

## АНОТАЦІЯ

**Казмірчук В.М. Розробка автоматизованої інформаційної системи впорядкування та обробки даних з використанням мови програмування PHP та IntelliJ IDEA для BuInvest. – Рукопис.**

Кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю 121 – Інженерія програмного забезпечення. – Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПм-61 // м.Тернопіль, 2020 // С. , рис. – , табл. – , додат. – , бібліогр. –

Метою кваліфікаційної роботи є питання повного переведення, інструментами сучасних інформаційних технологій, діяльності в віртуальне середовище, в якому питання роботи з великими об'ємами даних та інформаційного пошуку виходить на перший план.

Практичне значення кваліфікаційної роботи – робота виконана на замовлення компанії «BuInvest» з метою переведення діяльності на дистанційний режим роботи в умовах не прогнозованого карантину.

Технології розробки – мова гіпертекстової розмітки html, каскадна таблиця стилів css, мова програмування PHP, платформа Java Enterprise Edition, середовище розробки IntelliJ IDEA, база даних MySql.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ДИСТАНЦІЙНА РОБОТА, АЛГОРИТМ, СУБД, HTML, CSS, JEE.

## ABSTRACT

**Kazmirchyk V.M. Development of an automated information system for organizing and processing data using the programming language PHP and IntelliJ IDEA for BuInvest. – Manuscript.**

The master degree thesis for the qualification level of magistr in the specialty 121 — Software Engineering. – Ternopil Ivan Pul'ui National Technical University, Faculty of Computer Information Systems and Software Engineering, Software Engineering Department, group SPm-61 // Ternopil, 2020 // Pages. – , pictures. – , tables. – , supp. – , bibl.ref. –

The purpose of the qualification work is the complete transfer of activities into a virtual environment using the tools of modern information technology. The activity involves working with large volumes of data and information search.

Practical purpose of qualifying work – the work was ordered by BuInvest Company. Purpose is transfer the activity to distant work in conditions of unpredictable quarantine.

Development technologies - html hypertext markup language, cascading style sheet CSS, PHP programming language, Java Enterprise Edition platform, IntelliJ IDEA development environment, MySql database.

**KEYWORDS: INFORMATION SYSTEM, DISTANT WORK, DBMS, HTML, CSS, JEE.**

## ЗМІСТ

АНОТАЦІЯ .....	3
ABSTRACT .....	4
ВСТУП .....	7
1 АНАЛІЗ СТАНУ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ЗА МАТЕРІАЛАМИ ВІТЧИЗНЯНИХ І ЗАКОРДОННИХ ПУБЛІКАЦІЙ .....	9
1.1 Постановка задачі .....	14
1.2 Пошук актантів та варіантів використання .....	18
1.3 Опис ключових варіантів використання .....	20
2 ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ .....	26
2.1 Проектування бази даних .....	26
2.1.1 Вибір бази даних .....	26
2.1.2 Опис сутностей бази даних .....	28
2.2 Опис архітектури системи .....	36
2.2.1 Обґрунтування технологій використаних для розробки .....	36
2.2.2 Опис програмної системи .....	44
2.2.3 Взаємодія з серверами баз даних .....	49
2.3 Опис інтерфейсу користувача .....	51
3 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ СИСТЕМИ .....	54
3.1 Налаштування середовища сервера .....	55
3.2 Розгортання веб-сайту .....	61

4ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	62
4.1Охорона праці [24].....	62
4.2Створення метеорологічних умов виробничого середовища користувачів ВДТ, ЕОМ, ПЕОМ .....	68
4.3Особливості роботи та розлади здоров'я користувачів комп'ютерів, що формується під впливом роботи за комп'ютером .....	70
ВИСНОВКИ.....	76
ДОДАТКИ.....	81

## ВСТУП

В силу обставин, ми стали сучасниками такого періоду існування людства, коли віддалений доступ до інформації – це, без перебільшення, можливість зберегти життя та здоров'я, отримати освіту, знайти роботу та працювати, можливість спілкуватися, вирішувати найрізноманітніші побутові питання, здійснювати покупки продуктів та товарів першої, тай не тільки, необхідності. І якщо раніше, згідно статистичних даних фонду «Суспільна думка», активність аудиторії в Україні у всесвітній мережі на добу складала близько 67%, то сьогодні реалії такі, що цей відсоток значно виріс.

Надзвичайно важливим, в наш час, питанням є можливість доступу до інформації, адже беззаперечна істина якою є твердження про те, що хто володіє інформацією, той володіє і світом, стала як ніколи актуальною. Однак тут не все так просто. Проблема в тому, що сучасний світ переповнений інформацією. Реальність сьогодення така, що людина з раннього дитинства занурюється в інформаційне середовище як за допомогою мережі інтернет, так і при використанні сучасних інформаційних, комп'ютерних і телекомунікаційних технологій. При чому, віковий ценз знизився вже до рівня, коли користуватись джерелом може будь-хто, не зважаючи навіть на вміння читати і писати, оскільки вже давно існує голосовий пошук. Все це стало невід'ємною частиною повсякденного життя, що дозволяє оперативно, через інформаційні канали, отримати довідкову, наукову або розважальну інформацію з будь-якого куточку світу. Інформація стала однією з основних складових життєвого простору, так само як і інтернет, як соціальна та комунікативна сфера, став невід'ємною

частиною повсякденного життя не лише фахівця, але і звичайного людини, без прив'язки до віку.

Таким чином, все більш актуального значення набуває, для сучасної людини, інформаційна культура, а проблема грамотного і оперативного пошуку, обробки, зберігання та передачі інформації стає все більш актуальною. Але не потрібно забувати, що в умовах жорстких обмежень проблеми виникають не лише у звичайних людей, але й у сферах, які забезпечують життєдіяльність суспільства не лише напрямую, а й тих хто сплачує податки, які дадуть нам змогу пережити цей складний час. Отже, проблема повного переведення діяльності в віртуальне середовище, в якому питання роботи з великими об'ємами даних та інформаційного пошуку виходить на перший план, а власне пошук інформації в безмежному інформаційному полі є одним з основних процесів, який дасть змогу тим, хто швидко зреагує, зберегти всій бізнес та рівень життя.

Для реалізації системи опрацювання даних запропоновано комплексний підхід з використанням, при врахуванні специфіки діяльності, сучасних інформаційних технологій, технічних та програмно-математичних засобів вирішення класу означених проблем, стратегій, механізмів і методик.

## **1 АНАЛІЗ СТАНУ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ЗА МАТЕРІАЛАМИ ВІТЧИЗНЯНИХ І ЗАКОРДОННИХ ПУБЛІКАЦІЙ**

Інформаційними технологіями можна вважати методи, виробничі процеси і програмно-технічні засоби, об'єднано в так званий технологічний ланцюг, за допомогою якого забезпечується збір, обробка, зберігання, розповсюдження (транспортування) та відображення інформації, необхідні з ціллю зменшення як трудомісткості процесів при використанні інформаційного ресурсу, так і покращення показників надійності та оперативності.

Метою даного конкретного дослідження є приведення діяльності конкретного підприємства до відповідності вимогам сучасних непередбачуваних по термінам обмежень, зокрема через створення інформаційної системи, яка надасть змогу перенести роботу конкретної організації на віддалену роботу без втрати по якісним та кількісним показникам активної діяльності [1].

За останні роки суттєво змінилось співвідношення промислового та цивільного будівництва. Якщо раніше, більше вартості будівництва приходилось на промислові об'єкти, то сьогодні, в силу обставин, домінує цивільне будівництво, через, зокрема, значне збільшення будівництва приватного житлового фонду. Ринок «самостійно розставив пріоритети» виявивши найбільш ефективний напрямок, притаманний конкретно нашому регіону і, як виявилось, це зовсім не об'єкти великої промисловості. В основу подальшого

розвитку регіону закладено стратегічну концепцію будівництва об'єктів соціальної сфери – лікарні, школи, дитячі садки, житловий фонд. Хоча є всі сподівання вірити в те, що криза мене і від того обрана предметна область лише виграє і розширить свої горизонти освоєння.

Таким чином область дослідження – головні стратегічні пріоритети соціально-економічного розвитку регіону в складних умовах сьогодення, зокрема об'єкти цивільного будівництва, як основного і найбільш економічно обґрунтованого, та такого, що найбільше відповідає тенденціям з погляду ефективності через підвищення економічних показників.

Об'єктом дослідження є діяльність конкретної будівельної компанії «BuInvest», яка задля високих показників

конкурентоспроможності змушена перейти на віддалену роботу та дистанційне спілкування як з фактичними замовниками так і з потенційними клієнтами. Зокрема автоматизація системи дозволить не просто не втратити попередні напрацювання, а й оптимізувати роботу як в самій компанії так і підвищити ефективність та якість роботи з замовниками та максимально спростити доступ клієнтів до потрібної їм інформації. Робота ж з базою даних дозволить віддалено їх зберігати та підвищити показники безпеки роботи з ними. В цілому система призначена на підвищення ефективності та зручності у користуванні.

На сьогодні існують подібні системи як для даної предметної області, так і для суміжних областей. Однак їх кількість досить невисока, та й загального поширення не отримали, у зв'язку зі специфікою діяльності та тенденцією до миттєвого старіння існуючих систем, пов'язаного з швидкого розвитку інформаційних технологій. Специфікою ж нашого регіону, по обраній області дослідження, вважаю складності у веденні



малого бізнесу, в результаті чого керівники чи власники не зацікавлені у веденні діяльності відкрито через високі податки, довготривалий процес державної компенсації та невелику кількість замовлень, що стосується соціальної сфери.

Однак з розвитком цього виду діяльності виникає необхідність в автоматизації та допомозі стороннього сучаснішого програмного забезпечення, яке, для того, щоб його можна було вважати актуальним, було б розроблене з врахуванням не лише специфіки діяльності конкретного підприємства, але й останнім «словом» сучасних інформаційних технологій, методик, інструментів для створення програмного забезпечення [6, 7].

На сьогодні виявлено наступні питання, щодо організації роботи компанії:

1. Відсутність ефективних засобів зберігання інформації.
2. Складна обробка та пошук інформації.
3. Висока імовірність помилок.
4. Відсутня оперативна інформація та можливість оперативного обміну даними.
5. Не систематизованість даних.
6. Трудомісткість реєстрації.

З причини існування таких проблеми потрібно реалізувати засоби, які б могли забезпечити їх раціональне вирішення та допомогти процесу виконання основних завдань. Зрозуміло, що всі засоби в рамках однієї роботи реалізувати можливості немає, і це може бути предметом наступних реалізацій чи модернізацій.

З огляду на вище сказане, планується розробка системи, яка б автоматизувала прийом та обробку, оперування інформацією про замовлення в системі роботи з замовленнями. Дані повинні зберігати цілісність та дозволяти ефективно використовувати ресурси для пошуку та організації. Також варто спроектувати можливість додання нових об'єктів, приміщень (житлових нежитлових), встановлення статусу робіт (початок будівництва, завершене будівництво, здача в експлуатацію), додавання відгуків і побажань, додавання нових вакансій, видалення існуючих вакансій, замовити дзвінок оператора інформаційного центру.

Варто передбачити максимальну взаємодію всіх користувачів системи, та забезпечити актуальність даних, які надходять, опрацьовуються та зберігаються. Так як система розрахована на декількох основних користувачів, варто забезпечити механізм розподілення обов'язків користувачів та відповідно механізми роботи системи.

