно він складається з 3 основних частин, але у нашому випадку це веб-додаток, з клієнт-серверною архітектурою, тому у нас складових буде більше:

- операційна система;
- веб-сервер;
- мова програмування;
- база даних.

Тому, вибір стеку зводиться до вибору цих 4 пунктів. Слід зауважити, що можливо використання декількох технологій у кожному з пунктів, наприклад декількох мов, декількох видів баз даних, або веб-серверів.

Для нас є такі важливі характеристики технологічного стеку:

- можливість підтримки коду ПЗ великою кількістю розробників;
- можливість швидко включати нових розробників у роботу над програмним продуктом;
- мінімальна вартість розробки;
- мінімальна або відсутня плата за експлуатацію ПЗ(хостинг, ліцензії ПЗ, тощо);
- швидкість реалізації та майбутнього удосконалення системи.

2.2.3 Вибір операційної системи.

Фактично у нас є тільки 2 основних варіанта на вибір із систем за допомогою яких можна реалізувати веб додаток, це сімейство windows server або *nix (Unix, Linux, FreeBSD, тощо). Керуючись розповсюдженістю можна зробити висновок на користь якогось простого дистрибутива linux, який надається більшістю хостинг провайдерів, і є безкоштовним. Вибір windows server може бути актуальним, лише у випадку якщо для розробки веб-додатку ми будемо використовувати мову програмування C#. Отже обидві операційні системи мають свої переваги та недоліки. Звернувшись до оцінки продуктивності [14] можемо зробити остаточний вибір на користь linux систем, із-за переваг по продуктивності роботи.

Оскільки одним з критеріїв є мінімальна або відсутня плата за експлуатацію ПЗ, то тут теж Windows Server програє, бо система Linux поширюється на умовах ліцензії особливого роду, відомої під назвою GNU-ліцензії (GNU General Public License, або GPL). GNU-ліцензія була розроблена для проекту GNU організацією Free Software Foundation, і ця ліцензія встановлює кілька умов на поширення та зміну вільного програмного забезпечення.

Візьмемо у якості серверного дистрибутива Ubuntu Server 22.04, як найбільш сучасну, просту у використанні, та більш підтримувану спільнотою систему.

2.2.4 Вибір мови програмування.

Веб-додатки в цілому, і CRM систему як окремий випадок веб-додатку можна реалізувати на різних мовах програмування, або різних комбінаціях мов програмування.

Фактично нам треба обрати декілька мов програмування для бекенду і фронтенду. Крім того у нас є очевидний вибір щодо використання мови гіпертекстової розмітки HTML, та таблиць стилів CSS.

Виходячи з наших технічних вимог до технологій які ми будемо використовувати та порівнявши відносну кількість спеціалістів, та їх зарплатні очікування, на українських сайтах пошуку роботи (djinni.co, work.ua, robota.ua тощо) можемо зробити висновок, що наш вибір для бекенду – це мова php.

Ось кілька аргументів за допомогою PHP при розробці веб-додатків:

- Широке поширення: PHP є однією з найпопулярніших мов програмування для веб-розробки. Він має велику спільноту розробників і обширну документацію. Це означає, що є безліч ресурсів, підтримку та готові рішення для своїх проектів;

- Простота вивчення та використання: PHP має простий та інтуїтивно зрозумілий синтаксис, який передбачає його вивчення для новачків у програмуванні. Більше того, PHP легко інтегрується з HTML та іншими мовами розміток, що робить його зручним для розробки динамічних веб-сторінок;

- Додаткові можливості: PHP пропонує широкий набір вбудованих функцій і бібліотек, які спрощують розробку веб-1. Він підтримує роботу з базами даних, обробку форм, створення графіків, виправлення електронної пошти та багато іншого. Завдяки цим можливостям можна створювати функціональні та інтерактивні веб-додатки;

- Висока продуктивність: PHP добре оптимізований для роботи на веб-сервері та обробки HTTP-запитів. Завдяки своїй архітектурі та оптимізації PHP дозволяє досягти високої продуктивності та забезпечує швидку роботу програм;

- Масштабування: PHP надає гнучкі інструменти та архітектуру для створення масштабованих веб-додатків. Можна легко керувати ростом ваших додатків і додавати нові функції без істотних змін у коді;

- Велика спільнота та підтримка: Завдяки популярності PHP у вас буде доступ до величезної спільноти розробників. Можна обмінюватися досвідом, отримувати поради, вирішувати проблеми та використовувати готові рішення, що значно прискорить роботу.

У цілому PHP пропонує зручність, продуктивність, масштабування і широкі можливості для розробки веб-додатків.

На даний момент актуальною версією є версія 8.2 її і будемо використовувати.

Для фронтенд частини, маємо безальтернативний варіант – javascript.

2.2.5 Вибір веб-серверу

Оскільки у якості мови програмування ми обрали php у нас є 3 основних варіанти організації веб-серверу:

- nginx + fpm;
- apache2;
- nginx + apache2;

Порівняння використання Nginx + PHP-FPM, Apache2 та Nginx + Apache2 для розробки на PHP залежить від різних факторів, таких як продуктивність, гнучкість, налаштування та легкість використання.

Кінцевий вибір залежить від уподобань та вимог проекту. Якщо важлива висока продуктивність та ефективне використання ресурсів, Nginx + PHP-FPM може бути гарним варіантом. Якщо вам потрібна велика гнучкість у налаштуванні та широка підтримка модулів, Apache2 може бути кращим. Nginx + Apache2 (reverse проху) може бути корисним для обробки великого обсягу статичних файлів і одночасних запитів. Більш докладно переваги та недоліки кожного зі способів відображено в таблиці 2.2.5.1.

У нас до сервера невеликі вимоги:

- легкість налаштування та підтримки;
- робота на мінімальних ресурсах;
- можливість одночасно обробляти багато з'єднань.

Загалом сучасна практика не використовує apache2 у чистому вигляді, і із таблиці видно, що цей підхід нам не дає значних переваг, тому цей варіант можна відкинути.

Оскільки у нас є багато запитів і переважна більшість з них не статичні, а є результатом роботи php коду, то нам краще використовувати найбільш сучасний стек серверів nginx + fpm.

Таблиця 2.2.5.1 - Порівняння веб-серверів

	Переваги	Недоліки
nginx + fpm	<p>Nginx є легким та ефективним веб-сервером, який може обробляти велику кількість одночасних з'єднань. PHP-FPM (FastCGI Process Manager) забезпечує високу продуктивність виконання PHP-скриптів.</p>	<p>Налаштування Nginx та PHP-FPM може потребувати певного рівня досвіду.</p>
apache2	<p>Він легко інтегрується з іншими модулями та розширеннями, що робить його гнучким для розробки на PHP</p>	<p>Apache2 може споживати більше ресурсів системи, особливо під час обробки великої кількості з'єднань. Деякі користувачі також відзначають меншу продуктивність, ніж Nginx.</p>
nginx + apache2	<p>Дозволяє використовувати Nginx як проксі-сервер для статичних файлів та обробки навантаження, в той час як Apache2 займається виконанням PHP-скриптів. Це може призвести до ефективного використання ресурсів системи</p>	<p>Налаштування комбінації Nginx + Apache2 складніше, оскільки вам потрібно налаштувати обидва сервери та керувати конфігураціями обох.</p>

2.2.6 Вибір бази даних

Насамперед треба обрати тип бази даних реляційна чи об'єктно-орієнтована. Виходячи з посилань на наукові статті [22-27], можемо зробити висновок що для нас є оптимальним робота з реляційною СКБД, виходячи з таких переваг, як:

- строга структура даних заявок, клієнта, працівників(див. перший розділ);
- підтримка цілісності;
- невеликі вимоги до оперативної пам'яті;
- висока надійність цілісності, енергонезалежності даних;
- кращий захист особистих даних.

Тепер нам треба обрати реляційну базу даних. Трансформуючи вимоги з розділу 1, до вимог до бази даних, можемо виділити критерії:

- висока надійність;
- добрий захист даних;
- можливість одночасної роботи багатьох користувачів;
- низька вимогливість до ресурсів серверу;
- мінімальна або нульова вартість ліцензії та обслуговування.

Найбільш популярними безкоштовними базами даних є MySQL, PostgreSQL, SQLite, MongoDB. Розглянемо кожну з них.

MySQL: MySQL є однією з найпопулярніших баз даних, які використовуються з PHP. Вона надає широкі можливості для зберігання та керування даними, а також має розширену підтримку PHP через різноманітні бібліотеки та розширення. Для отримання додаткової інформації відвідайте офіційний веб-сайт MySQL.

PostgreSQL: PostgreSQL є потужною реляційною базою даних, яка підтримує широкий спектр функцій, включаючи складні запити, транзакції та географічні дані.

SQLite: SQLite є легким та вбудованим базою даних, добре підходить для малих проектів та локального розробки. Вона не вимагає окремого серверу баз даних і працює на рівні файлів, що дозволяє легко включати її у PHP-програми.

MongoDB: MongoDB є нереляційною (NoSQL) базою даних, яка забезпечує гнучкість схеми даних і швидкий доступ до документів, що робить її відмінним вибором для проектів з гнучкими вимогами до даних.

Складемо порівняльну таблицю баз даних на основі наукових статей [28-32], та проведемо аналіз яка з СКБД нам найбільш підходить.

Таблиця 2.2.6.1 - Порівняння СКБД

БД	Переваги	Недоліки
1	2	3
MySQL	Широко поширена і високо масштабована база даних. Швидкість та продуктивність. Велика кількість документації та ресурсів. Розширені можливості реплікації та кластеризації. Підтримка транзакцій та індексації.	Обмежена підтримка деяких складних запитів. Підтримка декількох одночасних записів може бути обмеженою. Менш розвинуті можливості для роботи з JSON. Обмежені можливості географічних даних.
PostgreSQL	Повна підтримка стандартів SQL. Підтримка розширень та налаштувань. Багаті можливості для роботи з геоданими. Потужна система транзакцій та безпеки. Розширені можливості оптимізації запитів.	Вимагає більшої кількості ресурсів в порівнянні з іншими базами даних. Є обмеження на швидкість запису та оновлення. Складніша налаштування та адміністрування.

SQLite	Легка, портативна та вбудована база даних. Не потребує окремого сервера та налаштувань. Ідеальна для невеликих проектів та вбудованого використання. Малі вимоги до ресурсів.	Обмежена масштабованість і низька продуктивність для великих проектів. Не підтримує деякі продвинуті функції, такі як реплікація та транзакції. Відсутність підтримки віддаленого доступу.
--------	---	--

Продовження таблиці 2.2.6.1

1	2	3
MongoDB	Гнучкість схеми даних та географічні запити. Швидкий доступ до документів та горизонтальне масштабування. Нереляційна структура та підтримка розподіленої бази даних. Підтримка високої доступності даних.	Високе споживання пам'яті та ресурсів. Підтримка складних запитів може бути складнішою. Відсутність транзакцій в раніших версіях. Потребує більшої уваги до безпеки та конфігурації.

MongoDB відкидаємо одразу, оскільки вона нереляційна. SQLite нам не підходить із-за низької продуктивності. PostgreSQL хоча і має переваги, але є складною у налаштуванні, а переваги роботи з географічними даними нам не потрібні, тому обираємо MySQL.

2.3 Висновки до другого розділу

У другому розділі ми розглянули та проаналізували існуючі аналоги вирішення поставленої задачі, та дійшли до висновку, що існуючі аналоги не підходять, і треба робити своє рішення у вигляді веб-додатку.

Огляд можна вважати повним, тому що ми крім типових рішень, розглянули також дуже прості, та дуже складні системи, що популярні у світі, та в Україні.

У рамках другого розділу був обраний технологічний стек оточення який буде уточнений у третьому розділі де буде розглянута практична частина кваліфікаційної роботи.

РОЗДІЛ 3.

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ

3.1 Опис алгоритму і програмного забезпечення

Для опису алгоритму будемо керуватись бізнес процесом описаним у першому розділі. Ми маємо дві окремі та незалежні одне від одного ролі користувачів:

- оператор відділу техпідтримки;
- технік кабельно-ремонтної служби.

Алгоритм роботи нашого веб-додатку ми будемо розглядати у розрізі кожної ролі окремо.

Типовий алгоритм для кожної ролі буде складатись з послідовних дій, авторизації, роботи з заявами, і виходу із системи, якщо це необхідно. Зобразимо його у вигляді простої блок-схеми (рис. 3.1.1), елементи якої поступово деталізуємо у наступних кроках.

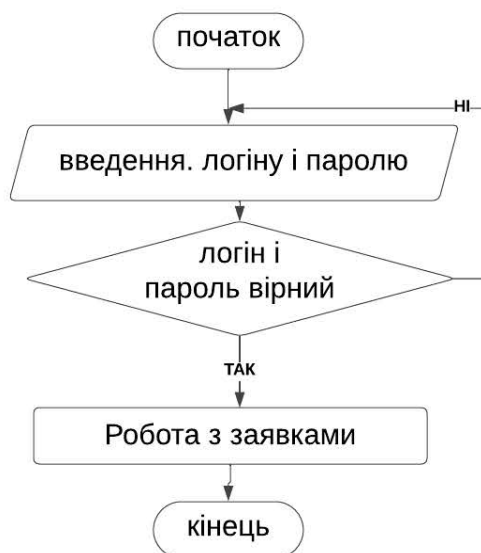


Рис. 3.1.1 Абстрактна блок-схема для обох ролей.