Необхідно виділити основних користувачів системи: ними стануть співробітники будівельної компанії, які будуть займатися доданням нових будинків (житлових і нежитлових приміщень), встановлення статусу вже існуючим будинкам (менеджер по роботі з клієнтами), користувач системи який має можливість переглядати список замовлених дзвінків та встановлювати їм статус (оператор call центру), та користувач, який зможе переглядати інформацію про будинки та ціни житлових та нежитлових приміщень, список вакансій, буде мати можливість заказати дзвінок та додати відгук чи побажання про компанію (клієнт).

1. Менеджер по роботі з клієнтами – працівник організації, що веде роботу з клієнтами послуг та адмініструє систему інформування клієнтів. До його повноважень належать: авторизація, додавання будинків, додавання (житлових, нежитлових приміщень), додавання нових вакансій та видалення вже існуючих.

2. Оператор інформаційного центру – працівник компанії, до функцій якого належать: перегляд списку замолених дзвінків компанії та встановлення статусу виконання

Так як дана система є інформаційною системою [8], існують фактори, що стосуються обміну та зберігання інформації:

1. Як зберігається інформація;
2. Яка структура даних;
3. Як часто оновлюються дані;
4. Як відбувається обмін інформацією між користувачами.

Основна причина проектування бази даних і власне самої системи – забезпечення цілісності даних, уникання можливої втрати даних, полегшення роботи користувачів, автоматизація обробки, збереження даних, додаткові функції для забезпечення цілісності.

Система повинна виконувати основні функції, які автоматизують компанію та полегшать умови праці користувачів організації. До основних функцій предметної області варто віднести:

- 1) Авторизація клієнтів компанії;
- 2) Додати будинок;
- 3) Додати приміщення;
- 4) Додати вакансію;
- 5) Корегувати інформацію про будинки, приміщення та вакансії.

б) Замовити дзвінок компанії;

7) Залишити відгук чи побажання; Основні поняття та їх атрибути:

1) Користувачі – менеджер по роботі з клієнтами, оператор інформаційного центру та клієнт;

2) Замовлення дзвінка з компанії (дата, електронна пошта, номер телефону, питання);

3) Встановлення відгуків та побажань компанії (дата, прізвище, ім'я, електронна пошта, відгук чи побажання)

Більш автоматизована система прийому замовлень дозволить скоротити час витрачений на інформування клієнтів, спростити доступ клієнтів до потрібної їм інформації. Робота з Базою даних дозволяє віддалено зберігати дані, більш захищено користуватися цими даними. Система спрямована на підвищення ефективності та зручності у користуванні.

## **1.1 Постановка задачі**

Проаналізувавши обрану предметну область, було виявлено основні недоліки та проблеми, що існують на даний час в галузі. Тому постає завдання, як і якими методами можливо вирішити ці питання.

Перше запитання, яке виникає перед розробником – навіщо розробляти цю систему, що нового вона в несе в можливості існуючої вже системи, які переваги та користь буде від розроблюваного програмного забезпечення. Саме тому, я виділив основне завдання, яке

повинна виконувати система – вести облік інформації та зберігати дані в загальну базу даних. Отже, виокремлюється завдання вибору бази даних для збереження цих даних.

Далі виникають і інші запитання, а саме яким чином будуть отримуватися дані про користувачів. Також потрібно точно та однозначно сформулювати всі вхідні та вихідні дані, які буде характеризувати система, поля та атрибути, необхідні для внесення в базу даних.

Наступне запитання – як спроектувати базу даних таким чином, щоб вона була зручною та оптимальною, ефективною та корисною для безпосереднього використання при роботі з інформацією.

Виникає завдання з точки зору користувацького інтерфейсу – які основні елементи та моделі варто використати для зручності та ефективності в користуванні даним ПЗ для кінцевого користувача. Адже саме цього прагне розробник, будь-якої автоматизованої інформаційної системи, і це не виключення – бути дружнім до користувача та підвищувати показники ефективності роботи в цілому.

Звісно перелік можна продовжувати, адже границь досконалості не існує, однак потрібно виділити основні завдання, необхідні для вирішення:

1. Визначити та обґрунтувати тип бази даних та тип з'єднання.
2. Визначити оптимальну структуру даних для представлення інформації в базі.
3. Виділити функції користувача. Користувацький інтерфейс.

4. Визначити можливості Представлення інформації, зокрема: сортування, пошуку в даних тощо.
5. Вирішити завдання формування звіту по роботі.
6. Встановити можливості автоматизації системи.
7. Здійснити розподіл відповідальностей.

Для вирішення поставлених завдань, спочатку я визначив основні сутності системи. Відповідно вони будуть характеризуватися даними про:

- 1) будинок;
- 2) житлові приміщення;
- 3) нежитлові приміщення;
- 4) замовленні дзвінки;
- 5) наявні вакансії;
- 6) відгуки та побажання.

Враховуючи перелічені вище завдання, я пропоную використати наступні функції за сторони:

- менеджера по роботі з клієнтами:

- 1) Додати будинок;
- 2) Додати житлові, нежитлові приміщення;
- 3) Встановлення статусу виконаних робіт;
- 4) Функції редагування будинків та приміщень;
- 5) Додати вакансію;
- 6) Функції редагування та видалення вакансій.

- оператора інформаційного центру:

- 1) Перегляд запрошених дзвінків; 2) Встановлення статусу виконаним.

- клієнта:

- 1) Перегляд будинків (житлових нежитлових приміщень);
- 2) Перегляд вакансій;
- 3) Заовлення дзвінка;
- 4) Доавання відгуків і побажань.

Основні дані, необхідні для бази даних:

- 1) про працівників компанії: логін і пароль;
- 2) про будинок: назва, місце розташування, статус виконання робіт;
- 3) про житловий будинок: номер квартири, загальна площа, житлова площа, ціна;
- 4) про комерційні приміщення: площа і ціна;
- 5) про заовлення дзвінка компанії: П.І.П., номер телефону, e-mail, запитання, час поступлення заовлення, статус обробки дзвінка;
- 6) про відгуки та побажання: нікнейм клієнта, e-mail, текст відгуку або побажання, дата.

Задля зручності в користуванні та при існуванні великої кількості записів необхідно використати функції сортування даних.

Всі функції будуть виконуватися користувачами з використанням інтерфейсу. Основні запити та методи обробки інформації будуть виконуватися автоматично і приховуватися під користувацьким інтерфейсом. Таким чином будуть проводитися перевірки, необхідні для якісного функціонування програмного забезпечення: перевірки на валідність записів, перевірки користувачів, обмеження та розмежування функцій різних користувачів.

## 1.2 Пошук актантів та варіантів використання

Заплановано, що розроблювана система планується для використання віддалено на сервері з метою надання консультацій клієнтам будівельної компанії з використанням сучасних інформаційних технологій через глобальну мережу інтернет. Згідно вимог розроблюваною системою я назву готовий веб-сайт, який дасть змогу переглянути потрібну користувачам інформацію. Цільовою аудиторією системи (користувачами) плануються працівники будівельної компанії, а також її клієнтів. Таким чином в системі передбачено присутність наступних актантів, а саме:

- 1) Менеджер по роботі з клієнтами; 2)
- Оператор інформаційного центру; 3)
- Клієнт.

Актанти системи представлені на рисунку 1.1.





Менеджер по роботі з клієнтами



Оператор інформаційного центру




Клієнт

Рисунок 1.1 – Аналіз актантів системи.

Проаналізувавши дану предметну область я тримав наступні актанти. Короткий опис функцій актантів системи я писав та представив в таблиці 1.1.

Таблиця 1.1. Аналіз актантів системи

<i>Актор</i>	<i>Короткий опис</i>	<i>Актор</i>
Менеджер по роботі з клієнтами	Розміщує дані про будинки (житлові нежитлові приміщення), встановлює статус робіт даного будинку. Додає нові вакансії в компанії, видаляє неактуальні вакансії.	
Оператор інформаційного центру	Має можливість переглядати список замовлених дзвінків клієнтами. Встановлює статус їх виконання.	
Клієнт		

Таким чином, аналізуючи представлення, я зробив висновок про те, що відповідно до варіантів використання системи, основних користувачів можна розподілити на два типи: працівники компанії (менеджер по роботі з клієнтами, оператор інформаційного центру) та клієнти компанії.

### 1.3 Опис ключових варіантів використання

Варіант використання – це зв'язний блок функціональності, яку надає класифікатор (система, підсистема або клас) [9]. Цей блок описує послідовність повідомлень, якими обмінюється система та один або кілька зовнішніх користувачів (актантів), а також дії, що здійснюються при цьому системою.

Варіант використання служить для визначення певної частини поведінки класифікатора (яким можна також вважати підсистему чи навіть систему в цілому), без вказівки на його внутрішню структуру. Кожен варіант використання описує певну послугу, яку надає своїм користувачам класифікатор. Іншими словами, це певна можливість використання класифікатора, який «видимий» зі сторони. Варіант використання описує всю послідовність повідомлень, що починає користувач (і моделі – актант), у термінах взаємодії між користувачем та класифікатором, включаючи відповіді класифікатора. До взаємодії, що стосується лише комунікації між системою та актантами. Внутрішня поведінка та реалізація приховані. Вся множина варіантів використання якого-небудь класифікатора або системи розділяється і повністю описує його поведінку. Кожен варіант використання представляє собою певну розумну долю функціональності, яка доступна для користувачів. Варта також пам'ятати, що під терміном «користувач» слід розуміти не лише людей, але й комп'ютери, а також інші об'єкти. Актант представляє собою таку собі ідеалізацію намірів користувача, не самого цього користувача. Один реальний користувач може відповідати декільком актантам, а один актант може представляти один і той самий намір відразу кількох користувачів.

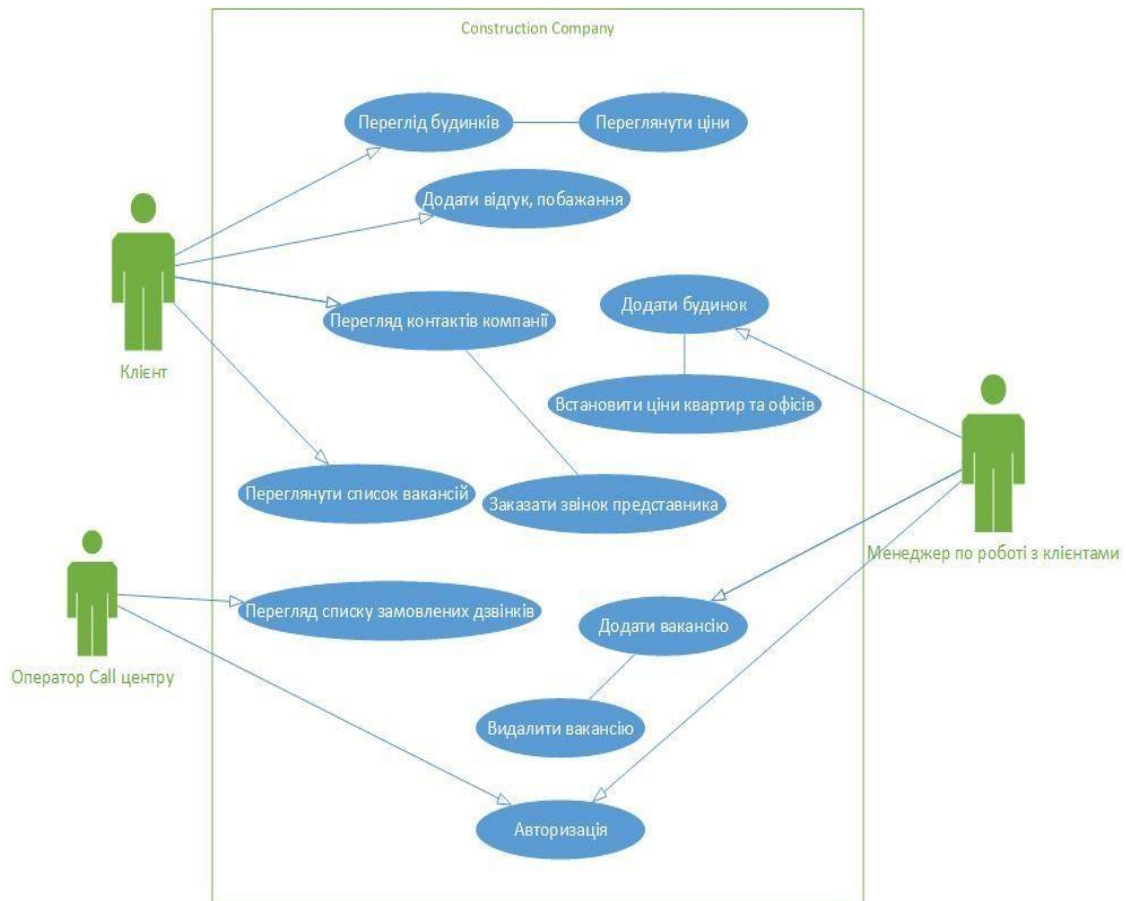


Рисунок 1.2. – Діаграма варіантів використання

Для розмежування функцій було використано такий варіант використання як «Авторизація». Цей варіант використання відповідає за систематизованість функцій залежно від типу користувача. Для клієнтів компанії щоб переглянути інформацію про будинки і наявні вакансії авторизуватись не потрібно. Під час «Авторизації» система просить в працівників компанії їх ім'я та пароль.



Рисунок 1.3 – Діаграма варіантів використання для менеджера

Варіант використання «Додати будинок» дозволяє менеджеру по роботі з клієнтами розміщувати на веб-сайті будинки компанії. Під час додавання будинку вказується і ряд інших його параметрів такі як: назва будинку, розташування (місто, вулиця, номер будинку), тип будинку (житлове або комерційне приміщення), статус будівництва. У варіанті використання «Встановлення цін квартир та офісів» залежно від типу приміщення. Якщо воно житлове приміщення вказується: номер квартири, загальна площа, житлова площа і ціна. Або нежитлове приміщення тоді вказується загальна площа та ціна. Якщо будинок добудовується то менеджер по роботі з клієнтами може встановити ціни квартир та офісів. До всіх даних реалізована функція редагування даних. Діаграма варіантів використання для менеджера по роботі з клієнтами представлена на рисунку 1.3.

Наступний варіант використання «Додати вакансію». Даний варіант використання дозволяє менеджерів по роботі з клієнтами додавати вакансії які на даний момент затребуванні в компанії і видаляти неактуальні. Діаграма варіантів використання представлена на рисунку 1.3.



Рисунок 1.4 – Діаграма варіантів використання для оператора інформаційного центру

Варіант використання «Перегляд списку замовлених дзвінків» дозволяє оператору інформаційного центру переглядати запит на дзвінків від клієнтів компанії. Та встановлювати статус виконаних. Діаграма варіантів використання представлена на рисунку 1.4.



Рисунок 1.5 – Діаграма варіантів використання для клієнта

«Перегляд будинків» варіант використання який служить інформуванням клієнта компанії про наявні будинки і стан виконаних робіт.

Наступний варіант використання «Додати відгук або побажання» створений для відображення реальних проблем в будівельній компанії. Де попередні клієнти можуть поділитися зі своїми наступними відгуками і побажаннями для самої компанії. Для цього потрібно буде заповнити інформацію про клієнта, таку як: нікнейм, email також сам текст відгуку або побажання.

Варіант використання «Перегляд контактів компанії» служить для швидкого доступу клієнтів до контактів компанії. Також буде реалізована функція замовлення дзвінка оператора інформаційного центру. Щоб замовити дзвінок потрібно вести наступні дані: прізвище, ім'я, по – батькові, номер телефону, e-mail, запитання.

Діаграма варіантів використання представлена на рисунку 1.5.

## 2 ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Проектування бази даних

#### 2.1.1 Вибір бази даних

В результаті аналізу було прийняте рішення про те, що оптимальним для проектованої системи буде можливість вибору БД(бази даних) [10] по типу MySQL – вільної системи призначеної для керування базами даних за реляційним типом. Прийняття в розробку системи управління БД було як таку, однією з особливостей якої є відкритий код, та яку було створено як альтернативну безлічі комерційних систем. Вата зазначити, що СУБД від початку була надто схожа до MySQL, однак, вона розвивалась, все більше розширювалась і сьогодні є однією із найпоширеніших систем керування базами даних. Перш за все тенденції її використання говорять про те, що її використовують в продуктах, в яких передбачено певну динаміку, так званих динамічних веб-сторінках. Це зумовлено тим, що базі цього типу притаманне хороше поєднання та підтримка різноманітними сучасними мовами програмування. По друге – це компактність багатопоточного сервера БД. По третє характеристики високої швидкодії, стійкості та простоти використання. СУБД прийнято вважати не просто хорошим, а оптимальним рішенням саме для додатків (продуктів) малих та середніх. Перевагою можна назвати й те, що вихідний код серверу можна компілювати при використанні великої кількості та різних платформ. Найбільше можливості серверу виявляються для UNIX – систем, в яких присутня підтримка



багатопотоковості, що дає можливість підвищити продуктивність системи в цілому. Не можу не зазначити і можливість безкоштовного використання MySQL. Отже перевагу серверу MySQL я надав із-за наступних показників:

- 1) простоти при встановленні і використанні;
- 2) підтримці необмеженою кількістю користувачів, які можуть одночасно працювати з базою;
- 3) великої кількості рядків в таблицях, яка може сягати до 50млн.;
- 4) високої швидкодії при виконанні команд;
- 5) наявності простої та з достатньою ефективністю системи безпеки.

Таким чином, без перебільшення, можна сказати про те, що MySQL є одним із стандартів в області систем управління базами даних для веб за стосунків. А сьогодні тенденції такі, що на базі неї розширюються можливості використання для будь – яких бізнес – додатків, які можна назвати «критичними». Таким чином воне не поступається таким системам управління БД виробники яких вже давно вважаються золотим стандартом, зокрема це Oracle, також IBM, і обов'язково потрібно згадати про Microsoft та Sybase.

Основні переваги MySQL:

- 1) багатопотоковість, можливість підтримки декількох запитів одночасно;
- 2) можливість оптимізації зв'язків з варіантами приєднання в один прохід багатьох даних;

- 3) фіксована та змінна довжина записів;
- 4) гнучкість системи привілеїв та паролів;
- 5) гнучкість підтримки формату чисел; змінна довжина рядків та часових міток;
- 6) масштабованість та швидкодія роботи;
- 7) сумісність із ANSI SQL;
- 8) ODBC драйвер;
- 9) безкоштовність;
- 10) підтримка зі сторони провайдерів для послуг хостинга;
- 11) застосування механізму InnoDB для швидкої підтримки транзакцій.

### **2.1.2 Опис сутностей бази даних**

Проаналізувавши обрану предметну область, було виявлено декілька сутностей, які відповідають за представлення інформації в базі даних. Так ними стали:

- 1) відомості про працівників компанії: логін і пароль;
- 2) відомості про будинок: назва, місце розташування, статус виконання робіт;
- 3) відомості про житловий будинок: номер квартири, загальна площа, житлова площа, ціна;
- 4) відомості про комерційні приміщення: площа і ціна;
- 5) відомості про замовлення дзвінка компанії: П.І.П., номер телефону, e-mail, запитання, час поступання замовлення, статус обробки дзвінка;

б) відомості про відгуки та побажання: нікнейм клієнта, email, текст відгуку або побажання, дата.

Розроблювана база даних містить такі основні таблиці предметної області: authorization, order\_call, house, apartament, commercial\_premises, reviews\_and\_wishes.

При розробці баз даних, між таблицями використовують наступні зв'язки:

1) один – до – одного – одній сутності однозначно відповідає інша сутність.

2) один – до – багатьох (багато – до – одного) – одній сутності (екземпляру сутності) відповідає декілька інших сутностей(екземплярів сутностей);

3) багато – до – багатьох – даний тип зв'язку реалізовувати недоцільно, оскільки виникне неоднозначність, що приведе до пошкодження цілісності опису предметної області. Такий тип зв'язку на практиці реалізують через два зв'язки: багато – до – одного та один – до – багатьох.

4) На рисунку 2.1 представлено зв'язки між таблицями розроблювальної системи.

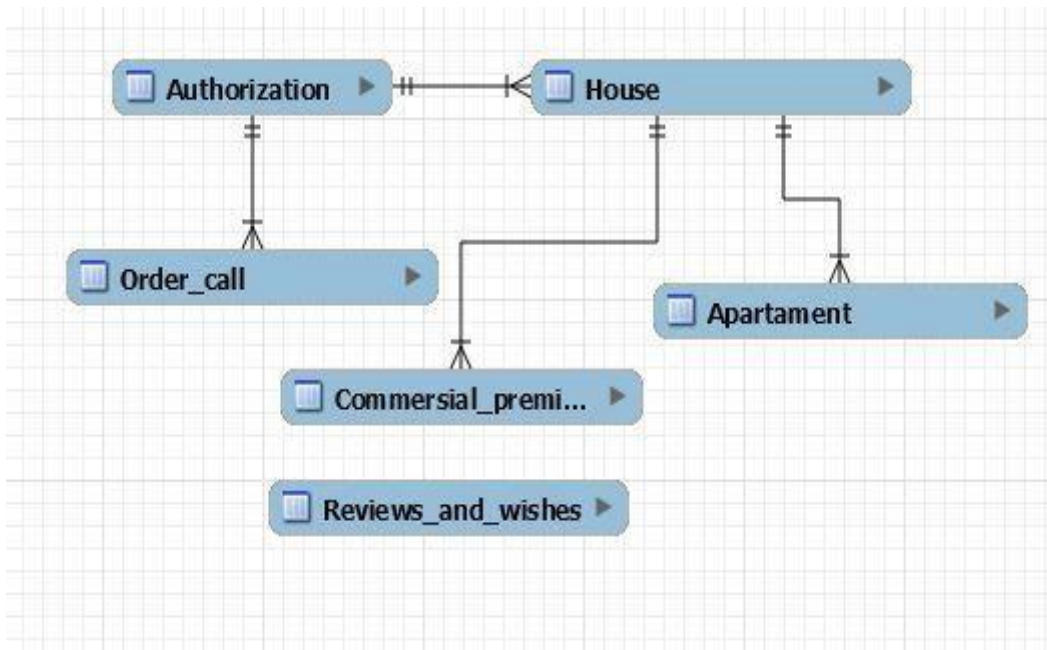


Рисунок 2.1 – ER діаграма бази даних

Відповідно до представлення на рисунку 2.1 між таблицями предметної області створено наступні зв'язки:

- 1) Authorization – один до багатьох, здійснений за допомогою таблиці order\_call;
- 2) Authorization – один до багатьох, здійснений за допомогою таблиці house;
- 3) House – один до багатьох, здійснений за допомогою таблиці apartment;
- 4) House – один до багатьох, здійснений за допомогою таблиці commercial\_premises;
- 5) Reviews\_and\_wishes – дана сутність не зв'язана зв'язками тому, що клієнтам компанії не потрібно авторизуватись щоб надіслати відгук або побажання компанії. Для цього їм необхідно вести достовірні дані про особу.