Блок авторизації спільний для обох ролей і працює однаково. А блок “Робота з заявками” є циклічним і потребує окремої схеми для кожної ролі.

Перелічені функції для кожної ролі перелічені у таблиці 1.3.1. Для ролі оператор у нас доступні функції створити заявку, переглянути заявку, змінити заявку. Для ролі технік КРС – лише переглянути і змінити.

Розробимо блоксхему для ролі оператор відділу техпідтримки (рис. 3.1.2).



Рис. 3.1.2 Блок-схема для ролі оператор техпідтримки.

І зразу деталізуємо схему і для техніка з відділу КРС. Відмінність функцій техніка КРС у тому що він не може перевести заявку на оператора, чи створити нову заявку, тому вона значно простіше. Більш наглядно це видно на рисунку 3.1.3.

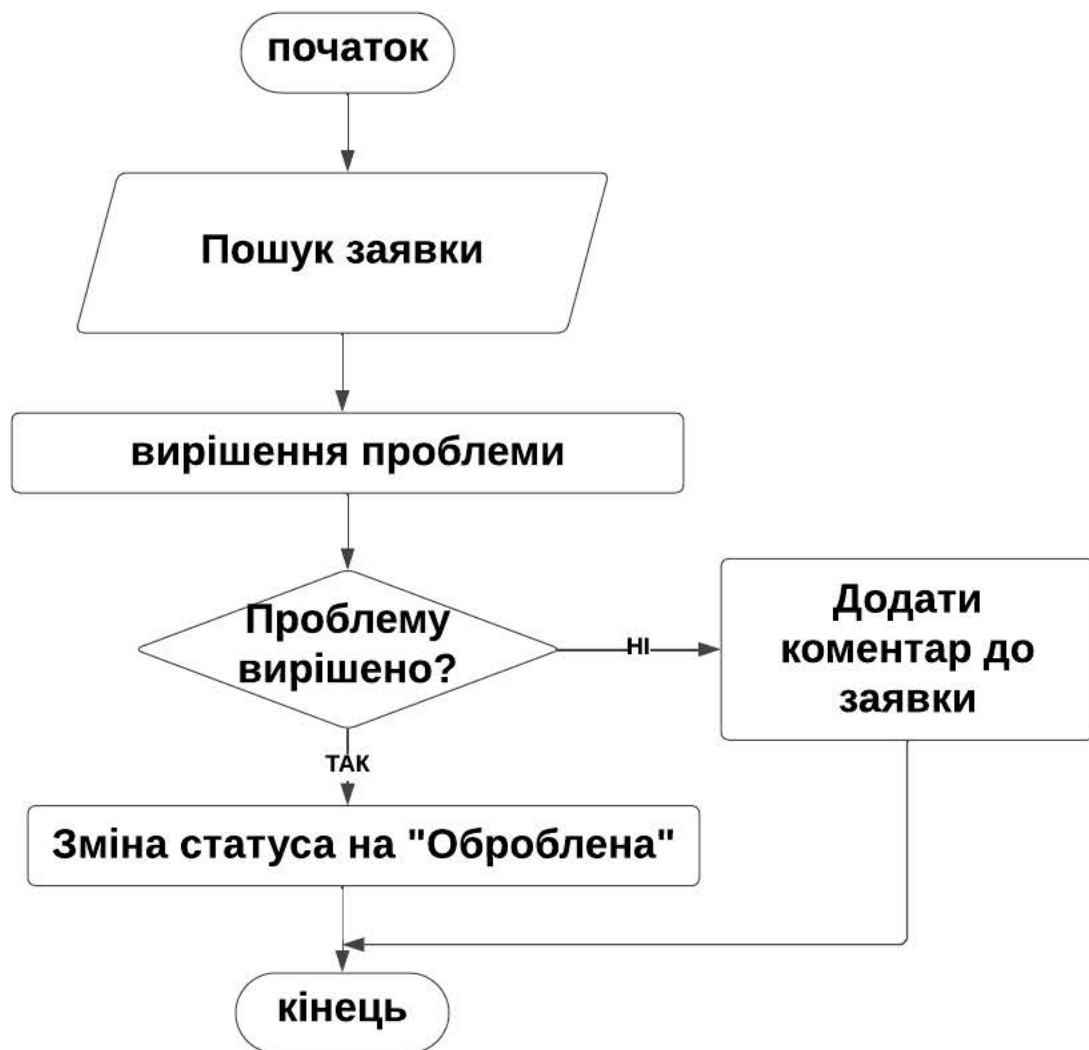


Рис. 3.1.3 Блок-схема для ролі техніка КРС

3.2 Вибір і обґрунтування структури проектування системи і її компонентів

Для спрощення розробки, зменшення строків, та коштів на розробку, треба вибрати CMS чи фреймворк який буде одночасно гнучким(зادля можливості подальшого вдосконалення програмного продукту) , популярним(для широкого вибору розробників), та з широкою екосистемою готових рішень типових задач, які ми можемо використати для економії часу на розробку, а отже і вартості.

Також нам необхідно обрати структуру системи. Фактично обираючи CMS

чи фреймворк ми автоматично також обираємо структуру, однак провівши аналіз існуючих структурних патернів, я зробив висновок, що треба обрати фреймворк або CMS, що дозволяють просто і зручно використовувати структуру MVC.

Основні переваги паттерна MVC (Model-View-Controller) у розробці на PHP:

1. Розділення логіки: MVC дозволяє розділити логіку додатку на три основні компоненти - модель, представлення та контролер. Це забезпечує легшу управління кодом та його повторне використання.
2. Модульність: Компоненти MVC можуть бути розроблені і тестовані окремо, що дозволяє розподілити завдання між розробниками та полегшити підтримку коду.
3. Розширюваність: Модель, представлення та контролер розділені, що дозволяє вносити зміни в один компонент без впливу на інші. Це полегшує розширення та модифікацію додатку.
4. Покращена організація коду: MVC надає структурований підхід до розробки, що полегшує розуміння та підтримку коду. Кожен компонент виконує свою відповідальність, що дозволяє зберігати код організованим та читабельним.

У рамках моделі MVC ми повинні виділити моделі які будуть використовуватись. Моделі це основна одиниця роботи з даними. Наші моделі будуть ґрунтуватися на сутностях, які ми виділили у першому розділі і являють собою відображення нашої бази даних. Тому перед тим як проектувати моделі необхідним є проектування бази даних.

Проектування бази даних полягає у створенні ER-діаграми, а потім перетворення її у sql код для створення бази даних, який приведемо у додатку.

Виділимо ключі, та зв'язки у діаграмі, та додатково добавимо декілька полів с датами, для того щоб відслідковувати динаміку (рис. 3.2.1).

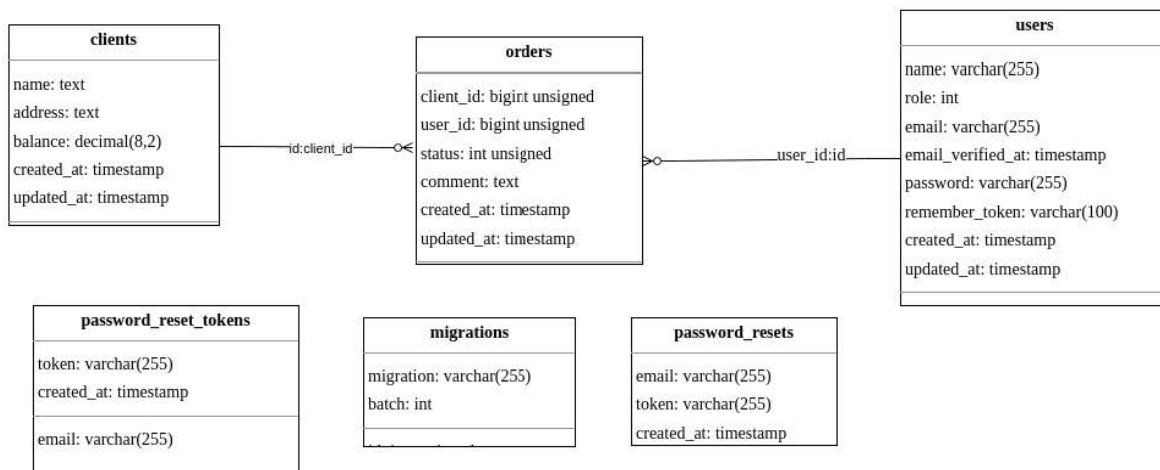


Рис. 3.2.1 ER-діаграма

У контролерах нашого проекту розмістимо реалізацію сторінок нашого веб-додатку.

Список сторінок проекту які однакові для обох ролей:

- авторизація;
- реєстрація;
- відновлення паролю;
- сторінка зі списком власних заявок техніка КРС;
- сторінка редагування(зміни статусу) і перегляду задач;
- сторінка зі списком не назначених на працівників задач, з
можливістю взяти до роботи

Список сторінок проекту які є тільки у оператора відділу підтримки:

- Створення заявки;
- список усіх заявок

Сумарно у нас будуть такі контролери:

- ConfirmPasswordController.php;
- ForgotPasswordController.php;
- LoginController.php;
- RegisterController.php;

- ResetPasswordController.php;
- VerificationController.php;
- KRSController.php;
- OperatorController.php;
- OrderController.php

По представленням все просто: одне загальне представлення для повторюваних компонентів, таких як меню, і по одному представленню для кожної з сторінок.

3.3 Опис використовуваного системного програмного забезпечення для розробки і експлуатації

Раніше у пункті 2.2.2 визначились з тим що наш веб-додаток буде працювати на так званому LNMP(Linux, Nginx, Mysql, PHP) стеку. Конкретизуємо:

- Ubuntu Server 22.04;
- php 8.2;
- nginx 1.25.1;
- mysql-server 5.7

Користувачу для комфортної роботи знадобиться будь-який сучасний браузер, не пізніше 2019 року релізу.

У локальному середовищі, де буде відбуватися розробка зручніше використовувати не linux, а windows, тому застосуємо одне з пакетних рішень для реалізації LNMP стеку. Візьмемо Openserver версії 5.4.3 на якому будемо розробляти веб-додаток.

3.4 Основні рішення з реалізації системи в цілому і її компонентів

Для початку визначемося з фреймворком що будемо використовувати. Сучасним лідером у світі php є фреймворк laravel. Серед іншого він дозволяє використовувати паттерн MVC.

Існує кілька причин, чому варто вибирати фреймворк Laravel для розробки веб-додатків:

1. Зручність та простота використання: Laravel надає чистий та елегантний синтаксис, що робить розробку веб-додатків більш інтуїтивною та швидкою. Він має зрозумілу документацію та безліч готових компонентів, що спрощує роботу та прискорює процес розробки.

2. Потужні функціональні можливості: Laravel пропонує широкий набір інструментів і бібліотек, які допомагають забезпечити високу продуктивність та розширюваність веб-додатків. Він підтримує такі функції, як маршрутизація, кешування, сесии, автентифікація користувачів та багато інших, що робить розробку додатків більш ефективною.

3. Велика активність співтовариства: Laravel має активне співтовариство розробників, яке постійно вносить внесок у вдосконалення фреймворку.

4. Безпека: Laravel забезпечує вбудовану систему захисту від атак, таких як хакерські атаки, перехоплення даних та інші загрози безпеки. Він має вбудовані функції шифрування, хешування паролів, захисту від міжсайтового скрипінгу(XSS) та інші заходи безпеки.

5. Підтримка бази даних: Laravel підтримує різноманітні бази даних, такі як MySQL, PostgreSQL, SQLite та багато інших. Також ORM спрощує роботу з базою даних за допомогою механізму моделей.

6. Масштабованість: Laravel надає зручні інструменти для масштабування веб-додатків. Можна легко розширювати функціонал додатку, додавати нові модулі та функції, що дозволяє додатку рости разом з

бізнесом.

7. Слідування структурі MVC.

Ці причини роблять Laravel привабливим вибором для розробки веб-додатків, особливо для проектів будь-якого розміру та складності.

Для реалізації фронтенд частини візьмемо css фреймворк bootstrap.

CSS-фреймворк Bootstrap має кілька переваг, які можна виділити:

1. Швидкість розробки: Bootstrap надає готові стилі та компоненти, що дозволяють розробникам ефективно працювати. Він забезпечує швидкий старт проекту, оскільки не потребує написання коду з нуля. Це збільшує продуктивність розробника та зменшує час, необхідний для розробки веб-сайту [37].

2. Адаптивний дизайн: Bootstrap пропонує адаптивну сітку та компоненти, що дозволяють веб-сайту гармонійно виглядати на різних пристроях та розмірах екранів. Це забезпечує кращий користувацький досвід і зручність використання.

3. Консистентність: Використання Bootstrap дозволяє створювати веб-сайти з єдиною стилістикою та дизайном. Він надає багато стандартних компонентів, які забезпечують одноковість у вигляді, розташуванні та інтерактивності елементів.

4. Підтримка браузерів: Bootstrap забезпечує сумісність з різними браузерами та версіями. Це дозволяє веб-сайту відображатися коректно на різних платформах та забезпечує широку аудиторію користувачів.

5. Розширюваність: Відкритий код Bootstrap дозволяє розробникам створювати власні компоненти та розширювати функціональність фреймворка. Це дає багато можливостей для індивідуалізації та пристосування фреймворка під конкретні потреби проекту.