Для зв'язку між таблицями використовується зовнішній ключ. Зовнішній (вторинний) ключ — це одне або кілька полів (стовпців) у таблиці, що містять посилання на поле або поля первинного ключа в іншій таблиці. Зовнішній ключ визначає спосіб об'єднання таблиць [11].

Дві логічно пов'язані таблиці одну називають таблицею первинного ключа або головною таблицею, а іншу таблицею вторинного (зовнішнього) ключа або підпорядкованою таблицею.

Таблиця «authorization» використовується для авторизації працівників компанії і показана на рисунку 2.2.

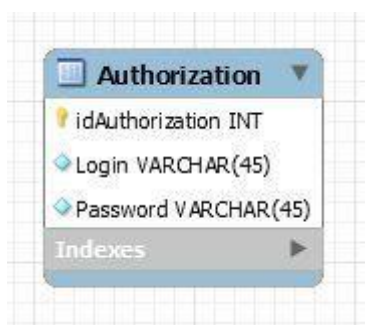


Рисунок 2.2 – Таблиця «authorization»

Таблиця авторизації містить такі поля: idAuthorization – тип INT, первинний ключ таблиці, генерується автоматично; Login – тип VARCHAR(45), логін працівника компанії для входу в систему; Password – тип VARCHAR(45), пароль для входу в систему.

Таблиця «authorization» використовується для авторизації працівників компанії і показана на рисунку 2.3.

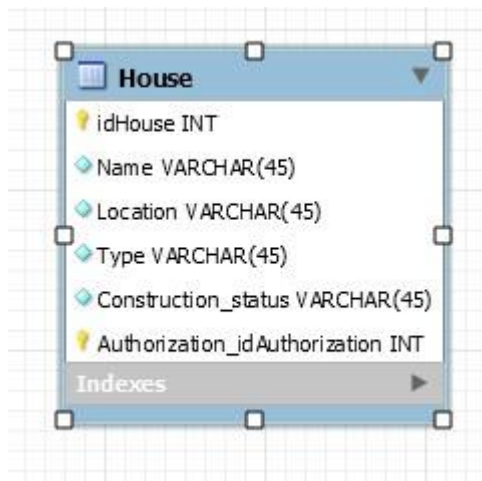


Рисунок 2.3 – Таблица «house»

Таблица «будинку» містить наступні поля: idHouse – тип INT, первинний ключ таблиці, генерується автоматично; name – тип VARCHAR(45), назва будинку; Location – тип VARCHAR(45), адреса розміщення будинку, Type – тип VARCHAR(45), тип будинку (житловий комерційний); Construction\_status – тип VARCHAR(45), статус виконання робіт; authorization\_idAuthorization – тип INT, вторинний ключ таблиці, вказує на працівника компанії.

Таблица «apartament» використовується для авторизації працівників компанії і показана на рисунку 2.4.

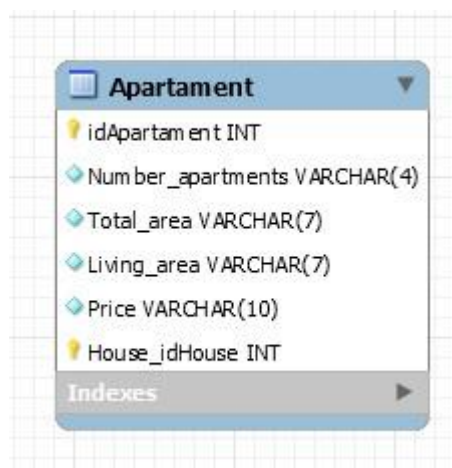


Рисунок 2.4 – Таблица «apartament»

Таблиця житлові приміщення містить такі поля: idApartament – тип INT, первинний ключ таблиці, генерується автоматично; Number\_apartments – тип VARCHAR(4), номер кімнати в будинку; Total\_area – тип VARCHAR(7), загальна площа квартири; Living\_area – тип VARCHAR(7), житлова площа квартири; Price – тип VARCHAR(10), ціна квартири; House\_id House – тип INT, вторинний ключ таблиці, вказує на будинок.

Таблиця «commercial\_premises» використовується для авторизації працівників компанії і показана на рисунку 2.5.



Рисунок 2.5 – Таблиця «commercial\_premises»

Таблиця «нежитлові приміщення» містить наступні поля: – idCommercial\_premises тип INT, первинний ключ таблиці, генерується автоматично; Area – тип VARCHAR(6), містить площу приміщення; Price – тип VARCHAR(8), містить ціну приміщення; House\_id House – тип INT, вторинний ключ таблиці, вказує на будинок.

Таблиця «commercial\_premises» використовується для авторизації працівників компанії і показана на рисунку 2.6.

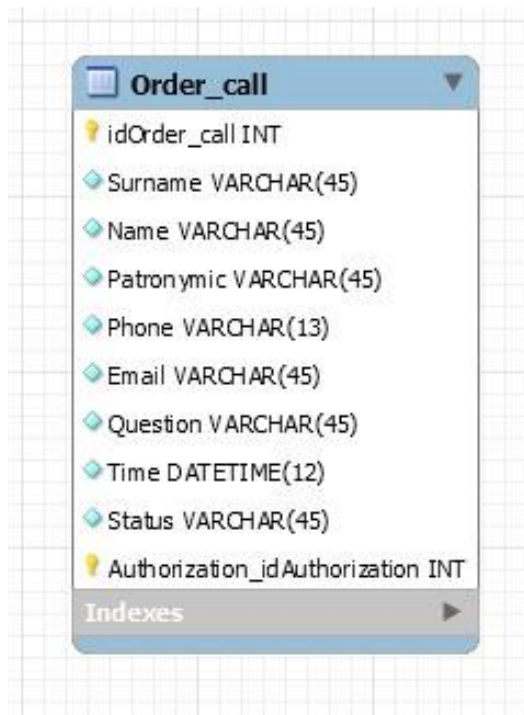


Рисунок 2.6 – Таблиця «order\_call»

Таблиця «замовити дзвінок» містить наступні поля про клієнта: idOrder\_call тип INT, первинний ключ таблиці, генерується автоматично; Surname – тип VARCHAR(45), прізвище; Name – тип VARCHAR(45), ім'я; Patronymic – тип VARCHAR(45), по – батькові; Phone – тип VARCHAR(13), номер телефону; Email – тип VARCHAR(45), адрес електронної пошти; Question – тип VARCHAR(45), питання до оператора call центру; Time – тип DATETIME(12), час подання заявки; Status – тип VARCHAR(45), статус опрацювання заявки; authorization\_idAuthorization – тип INT, вторинний ключ таблиці, вказує на працівника компанії.

Таблиця «reviews\_and\_wishes2 використовується для авторизації працівників компанії і показана на рисунку 2.7.



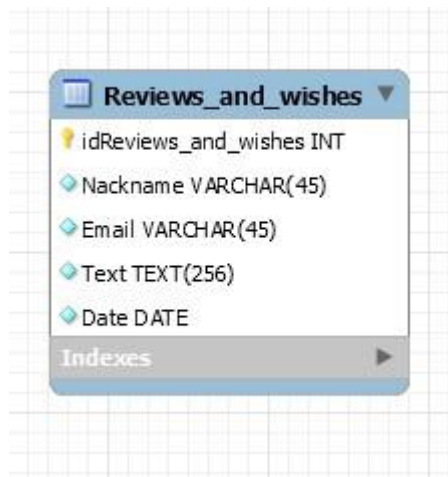


Рисунок 2.7 – Таблиця «reviews\_and\_wishes»

Таблиця «відгуки та побажання» містить такі поля: idReviews\_and\_wishes – тип INT, первинний ключ таблиці, генерується автоматично; Nickname – тип VARCHAR(45), нікнейм клієнта; Email – тип VARCHAR(45), адрес електронної пошти клієнта; Text – тип TEXT(256), текст відгуку або побажання; Date – тип Date, дата написання відгуку або побажання.

Проаналізувавши всі таблиці було отримали наступні сутності та зв'язки між ними. Повна діаграма бази даних представлена на рисунку 2.8.

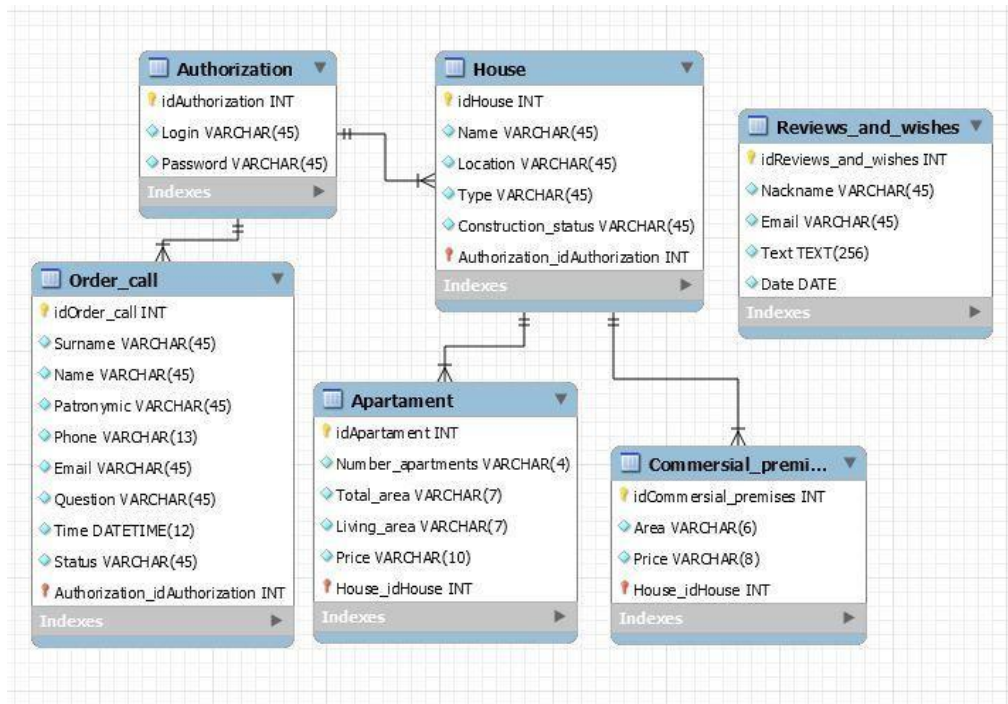


Рисунок 2.8 – ER діаграма бази даних в цілому

## 2.2 Опис архітектури системи

### 2.2.1 Обґрунтування технологій використаних для розробки

Варта зазначити, що присутня певна подібність до системи баз даних архітектури —клієнт-сервер. Така архітектура є цікавою й актуальною, і це зумовлено основною й тому, що дає змогу забезпечити достатньо просте та відносно не вартісне рішення, яке стосується питання про колективний доступ до бази даних локальної мережі.

Застосування архітектури по типу —клієнт – сервер може бути зацікавим та виправданим у тому випадку, якщо той хто приймає рішення відповідає за безпосередню розробку програмного продукту, коли йдеться про постійне звернення до даних з використанням локальної мережі чи до файлового серверу. До такого програмного забезпечення можливе звернення кількох користувачів. З часом інших

програм, які призначені для обробки цих даних певним чином, відповідно до вимог [12].