Після вибору технологій перейдемо до реалізації основних компонентів.

Умовно ми можемо поділити веб додаток на декілька частин:

- модуль безпеки(авторизація, реєстрація, відновлення паролю);

- кабінет оператора техпідтримки;
- кабінет оператора КРС

Спочатку побудуємо базу даних за допомогою міграцій у laravel, проставивши необхідні ключі. Механізм міграцій допоможе нам відстежувати історію змін структури бази. Для цього використовуємо команди `php artisan make:migrations` і `php artisan migrate` для застосування у базу. Отримаємо готову до використання базу даних (див. додаток А).

Спочатку реалізуємо модуль безпеки, для створень представлень авторизації використаємо готовий пакет “laravel/ui” і створимо представлення для модулю авторизації командою `composer require laravel/ui`. Цей пакет надасть готові шаблони для сторінок авторизації, реєстрації, відновлення паролю.

```
php artisan ui bootstrap --auth
```

Ми отримали майже готову систему авторизації (рис. 3.4.1, 3.4.2, 3.4.3). Після незначних правок, що підвищують безпеку, та правок візуального вигляду маємо працюючі форми авторизації і відновлення паролю. Сторінку реєстрації, яку надає ларавел, ми видалимо, для того щоб наша система була максимально закрита.

CRM Провайдеру Вхід

Вхід

Email

Пароль

Запам'ятати мене

[Забудили пароль?](#)

Рис. 3.4.1 Сторінка входу

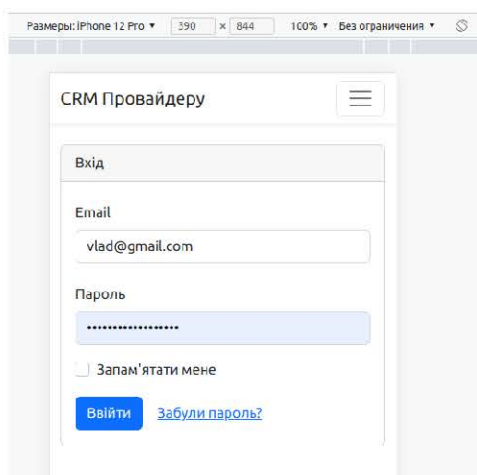


Рис. 3.4.2 Сторінка входу в режимі емуляції мобільного пристрою Iphone 12 Pro

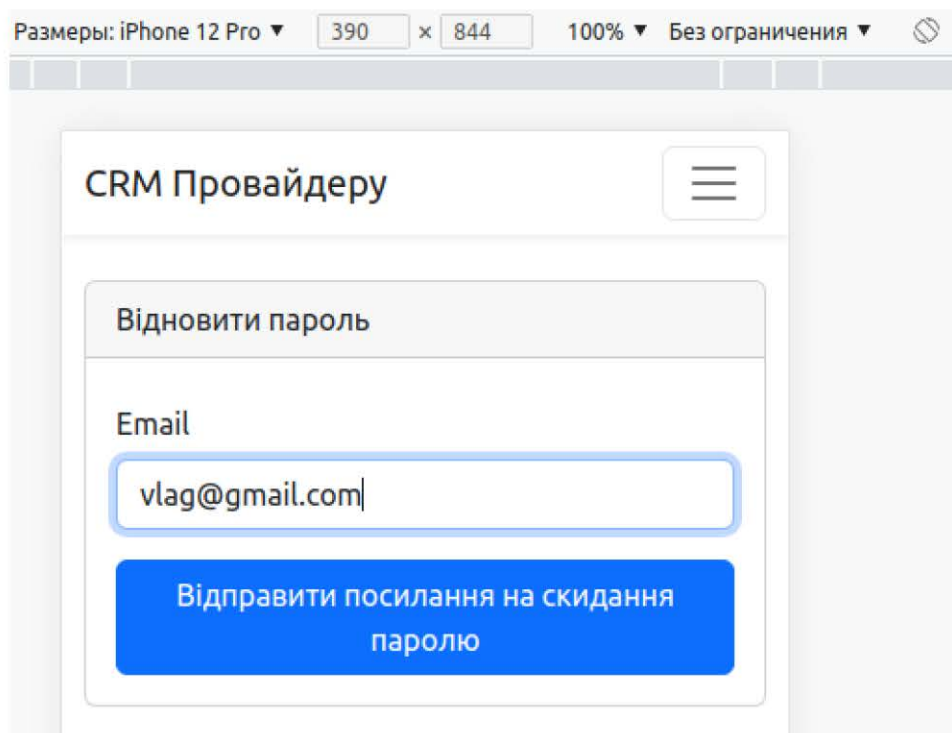


Рис. 3.4.3 Сторінка відновлення паролю в режимі емуляції мобільного пристрою Iphone 12 Pro

Модуль безпеки готовий переходимо до особистих кабінетів користувачів. Почнемо з техніка КРС і спільних сторінок з оператором. Спершу створимо форму, де будемо виводити заявки (рис. 3.4.4, 3.4.5), що адресовані користувачу, що

ввійшов у систему. Це однакова сторінка для обох ролей, і відрізняється тільки заявками, точніше їх статусами. Згідно таблиці 1.3.2, для оператора КРС виводимо лише ті заявки, які у статусі “потребують виклика майстра”, а для оператора техпідтримки “потребують дзвінка”. Для більш зручної роботи зі статусами заявок пронумеруйте їх, і в коді представимо константами класу моделі заявок:

```
const STATUS_OPEN=1;
const STATUS_NEED_CALL=2;
const STATUS_NEED_KRS=3;
const STATUS_CLOSED=4;
```

Для зручного і зрозумілого інтерфейсу використаємо bootstrap.

CRM Провайдеру

Заявки що потребують уваги

#	Клиент	Статус	Коментар	Дата та час звернення
4	Василенко	Потребує майстра КРС	немає пінгу	12:42 11-06-2023
6	Вознесенська	Потребує майстра КРС	зроблено	13:19 13-06-2023

Рис. 3.4.4 Сторінка зі списком власних заявок техніка КРС

CRM Провайдеру

Заявки що потребують уваги

#	Клиент	Статус	Коментар	Дата та час звернення
2	Герасимчук	Потребує дзвінка	Не терміново	15:11 11-06-2023
7	Василенко	Потребує дзвінка	немає пінгу	12:42 11-06-2023

Рис.3.4.5 Сторінка зі списком власних заявок оператора відділу підтримки

Далі розробимо сторінку вибору вільних задач по аналогічному алгоритму. На цій сторінці є можливість взяти до роботи нову задачу (рис. 3.4.6, 3.4.7). У цьому

випадку задача закріплюється за конкретним користувачем і після цього доступна на попередній сторінці.

CRM Провайдеру Вхід

Заявки що потребують уваги

#	Клиент	Статус	Коментар	Дата та час звернення	
8	Проколенко	Потребує майстра КРС	немає пінгу	12:42 11-06-2023	Взяти до роботи
10	Герасимчук	Потребує майстра КРС	поломався кабель	07:22 12-06-2023	Взяти до роботи

Рис. 3.4.6 Сторінка зі списком вільних заявок техніка КРС

CRM Провайдеру Вхід

Заявки що потребують уваги

#	Клиент	Статус	Коментар	Дата та час звернення	
11	Василенко	Потребує дзвінка	проблеми у налаштуваннях клієнта	12:42 11-06-2023	Взяти до роботи
12	Михайлов	Потребує дзвінка	проблеми на лінії	07:22 12-06-2023	Взяти до роботи

Рис. 3.4.7 Сторінка зі списком вільних заявок оператора відділу підтримки

3.5 Інструкція роботи користувача з системою

3.5.1 Інструкція роботи техніка КРС

Першим кроком роботи з системою технік повинен увійти у систему (рис. 3.5.1.1) ввівши свій логін і пароль, наданий керівництвом. Якщо авторизація з власного пристрою, рекомендується натиснути галочку “Запам'ятати мене”.

Рис. 3.5.1.1 Вхід техніка КРС

Далі слід перевірити, чи є термінові задачі, які потребують уваги (рис. 3.5.1.2). Одразу після авторизації з'явиться ця сторінка, але якщо ви давно авторизувалися на данному пристрою треба повторно перейти на пункт меню “Термінові задачі”.

Заявки що потребують уваги

#	Клиент	Статус	Коментар	Дата та час звернення
4	Василенко	Потребує майстра КРС	немає пінгу	12:42 11-06-2023
6	Вознесенська	Потребує майстра КРС	треба перезавантажити обладнання	13:19 13-06-2023
8	Прокопенко	Потребує майстра КРС	немає пінгу	12:42 11-06-2023
10	Герасимчук	Потребує майстра КРС	поломався кабель	07:22 12-06-2023

Рис. 3.5.1.2 Перша сторінка техніка КРС

Якщо такі задачі є, то приступити до виконання. Якщо таких задач нема, то перейти на пункт меню “нерозподілені задачі” (рис. 3.5.1.3) і взяти у роботу будь-яку задачу зі списку.

Заявки що потребують уваги

#	Клієнт	Статус	Коментар	Дата та час звернення	
8	Прокопенко	Потребує майстра КРС	немає пінгу	12:42 11-06-2023	Взяти до роботи
10	Герасимчук	Потребує майстра КРС	поломався кабель	07:22 12-06-2023	Взяти до роботи

Рис. 3.5.1.3 Нерозподілені задачі техніка КРС

Після виконання задачі, треба відмітити результат роботи по задачі. Для цього треба натиснути на заявку на сторінці “термінові задачі”. І відредагувати задачу в залежності від результату роботи (рис. 3.5.1.3).

Редагування заявки № 2

Клієнт: Герасимчук

Працівник: Микола

Дата звернення: 15:11 11-06-2023

Статус:

Коментар:

[Змінити](#)

Рис. 3.5.1.3 Зміна статусу чи коментаря техніком КРС

Після закінчення роботи с поточною заявкою майстер КРС обирає нову заявку для роботи.

3.5.2 Інструкція роботи оператора відділу підтримки

Першим кроком роботи з системою оператор відділу техпідтримки входить до системи вводячи свої авторизаційні дані (рис. 3.5.2.1), надані керівництвом.

Якщо авторизація з власного пристрою, рекомендується натиснути галочку “Запам'ятати мене”.

CRM Провайдеру Термінові задачі Нерозподілені задачі

Рис. 3.5.2.1 Вхід оператора відділу техпідтримки

Далі слід перевірити чи є термінові задачі які потребують уваги. Одразу після авторизації з'явиться ця сторінка, але якщо ви давно авторизувалися на данному пристрою треба повторно перейти на пункт меню “Термінові задачі” (рис. 3.5.2.2).

CRM Провайдеру Термінові задачі Нерозподілені задачі

Заявки що потребують уваги

#	Клиент	Статус	Коментар	Дата та час звернення
2	Герасимчук	Потребує дзвінка	Не терміново	15:11 11-06-2023
7	Василенко	Потребує дзвінка	немає пінгу	12:42 11-06-2023
11	Василенко	Потребує дзвінка	проблеми у налаштуваннях клієнта	12:42 11-06-2023
12	Михайлов	Потребує дзвінка	проблеми на лінії	07:22 12-06-2023

Рис. 3.5.2.2 Термінові задачі оператора відділу техпідтримки

Якщо такі задачі є то приступити до виконання. Якщо таких задач нема, то перейти на пункт меню “нерозподілені задачі” (рис. 3.5.2.3) і взяти у роботу будь-яку задачу зі списку.

CRM Провайдеру [Термінові задачі](#) [Нерозподілені задачі](#)

Заявки що потребують уваги

#	Клієнт	Статус	Коментар	Дата та час звернення	
11	Василенко	Потребує дзвінка	проблеми у налаштуваннях клієнта	12:42 11-06-2023	Взяти до роботи
12	Михайлов	Потребує дзвінка	проблеми на лінії	07:22 12-06-2023	Взяти до роботи

Рис. 3.5.2.3 Нерозподілені задачі оператора відділу техпідтримки

Після виконання задачі, треба відмітити результат роботи по задачі. Для цього треба натиснути на заявку на сторінці “термінові задачі”. І відредагувати задачу в залежності від результату роботи (рис. 3.5.2.4).

CRM Провайдеру [Термінові задачі](#) [Нерозподілені задачі](#)

Редагування заявки № 7

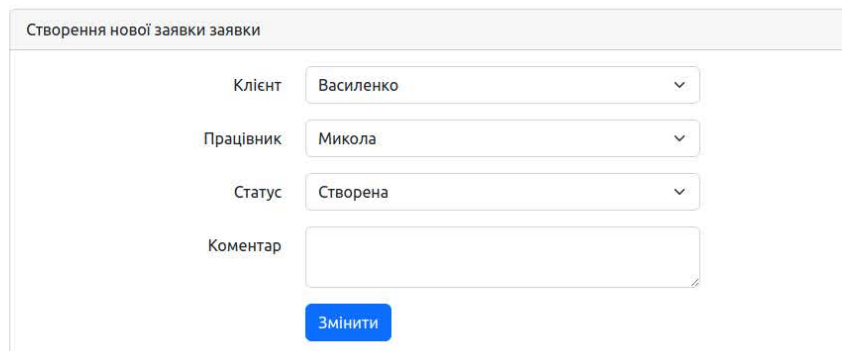
Клієнт	Василенко
Працівник	Владіслав
Дата звернення	12:42 11-06-2023
Статус	<input type="text" value="Потребує дзвінка"/>
Коментар	<input type="text" value="немає пінгу"/>

[Змінити](#)

Рис. 3.5.2.4 Зміна статусу чи коментаря оператором відділу техпідтримки

Після закінчення роботи с поточною заявкою майстер КРС обирає нову заявку для роботи.