В загальному ж хотів би зазначити, що найбільш реального поширення архітектура за типом «клієнт-сервер» набула завдяки тенденції, яка обумовлена розвитком та широким впровадженням у практику певної концепції яка передбачає відкриті системи. Саме тому варта дуже коротко навести відомості про такі системи.

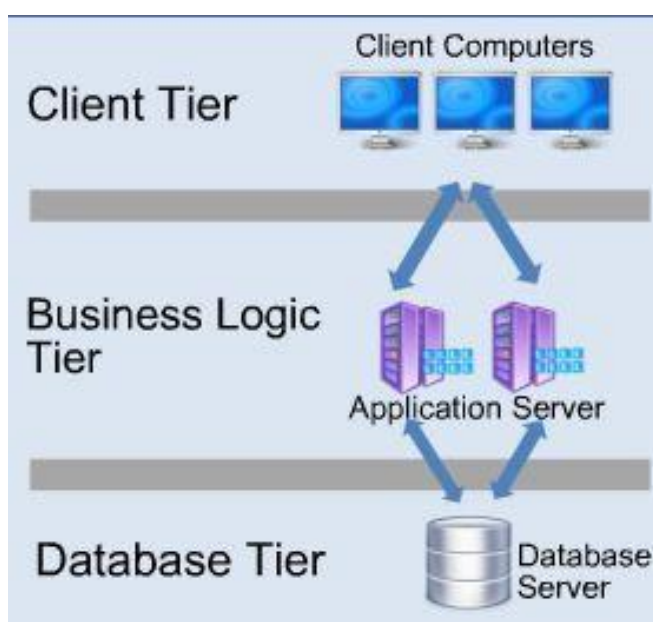


Рисунок 2.9 – Тьох рівнева архітектура типу «клієнт-сервер»

Якщо говорити про основний зміст такого підходу (системи відкритого типу), то це йдеться про можливість спрощення комплексування для обчислювальних систем. Цього можна досягти за рахунок як міжнародної так і національної стандартизацій як апаратних так і програмних інтерфейсів. Головною ж причиною для розвитку такої концепції (системи відкритого типу) є тенденція до переходу використання саме локальної комп'ютерної мережі (чи комп'ютерних

мереж), а також проблеми що можуть виникати при комплексуванні апаратно-програмних засобів, викликані цим переходом. Не дивлячись ні на що сьогодні завдяки бурхливому розвитку технологій для глобальних комунікацій системи відкритого типу здобувають все більшого й більшого значення та масштабності.

Ключовим моментом систем відкритого типу, спрямованим в сторону користувачів, може бути є незалежність, зокрема від певного конкретного постачальника. Якщо орієнтуватись на продукти тих компаній, які притримуються стандарту систем відкритого типу, споживач, який отримує будь-який продукт цих компаній, не потрапляє до неї в залежність. Це є важливим моментом, оскільки користувач має можливість продовжити нарощувати потужності своєї системи відкритого типу додаючи чи придбаючи продукти будь-якої іншої з компаній, яка дотримується стандартів. Це стосується і апаратних і програмних таких засобів.

Стандартизовану операційну систему можна вважати за практичну опору системних та прикладних таких програмних засобів для систем відкритого типу. На сьогодні такою системою можна вважати об'єднання операційних систем багатозадачного типу UNIX [13]. Постачальникам різноманітних варіантів таких операційних систем в результаті довготривалої роботи все таки вдалось дійти згоди щодо основних стандартів такої операційної системи. Всі найпоширеніші версії UNIX сьогодні в загальному є сумісними в частині інтерфейсів, запропонованих прикладним (та й для більшості випадків системним) програмістам. Таким чином, у найближчі роки, навіть не дивлячись на те, що існує система яка є претендентом на

стандарт, операційна система багато задачного типу UNIX залишається базою для систем відкритого типу.

Технології та стандарти систем відкритого типу дають змогу забезпечити реальну та перевірену на практиці можливість для створення системних та прикладних програмних засобів з такими властивостями як мобільність(portability) означає порівняну простоту для переносу програмної системи у широкому спектрі апаратнопрограмних засобів, які мають відповідати стандартам; інтеперабельність(interoperability) означає відносно спрощення комплексування призначене для нових програмних систем на базі використання готових компонентів з стандартними інтерфейсами.

Таким чином за перевагу для користувачів можна прийняти факт того, що їм доступна можливість поступової заміни компонентів системи на їх оновлені чи й нові версії, при цьому без втрати працездатності в цілому системи. Саме в цьому приховане вирішення проблеми з поступового нарощування як обчислювальних і інформаційних так і інших потужностей для комп'ютерної системи.

Очевидно, що, в загальному, щоб прикладна програма, яка виконується робочою станцією, могла «запросити послугу» в певного сервера, зазвичай виникає мінімальна потреба в інтерфейсному програмному рівні, яким підтримується такого роду взаємодія, оскільки досить неприродно вимагати, щоби прикладною програмою напряду застосовувалися засоби транспортного рівня для локальної мережі. Саме це і є з такої взаємодії та такі принципи і належать до основних в плану принципів системної архітектури по типу—клієнтсервер».

Таким чином систему розбивають на дві частини, які можуть реалізовуватись різними вузлами мережі, тобто здійснюється поділ на

клієнтську частину та серверну. Власне взаємодія з клієнтською частиною системи здійснюється прикладною програмою або кінцевим користувачем. В найпростішому з випадків це забезпечується над мережним інтерфейсом. При необхідності так звана частина «клієнт» системи формує звернення (мережею) до частини «сервер». Відзначу, що для більш сучасних систем звернення по мережі до частини «сервер» можливо й не надасться у випадку, якщо вона (система) зможе «вгадувати» потребу користувача, а дані, які здатні відповідати наступному запиту користувача втримуються і частиною «клієнт».

В цілому ж про interface самої частини «сервер» варта говорити, як про такий що є визначеним та фіксованим. Ось чому можливим є утворення саме інших «клієнт-частин» для існуючої вже системи.

Основною проблемою систем подібного типу, які засновані на клієнт-серверній архітектурі, є необхідність мобільності в якомога можна більш широкому класі для апаратних та програмних рішень, які можуть відповідати для систем відкритого типу. Проте, навіть якщо і обмежуватися так званими UNIX-орієнтовними такими локальними мережами, то у інших мережах застосовують як різну апаратуру так і протоколи для зв'язку. Спроби ж по створенню систем, які підтримують різні наявні такі протоколи, призводять як до їх перенавантаження так і втрат по функціональності.

Ще більш складним аспектом означеного питання є можливість використання різноманітних представлень даних в різних вузлах локальною мережею, яка, природно є неоднорідною. Відомо, що в для різних комп'ютерів притаманна наявність як різної адресації і представлення для чисел так і відмінність в кодуванні символів тощо. Це має найбільш істотне істотне значення саме для так званих

високорівневих серверів, які використовуються при телекомунікації, обчислюваннях та баз даних.

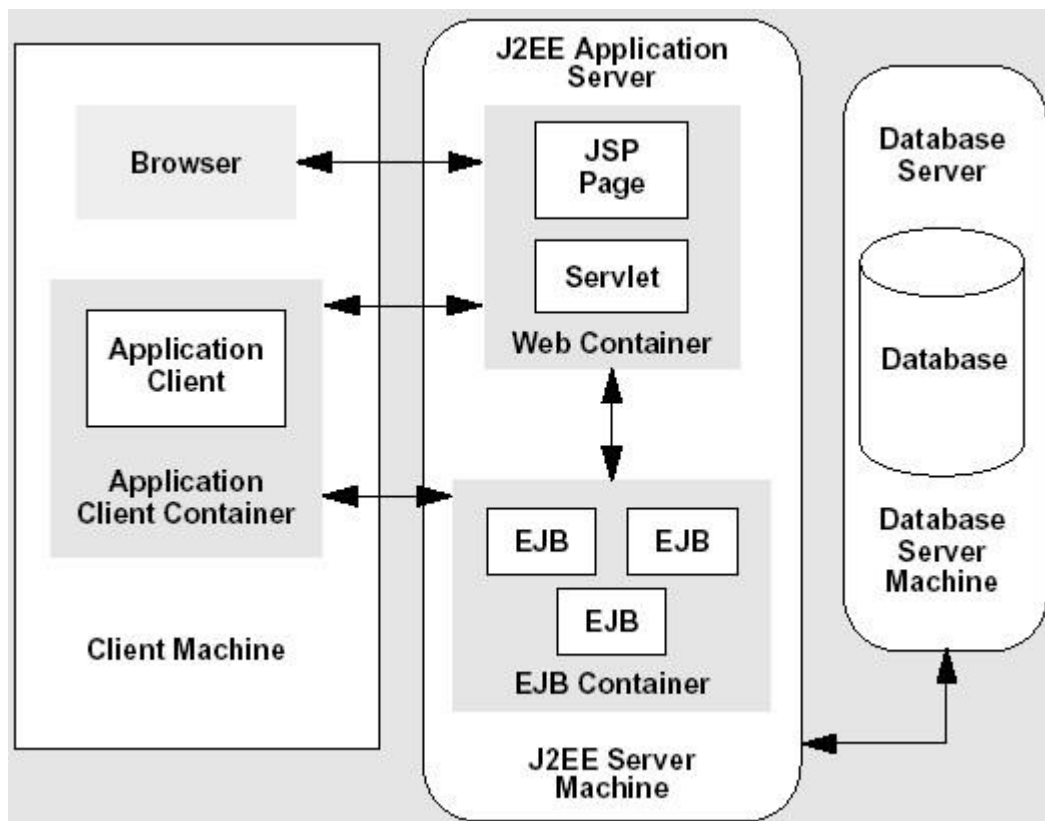


Рисунок 2.10 – Трьох рівнева клієнт-серверна архітектура

Як загальне рішення для проблем пов'язаних з мобільністю систем, які засновані на клієнт-серверній архітектурі, можна запропонувати скерування на такі «програмні пакети», які можуть реалізувати так звані протоколи «вилученого виклику процедур» (англ. Remote Procedure Call). Запит до сервісу в вилученому вузлі, якщо використовуються такі засоби, набуває вигляду звичайного виклику процедури. Засоби вилученого виклику процедур, у яких, і це природно, може утримуватися всі відомості стосовно специфіки апарату ратної складової коли мережа локальна мережі та протоколів

(мережевих), перетворює виклики в послідовності мережевої взаємодії. Таким чином, виявлено, що особливість мережевого середовища та протоколів, тобто їх специфіка, прихована безпосередньо від прикладного спеціаліста по програмуванню.

У випадку викликові вилучених процедур «вилученого виклику процедур» робляться видозміни форматів клієнтських даних у машинно-незалежні проміжні формати, а вже потім перетворюється в формати даних сервера. А вже під час передачі якихось певних параметрів, виконуються такі аналогічні переміни.

У випадку коли система реалізовується принципами стандартного пакета «вилученого виклику процедур», її можна з легкістю перенести у будь-яке з відкритих середовищ.