Також у оператора техпідтримки, є функція приймати дзвінки від клієнтів. Коли клієнт заявив про проблему оператор повинен створити заявку і присвоїти їй відповідний проблемі статус (рис. 3.5.2.5).



Створення нової заявки

Клієнт: Василенко

Працівник: Микола

Статус: Створена

Коментар:

Змінити

Рис. 3.5.2.5 Створення нової заявки

Оператор також може призначити працівника який буде виконувати цю заявку. Якщо працівник не буде назначений, то заявка буде відображатись на сторінці “Нерозподілені задачі”, залежно від статусу відповідно ролі, до моменту коли хтось не візьме її у роботу. І далі ця заявка проходить шлях за циклом зазначеним вище, у ролі оператора або техника КРС.

3.6 Висновки до третього розділу

У третьому розділі розкривається практична частина кваліфікаційної роботи і технічні вимоги до веб-додатку(сервер, операційна система, програмний стек, веб-сервер).

Робота яка була виконана у третьому розділі показує, що можливо зробити якісний і корисний додаток з мінімальними витратами, але при цьому функціональній, простий і зручний, який може вирішувати поставлену задачу.

Вибір технологій хоча і є оптимальним, але не є єдиним, що підходить. Цю ж задачу можна було реалізувати на іншому оточенні, іншими мовами, чи з іншою конфігурацією сервера, але це б відобразилося на швидкодії, ціні чи подальшому розвитку веб-додатку.

Щодо можливого розширення CRM системи, то воно буде відбуватися у

напрямку реалізації інших бізнес-процесів підприємства, уточненню цього бізнес-процесу чи визначення точок контролю продуктивності працівників (KPI), і відстежування їх показників, або динаміки. Умовно можна всі можливі вдосконалення поділити на види:

- удосконалення інтерфейсу;
- додавання нових аспектів поточного бізнес-процесу;
- реалізація інших бізнес-процесів підприємства де задіяні ці працівники та клієнти;
- графіки та звіти що відображають динаміку у часі результатів роботи та ефективності конкретних співробітників, відділу, підприємства, тощо;
- додавання нових ролей що відповідають іншим працівникам інтернет-провайдеру.

В цілому CRM система зроблена відповідно поставленій задачі, і повністю виконує потрібні функції, але я напрями розвитку, які будут реалізовані у магістерській роботі.

ВИСНОВКИ

Об'єктом дослідження є бізнес-процес взаємодії технічної підтримки інтернет-провайдера та їх абонентів, а предметом є автоматизація бізнес-процесу обслуговування абонентів даного інтернет-провайдера.

Метою дипломного проекту є розробка CRM системи обліку заявок клієнтів для даного інтернет-провайдера.

Основною задачею дипломного проекту є знаходження способу автоматизації комунікації між відділом технічної підтримки та клієнтами за допомогою програмно-інформаційних засобів. В результаті вирішення задачі ми досягли таких результатів:

- Зменшення часу відгуку: CRM система дозволяє швидко отримувати та обробляти заявки клієнтів, що сприятиме скороченню часу відгуку технічної підтримки на проблеми абонентів;
- Покращення якості обслуговування: Завдяки автоматизації, інтернет-провайдер зможе забезпечити стандартну обробку заявок та послуг для клієнтів, що призведе до покращення якості надання послуг;
- Ефективніше використання ресурсів: Автоматизація допоможе оптимізувати розподіл ресурсів технічної підтримки, спрямовуючи їх на найбільш важливі та складні проблеми, що забезпечує ефективне використання часу та зусиль працівників;
- Покращення зв'язку з клієнтами: Система CRM дозволить зберігати та аналізувати інформацію про клієнтів, їх заявки та попередні контакти, що сприятиме покращенню зв'язку з клієнтами та індивідуальному підходу до їх потреб

Розробка CRM системи обліку заявок клієнтів інтернет-провайдера є важливим кроком у вдосконаленні сервісу підприємства та підвищенні задоволеності клієнтів, що забезпечує успішну роботу технічної підтримки та розвиток інтернет-провайдера, за рахунок утримання існуючих клієнтів, або

залучення нових.

В ході роботи було також розглянуто низку готових рішень, задач, але з багатьох причин вони не підходили нам, тому було прийнято рішення писати самостійно, не використовуючи аналоги. Слід зауважити, що аналогічні рішення, хоча і більш складні і більші, однак не вирішують нашої задачі, що свідчить про те, що краще писати ПЗ под кожен бізнес-процес окремо і не намагатись підлаштуватись під вже існуючі практики.

Із переваг нашої CRM системи можна виділити:

- простий зручний, інтуїтивно зрозумілий інтерфейс;
- підлаштування під саме наше підприємство і як наслідок, не перегруженість зайвими функціями, проста інтеграція в робочий процес;
- економічна цілесобразність;
- швидкий ввід у експлуатацію

З огляду готових рішень можна зробити висновок щодо напряму вдосконалення програмного продукту це:

- відстежування показників SLA, KPI тощо;
- можливість інформувати клієнта завчасно про візит майстра кабельно-ремонтної служби та дзвінок оператора ;
- підключення функцій білінгу та онлайн оплати клієнтом;
- сповіщення клієнтів про малий залишок на рахунку;
- додавання карти підключень, для спрощення відповіді щодо покриття інтернет-провайдеру.

Це мінімальна кількість напрямів розвитку CRM системи, можна багато ще сформулювати виходячи з аналогічних рішень, але перш за все ми повинні орієнтуватися на потреби бізнесу. Ці і деякі інші покращення будуть втілені у магістерській роботі.

Із недоліків створеного ПЗ можна виділити:

- мала кількість функціоналу;
- не дуже зручне додавання користувачів;

- малий об'єм даних, що стане зайвою роботою при розширенні функціоналу;
- мало інформованість клієнтів через цю CRM.

Більшість цих недоліків не заважає працювати інтернет провайдеру, проте їх усунення збільшить лояльність абонентів.