Основний принцип технології типу —клієнт-сервер‖ у стосунку до системи керування для баз даних варта звести до її поділу. Зручно такий поділ виконувати на дві частини. Перша - у вигляді додатку типу «клієнт» (або front-end), а друга як сервер баз даних (або backend). Ця така архітектура може сполучати все краще що має стосунок до обробки даних як на мейнфреймах, так і технології типу —файлсервер‖. В мейнфреймів є щось, що запозичене для технології —клієнтсервер‖, а саме такі риси, як:централізоване адміністрування та надійність при хорошій безпеці. А від технології типу —файл-сервер‖ успадковано низьку вартість та можливість розподілу обробки даних, при використанні ресурсів комп'ютерів-клієнтів.

Варта зазначити, що станом на сьогодні стандартне для систем запиту клієнта до сервера – це графічний користувацький інтерфейс. Окрім цього, клієнт-серверною архітектурою значно спрощується та прискорюється розробка аплікацій (додатків). Це відбувається в



рахунок того, що безпосередньо на сервері знаходяться правила перевірки даних на цілісність. Так некоректно працюючий клієнтський додаток не зможе призвести до втрати або ж «перекручування» даних. Все це раніше було властиве лише складним і дорогим системам. Сьогодні доступність сягла навіть невеликих організаціям. Вартість як устаткування так і програмного забезпечення заразом і обслуговування персональних комп'ютерів є у десятки разів нижчою ніж вона становить для мейнфреймів.

Компілятор Java може транслювати вихідні тексти Java-програм в код Java-машин. В загальному кажучи, Java-машина є віртуальна в тому розумінні, що її існування в вигляді реальних мікросхем та й інших пристроїв є неможливим. Вона представляє собою такий програмний емулятор, який реалізовується на якійсь-небудь так званій «традиційній» апаратній платформі. Імовірно, що вже найближчим часом можна сподіватись на появу та на все більш широке поширення прямих апаратно спрямованих реалізацій для Java машини [14, 15].

В основі проєкта Java лежить специфікація «ефективної ізольованої копії реальної машини» (віртуальна машина), що може бути як частина для цілого комплексу мір, які спрямовані на стандартизацію Java-середовища та на забезпечення його незалежності безпосередньо від апаратної та програмної платформ. Крім того, також варта враховувати й те можливо дещо специфічне середовище, в якому притаманно готуються та працюють Javaпрограми. Якщо ж Web-сторінка містить так звані Java-аплети, то вони можуть передаватися безпосередньо по мережі. Таким чином виходить, що перевага надається Java-коду, який би був якомога більш компактним. В протилежному в випадку час для завантаження сторінки може стати

надто великим. Архітектура і система команд для Java-машин проектувалися з врахуванням того, щоб власне досягти оптимальної компактності того коду. З іншого ж боку, формат команд Java-машини є достатньо простим, оскільки звичайно ж команди не мають операндів і мають розмір (займають) близько одного байта, тому це уможлиблює її (тієї машини) ефективність емуляції.

### 2.2.2 Опис програмної системи

Найвищим рівнем програмної системи є так званий web - рівень. Він відповідає за обробку користувацького вводу і повернення коректної відповідь назад користувачеві. Web рівень також повинен обробляти виключення, що генеруються іншими рівнями. Оскільки веб рівень відповідає за роботу з користувачем і аналізом надісланих користувачем даних, він повинен враховувати коректність введених користувачем даних і відповідати за авторизацією користувачів.

В основі веб рівня використана технологія Spring MVC [16, 17]. Модуль SpringMVC надає можливість розгорнутої підтримки для шаблону MVC, так само як для других засобів, які дають можливість спростити реалізацію рівня презентацій. Веб-фреймворк Spring MVC так само як і інші запити орієнтовані web MVC фреймворків побудований навколо центрального сервлету, який перехоплює усі запити і направляє їх до контролерів та надає інші функції, що полегшує розробку web-додатків. В Spring MVC клас DispatcherServlet є центральним сервлетом, який отримує запити і направляє їх відповідним контролерам. У додатку Spring MVC може існувати довільна кількість екземплярів DispatcherServlet.

DispatcherServlet, призначені для різних цілей (наприклад, для обробки запитів користувача інтерфейсу, запитів web-служб REST і т.д.). Кожен екземпляр DispatcherServlet має власну конфігурацію WebApplicationContext, яка визначає характеристики рівня сервлету, такі як контролери, що підтримують сервлет, відображення обробників, розпізнавання уявлень, інтернаціоналізація, оформлення темами, перевірка достовірності, перетворення типів і форматування.

На рисунку 2.11 представлено життєвий цикл обробки запиту в Spring MVC.

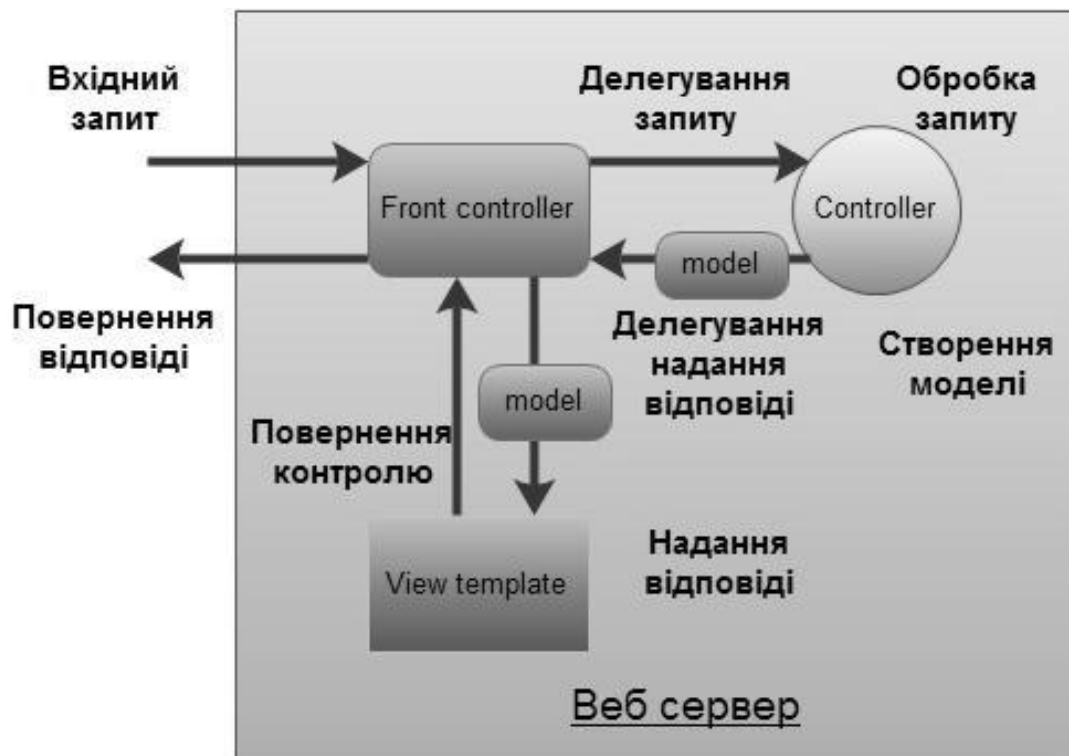


Рисунок 2.1 – Життєвий цикл обробки запиту Spring MVC

Життєвий цикл запиту Spring MVC включає такі етапи:

- Клієнт надсилає запит web-серверу у формі HTTP запиту.
- Цей запит перехоплює фронт-контролер (DispatcherServlet) і потім намагається знайти відповідний обробник відображень.

- За допомогою обробника відображень `DispatcherServlet` направляє запит до відповідного контролера.

- Контролер намагається обробити запит і повертає об'єкт моделі та представлення у вигляді екземпляра класу `ModelAndView` до фронт-контролера.

- Фронт-контролер надсилає запит, що містить назву представлення до `ViewResolver`, який повертає відповідне представлення, яке може бути JSP сторінкою, `Velocity` чи `Freemarker` шаблоном.

- Після цього фронтконтролер надсилає об'єкт моделі до об'єкта представлення, яке, після цього, надсилається клієнтові.

Web-рівень включає в себе контролери. Контролер - це клас завданням якого є обробка запитів клієнта і на основі введених користувачем даних і даних отриманих з рівня сервісів повернути користувачеві результат.

Контролер є частиною шаблону проектування MVC. Контролери отримують доступ бізнес-логіки web-застосунку за допомогою сервіс інтерфейсів. Контролери інтерпретують ввід користувача і трансформують його у вигляд моделі, яка буде повернена користувачеві у вигляді представлення. Поняття контролера в Spring дуже абстрактно, даючи можливість для створення широкого спектру різних видів контролерів, які можуть бути створеними.

В основі архітектури Spring контролерів є інтерфейс `org.springframework.web.servlet.mvc.Controller` [17]. Вихідний код цього інтерфейсу наведено в лістингу 2.1.

Лістинг - Код інтерфейсу Controller

```
public interface Controller {  
  
    /**  
     * Process the request and return a ModelAndView object which the  
     * DispatcherServlet  
     * will render.  
     */  
    ModelAndView handleRequest(  
        HttpServletRequest request,  
        HttpServletResponse response) throws Exception; }  
}
```

При побудові користувацького інтерфейсу використовують технологію `JavaServerPages` [18] особливістю якої є те, що за допомогою неї `web-розробники` можуть створювати такий контент, який характеризуватиметься не лише статичними але й динамічними компонентами. Сторінка `JavaServerPages` може містити текст двох типів, зокрема у вигляді статичних вихідних даних, які можна оформити в якомусь одному текстовому форматі, `HTML`, `SVG`, `WML`, чи `XML`, та у вигляді `JSP` елементів, якими конструюється безпосередньо динамічний контент. Також можна використовувати бібліотеки `JSPтегів`, так само як і `EL` – компактний синтаксис призначений для здійснення звернення до масивів, колекцій, об'єктів та їх властивостям всередині `JSP` [19]. `Expression Language` призначено для отримання можливості впроваджувати `Java-code` безпосередньо до статичного контенту `JavaServerPages` сторінок. Фрагмент `JSP` сторінки, який відповідає за виведення списку користувачів продемонстровано в лістингу 2.2

## Лістинг 2.2 - Фрагмент JSP сторінки

```
<c:forEach      items="{authorization}"      var="
authorization">
  <tr>
    <td>${ authorization.login}</td>
    <td>${user.password}</td>
    <td><a
href="/authorization/confirm">Підтвердити</a></td>
    <td><a
href="/authorization/reject">Відхилити</a></td>
  </tr>
</c:forEach>
```

В представленому лістингу продемонстровано використання компактного синтаксису призначеного для здійснення звернення до масивів, колекцій, об'єктів та їх властивостям всередині JSP – скриптової мови виразів , яка дозволяє отримати доступ до Java компонентам (JavaBeans) з JSP. Починаючи з версії JSP 2.0 її використовують зазвичай всередині JSP тегів в метою відокремлення Java code від JSP для забезпечення легкого доступу до Java компонентам. Розвиток EL відбувався з метою зробити його більш простим для дизайнерів, які мають мінімальні пізнання в мові програмування Java.

Для організації циклу в наведеному коді використовується тег forEach, який міститься в стандартній бібліотеці тегів. Стандартна бібліотека тегів JSP (JSTL) - розширення специфікації JSP , що додає бібліотеку JSP -тегів для загальних потреб , таких як розбір XML - даних, умовна обробка, створення циклів та підтримка інтернаціоналізації .

JSTL є альтернативою такому виду вбудованої в JSP логіки, як скріплет, тобто прямі вставки Java коду. Використання стандартизованих тегів краще, оскільки одержуваний код легше підтримувати і простіше відокремлювати бізнес-логіку від логіки відображення.

Для наповнення JSP сторінки даними використовується об'єкт ModelMap, який містить набір ключ-значення, в якості ключа використовується стрічка, в якості значення - будь-який Java об'єкт. Умовна обробка та створення циклів в середині JSP відбувається за допомогою стандартної бібліотеки тегів JSTL.

### 2.2.3 Взаємодія з серверами баз даних

JDBC (Java DataBase Connectivity) [20] - \_ набором класів та методів, що можуть використовуватися в такій мові програмування як Java, основна функція яких - робота з базами даних. JDBC є засобом, що дає можливість просто, універсально та при відмінній адаптації взаємодіяти з різними системами управління базами даних. Java DataBase Connectivity Interfaces було розроблено в корпорації Sun. Ці інтерфейси дають можливість забезпечити реалізацію усіх стандартизованих операцій при роботі із SQL базами даних. При чому розробник надає конкретної реалізації власне цих Interfaces (робиться вся взаємодія з БД. Сюди можна віднести як підключення та реєстрацію, так і виклики для збережених процедур тощо). Проектовані інтерфейсів здійснено так, що програма, в якій використовується Java DataBase Connectivity, має можливість приєднатися до будь-якої

сумісної з Java DataBase Connectivity баз, при чому, без модифікування кода. Однак все ж таки потрібно звернути увагу на певні деякі обставини, а саме:

1 – при використанні Java DataBase Connectivity не виконується лексичний аналіз чи перевірка синтаксису SQL на клієнтській стороні. Тут команди просто адресуються до бази даних незважаючи на їх коректність. Отже, якщо код SQL працюватиме в одній системі управління базами даних, проте не підходить іншій, реалізація не дасть реакції аж допоки не буде встановлено фактичне приєднання й переадресація SQL кода. Тоді ж Sim намагатиметься врегулювати це питання. Висока ймовірність того, що буде внесено корегування до наступних версій JDBC чи, можливо, при переході до критеріїв іншого стандарту.

2 – додаткові класи, які додаються до реалізації. Такі класи є спеціалізованими під певний конкретний продукт. Якщо використати такі певні класи, то швидше за все програма не працюватиме в інших сумісних з JDBC базах даних.

3 – однією з переваг драйверу JDBC є те, що це є драйвер так званого 4-го типу. Це значить, що при його написанні використано мову Java в її чистому вигляді. Саме це дозволить перенести його куди потрібно і надасть можливість використання для будь-якої платформи при підтримці TCP/IP, тому що такий драйвер приєднується лише за допомогою TCP/IP.

Отже, класи JDBC є основними компонентами для взаємодії програмного продукту та SQL. Для всіх основних класів JDBC, а саме: Connection, Statement, ResultSet, Blob й Clob характерне існування аналогів в SQL. Окрім цього до JDBC включено також і допоміжні



класи `ResultSetMetaData` та `DatabaseMetaData` основним призначенням яких є робота з мета даними. Зокрема, їх можуть використовувати з метою отримання інформації стосовно можливостей бази даних, також і з метою перевірки типу результатів запиту, в процесі налаштування та й просто у випадках коли інформація щодо робочих даних відсутня .

`JDBC Interface` вміщає класи по роботі з нестандартними розширеннями

### **2.3 Опис інтерфейсу користувача**

Головна сторінка сайту «Будівельної компанії» представлена на рисунку 2.12.



Рисунок 2.12 – Головна сторінка

На вкладці «Про компанію» розміщено головну сторінку сайту на якій знаходяться вгорі кнопки навігації по сайту. Внизу кнопок розміщений банер компанії. За банером знаходиться інформація про саму компанію. Також описуються основні цінності компанії.

На рисунку 2.13 представлено вкладку будинки на якій розміщені фотографії будинків представлених компанією.

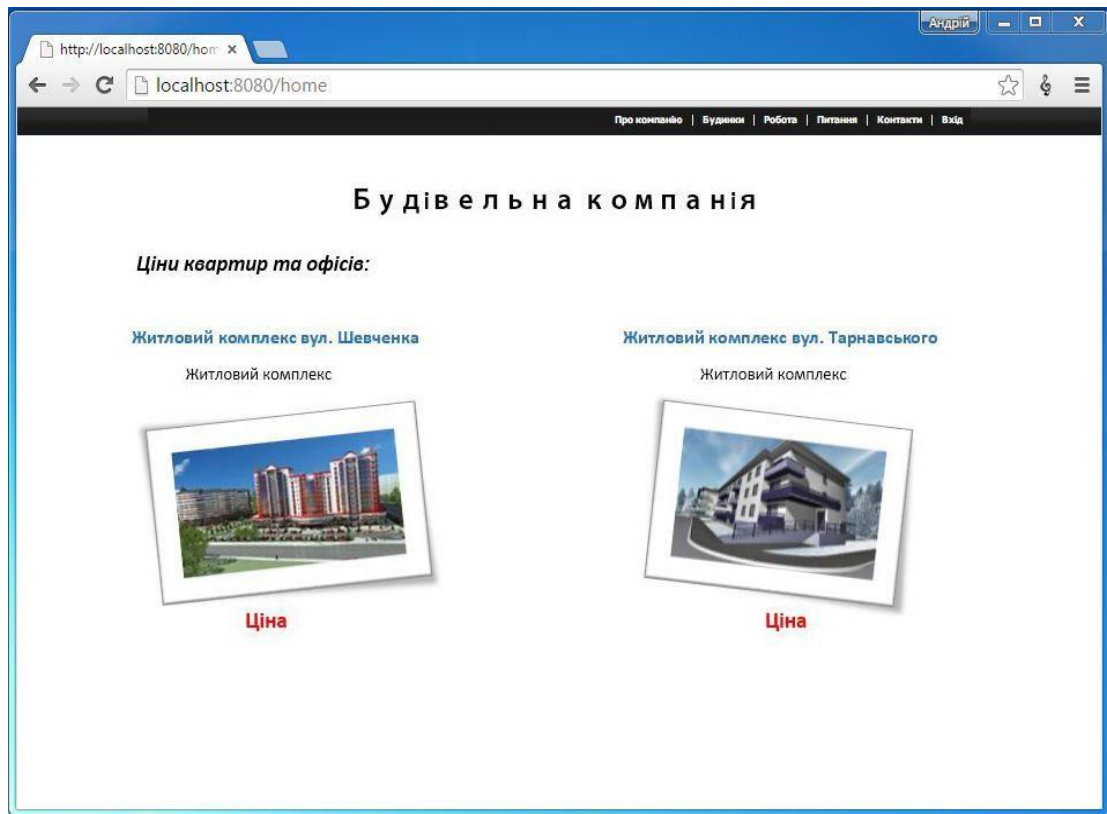


Рисунок 2.13 – Вкладка «Будинки»

Вкладка «Робота» представлена на рисунку 2.14. На дані вкладці клієнти компанії можуть дізнатись корисну та іктуальну інформацію, зокрема про вільні вакансії в будівельній компанії.

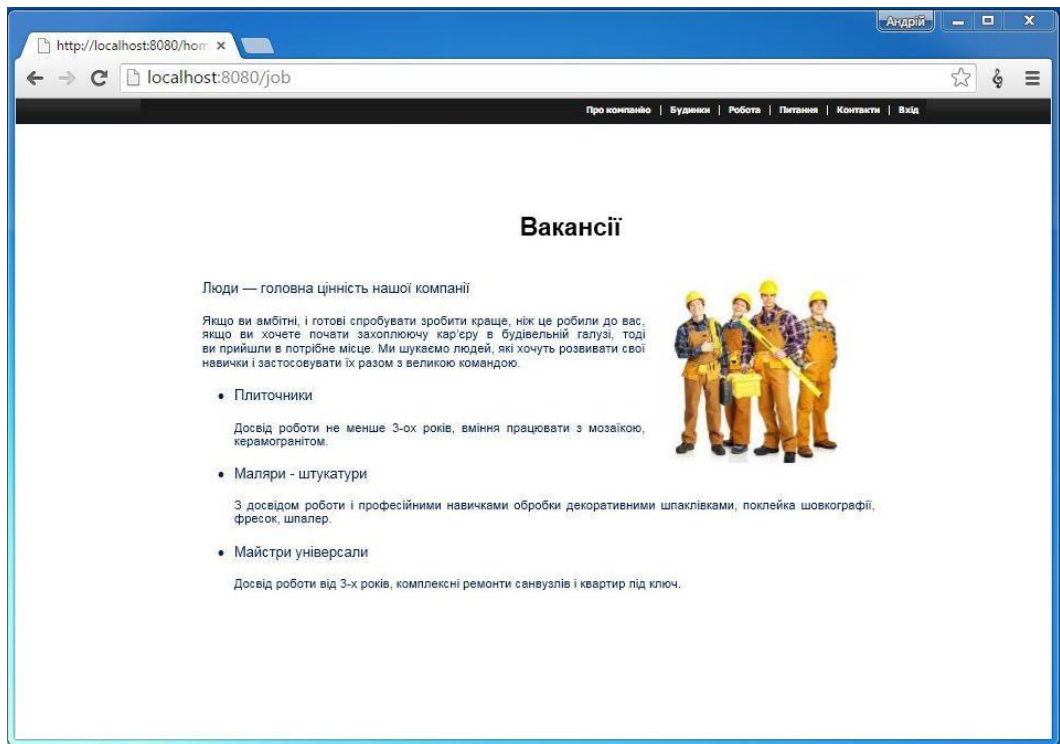


Рисунок 2.14 – Вкладка «Робота»

## 3 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ СИСТЕМИ

### 3.1 Налаштування середовища сервера

Перш ніж налаштувати сайт потрібно розгорнути веб – сервер. Налаштування сервера полягає у встановленні та налаштуванні JDK, веб-сервера Apache Tomcat і системи керування реляційними базами даних MySQL [21].

Веб-сайт написаний об'єктно-орієнтованою мовою Java, а отже для коректності його роботи потрібно встановити Java Runtime Environment (англ. JRE – середовище виконання для Java) – є мінімальною реалізацією для машини віртуального типу (Java Virtual Machine), необхідною для виконання додатків Java, без застосування компіляторів чи інших засобів для розробки. Його складовими є віртуальна машина Java Virtual Machine та бібліотека класів Java.

Java Virtual Machine – є основною частиною виконуючої Javасистеми JRE (англ. Java Runtime Environment або JRE). Вона (ця система) реалізовує байт-код Java, попередньо створений з початкового тексту Java програми одноіменним Java компілятором. Так само Java Virtual Machine може застосовуватись з метою реалізації програм, які пишуться іншими мовами програмування.

Java Virtual Machine зазвичай включають «Інтерпретатор байткоду», однак, з метою підвищення продуктивності в багатьох машинах також застосовується JIT-компіляція часто виконуваних фрагментів байт-коду в машинний код.

JRE поширюється вільно і для більшості платформ може бути завантажена з сайту Oracle.

Після завершення завантаження і запуску інсталятора JRE [22] запускається діалогове вікно, в якому слід натиснути унопку Install (див. рисунок 3.1)



Рисунок 3.1 – Вікно інсталяції JRE

Після натиснення кнопки Install з'являється вікно, що містить стан процесу інсталяції (див. рисунок 3.2).



Рисунок 3.2 – Вікно стану інсталяції

В кінці інсталяції з'являється вікно, що повідомляє про успішне завершення інсталяції (див. рисунок 3.3).