За підсумком я вважаю що задача зроблена згідно поставленої мети, а недоліки і напрямки для майбутньої розробки, є хорошою темою магістерської роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation). // Official Journal of the European Union. – 2016.
2. Технічний комітет зі стандартизації «Інформаційні технології» (ТК 20). ДСТУ ISO/IEC 27032:2016 Інформаційні технології. Методи захисту. Настанови щодо кібербезпеки (ISO/IEC 27032:2012, IDT) // ДП «УкрНДНЦ». – 2018.
3. «15 Best CRM Software in 2023», JUN 2, 2023, <https://research.com/software/best-crm-software>
4. SEO Ukraine / ТОП-10 найкращих CRM-систем в Україні / URL: <https://seoukraine.com.ua/top-10-naikrashchychk-crm-system-v-ukraini/>
5. Владислава Рикова, ТОП-20 CRM-систем для України (червень 2023) / <https://vlada-rykova.com/ua/crm-sistemy/>
6. Salesforce / The simple definition of CRM / URL: <https://www.salesforce.com/crm/what-is-crm/#>
7. Salesforce / Running business without CRM can cost you real money / URL: <https://www.salesforce.com/crm>
8. The CRM metrics: how to measure the performance of CRM / URL: <https://www.predictiveanalyticstoday.com/the-crm-metrics-how-to-measure-theperformance-of-crm/>
9. CRM Metrics Third Edition Paperback – October 17, 2018 by Gerardus Blokdyk
10. Шилук Олена Сергіївна / Впровадження інноваційних інформаційних технологій на підприємстві / URL: https://ela.kpi.ua/bitstream/123456789/28355/1/Shiluk_bakalavr.pdf
11. Криворучко К.М. / Використання хмарних технологій в сучасних системах автоматизації процесу управління підприємством / URL: http://dspace.kntu.kr.ua/jspui/bitstream/123456789/4631/1/46_2013-263-268.pdf
12. Бітрікс24 / Результати дослідження ринку CRM в Україні / URL: <https://www.bitrix24.ua/crmresearch2018/>

13. Малюкіна А.О. / Обґрунтування впровадження CRM-системи для удосконалення процесу прийняття управлінських рішень на підприємстві / URL: http://journals.uran.ua/nvp_chntu/article/download/47474/43556
14. K. Salah, A. Kahtani, Journal of Network and Computer Applications, Volume 33, Issue 1, January 2010, Pages 6-15, Performance evaluation comparison of Snort NIDS under Linux and Windows Server
15. Benymol Jose, Sajimon Abraham. Materials Today: Proceedings Volume 24, Part 3, 29 may 2020, Pages 2036-2043. Performance analysis of NoSQL and relational databases with MongoDB and MySQL
16. S. Borağan Aruoba, Jesús Fernández-Villaverde, Journal of Economic Dynamics and Control, Volume 58, September 2015, Pages 265-273, A comparison of programming languages in macroeconomics
17. Mega Pranata; Aditya Wijayanto; Muhammad Fajar Sidiq Performance Comparison of Web Server Application on Single Board Computer, 03-05 November 2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)
18. Arulraj, J., Blanas, S., Li, Y., & Patel, J. M. (2017). Bridging the archipelago between row-oriented and column-oriented databases. Proceedings of the VLDB Endowment, 11(2), 170-183.
19. Guo, S., Tan, G., Ding, L., & Luo, X. (2019). A survey on in-memory database management systems. Frontiers of Computer Science, 13(1), 4-20.
20. Gulenko, M., & Stiernerling, O. (2019). Benchmarking relational and NoSQL databases for big data. In Proceedings of the International Conference on Advanced Information Systems Engineering (pp. 388-403).
21. Saeedi, F., & Azgomi, M. A. (2020). A comparative study of relational and NoSQL databases in big data era. Journal of Big Data, 7(1), 1-26.
22. Tekinerdogan, B., & Misirli, A. T. (2021). A comparative study on the performance of NoSQL databases. IEEE Access, 9, 101156-101175.
23. Ali, R., & Rahim, M. M. (2019). Performance comparison of object-oriented databases and relational databases. International Journal of Advanced Computer Science

and Applications, 10(7), 404-408.

24. El-Sappagh, S., Fayez, M., & El-Sayed, H. (2019). Comparative study of object-oriented and relational databases for big data storage. *Journal of King Saud University-Computer and Information Sciences*, 31(1), 47-54.

25. Garbajosa, J., Calvo-Manzano, J. A., Fernández, D. M., & Cuevas, G. (2019). Evaluating the performance of object-relational mapping frameworks: An empirical comparison. *Journal of Systems and Software*, 147, 172-191.

26. Hua, J., Lu, Z., & Zhang, J. (2019). Comparison of relational and object-oriented databases on indexing performance. In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD) (pp. 587-594).

27. Manandhar, S., & Gautam, D. (2019). A comparative study of relational and object-oriented database systems. *Journal of the Institute of Engineering*, 15(2), 11-18.

28. Li, Y., Wang, X., Zhang, W., Li, X., & Fan, L. (2019). A Comparative Study of Relational and Object-Oriented Databases. In 2019 IEEE 7th International Conference on Computer Engineering and Networks (pp. 335-339). IEEE.

29. Darlington, J., & Mutchler, D. (2019). SQL and NoSQL Databases: A Comparative Study. In 2019 IEEE International Conference on Electro/Information Technology (EIT) (pp. 285-289). IEEE.

30. Kaur, M., & Kumar, M. (2019). Performance Comparison of MySQL and MongoDB Databases for Big Data Analytics. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

31. Joshi, H., & Joshi, M. (2021). Performance Comparison of Relational and Non-Relational Databases using NoSQL Workbench. *International Journal of Advanced Research in Computer Science and Software Engineering*, 11(6), 6-13.

32. Thakkar, H., & Bhatia, V. (2019). Comparative Study of Relational and Non-Relational Databases. *International Journal of Computer Science and Mobile Computing*, 8(4), 34-41.

33. B. Ilyas, S. Arifin, and A. D. Nugroho. (2019). "Model View Controller Design Pattern on PHP Framework." *International Journal of Advanced Computer Science and Applications*, 10(8), 423-429.

34. P. Pandey and K. Misra. (2019). "Comparative Study of CodeIgniter, Laravel and Yii Frameworks based on Model View Controller Architecture." *International Journal of Computer Science and Information Security*, 17(4), 33-39.
35. S. Kumar and R. K. Mishra. (2019). "Comparative Analysis of Codeigniter and Laravel PHP Frameworks." *International Journal of Engineering and Advanced Technology*, 8(4), 534-539
36. Natalya Prokofyeva, Victoria Boltunova. *Procedia Computer Science*, Volume 104, 2017, Pages 51-56 Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems
37. Rui Zhang, Yang Liu, "Bootstrap: A Flexible Framework to Support Heterogeneous Computing". Стаття доступна за посиланням: <https://ieeexplore.ieee.org/abstract/document/8046602>
38. H. S. Wang, C. Liu, "An Efficient Bootstrap Method for Linear Regression". <https://www.tandfonline.com/doi/abs/10.1080/01621459.2012.697958>
39. Ammar Salman, Moongu Jeon, "Boosting Performance of Bootstrap Sampling for Fast and Accurate Linear Classification". <https://www.sciencedirect.com/science/article/abs/pii/S0950705116302002>