Рисунок 3.3 – Вікно успішного завершення інсталяції

Для розгортання веб-сайту потрібно встановити веб-сервер Apache Tomcat [23]. Після запуску інсталятора завантаженого з офіційного сайту Apache відображається вікно зображене на рисунку 3.4.



Рисунок 3.4 – Вікно інсталяції Apache Tomcat

В наступному вікні користувач повинен погодитися з ліцензійними умовами (див. рисунок 3.5).



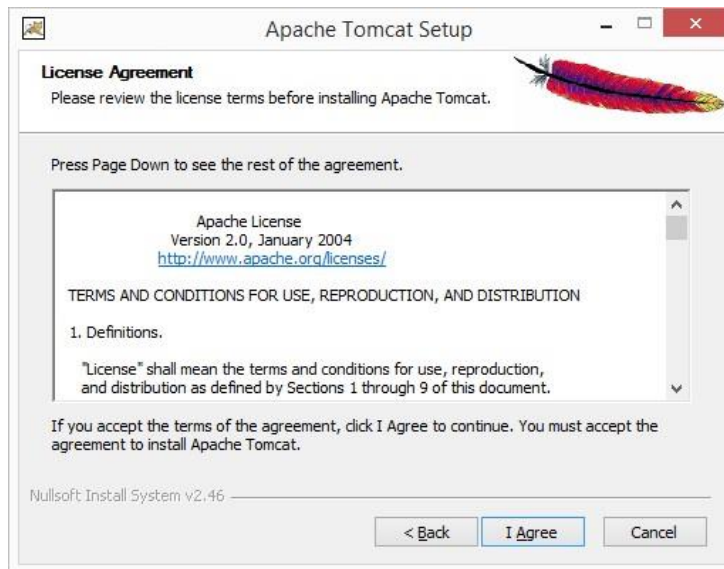


Рисунок 3.5 – Ліцензійні умови

В наступному вікні користувач вибирає компоненти, які потрібно встановити (див. рисунок 3.6).

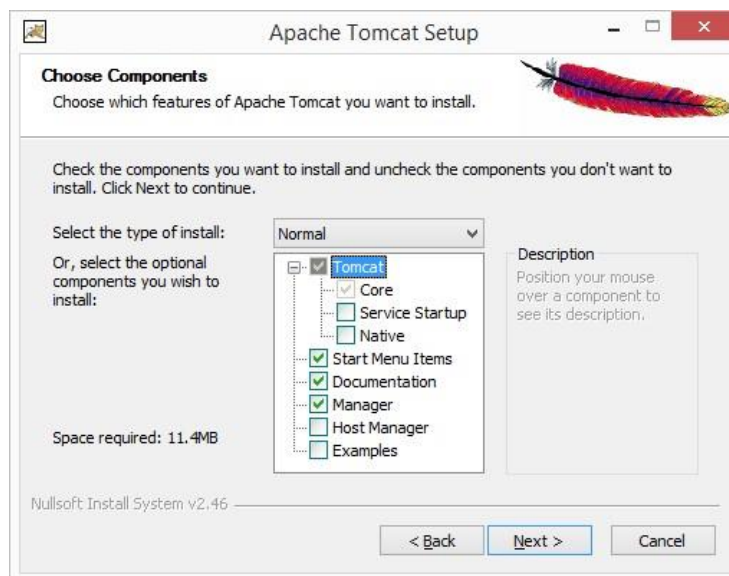


Рисунок 3.6 – Вибір компонентів для встановлення

У вікні продемонстрованому на рисунку 3.7 користувач повинен вказати папку із встановленим середовищем виконання Java (JRE).

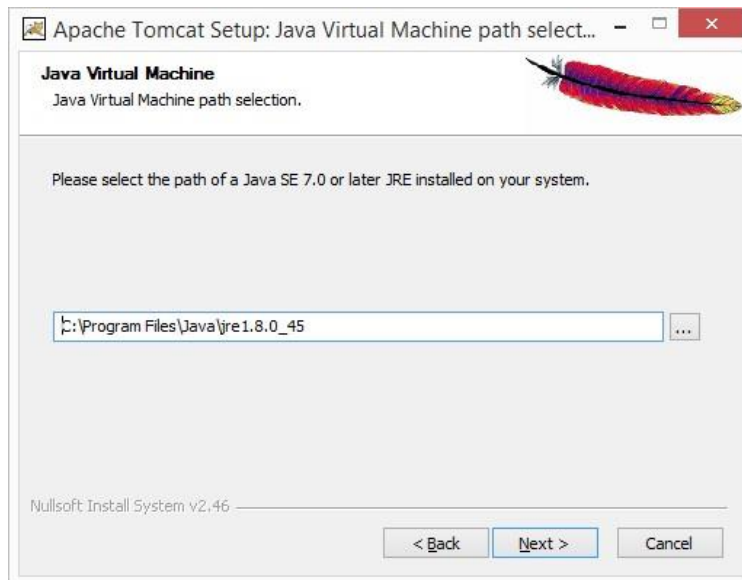


Рисунок 3.7 – Вибір папки із JRE

В наступному вікні необхідно вказати папку, в яку буде інстальовано веб-сервер Apache Tomcat (див. рисунок 3.8).

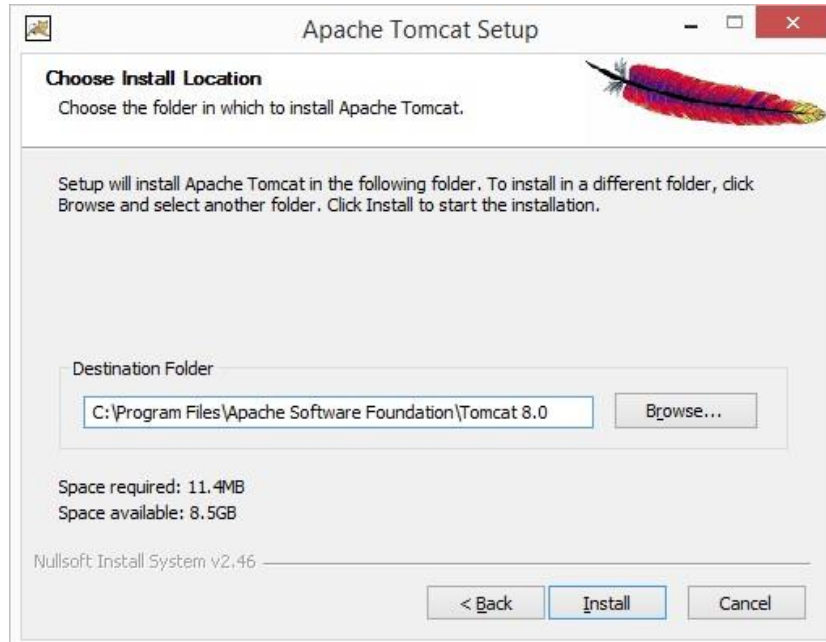


Рисунок 3.8 – Вибір папки для інсталяції веб-сервера

Далі потрібно встановити СКБД MySQL. Після встановлення треба відкрити Workbench та створити нову схему з назвою `tour_operator` (див. рисунок 3.9).

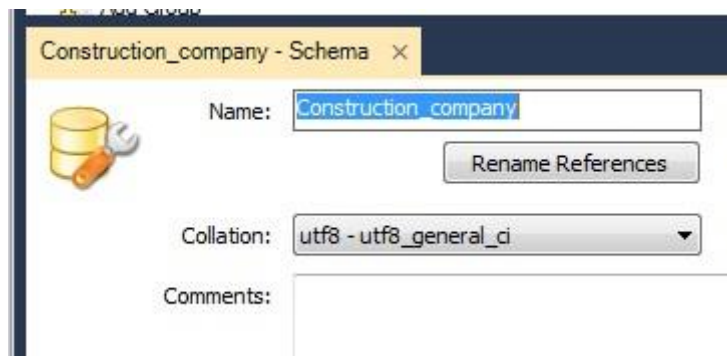


Рисунок 3.9 – Створення схеми

Після створення схеми потрібно вибрати в меню пункт `Server Data Import`. І у вікні `Data Import` вибрати файл `Construction_company.sql` для створення схеми бази даних і наповнення її даними.

### 3.2 Розгортання веб-сайту

Розгортання веб-сайту за допомогою веб-сервера Apache Tomcat можна здійснити двома способами:

- 1) Скопіювати файл `ConstructionCompany.war` у папку `$CATALINA_BASE/webapps/`, де `$CATALINA_BASE` – папка у яку було встановлено веб-сервер Apache Tomcat;
- 2) За допомогою веб-інтерфейсу Apache Tomcat (див. рисунок 3.1)
- 3) Для розгортання веб-сайту за допомогою веб-

інтерфейсу Apache Tomcat потрібно вибрати war файл, після чого натиснути кнопку Deploy.

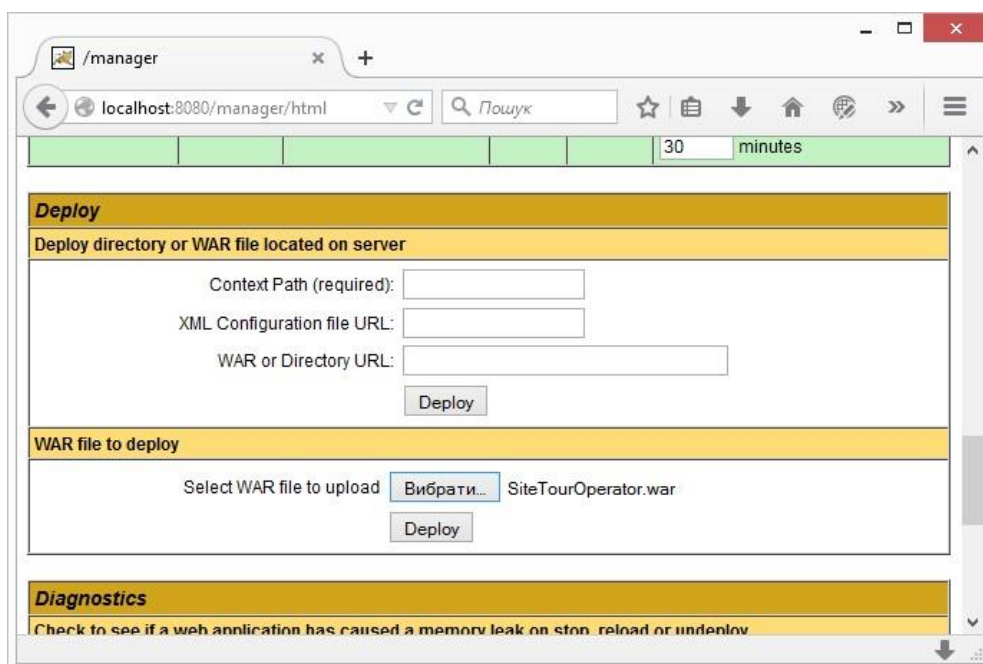


Рисунок 3.10 – Веб-інтерфейс Apache Tomcat

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1 Охорона праці [24]

Тема магістерської роботи пов'язана із розробкою програмного продукту – автоматизованої інформаційної системи обробки даних. Приміщення, в якому проводилася розробка, маючи площу 30 м<sup>2</sup>, містить 4 комп'ютеризовані робочі місця. Для зберігання документів використовуються дві шафи. Приміщення містить два вікна та одні двері, які розташовані по середині, між робочими місцями. Детальний план приміщення наведено на рисунку 4.1.

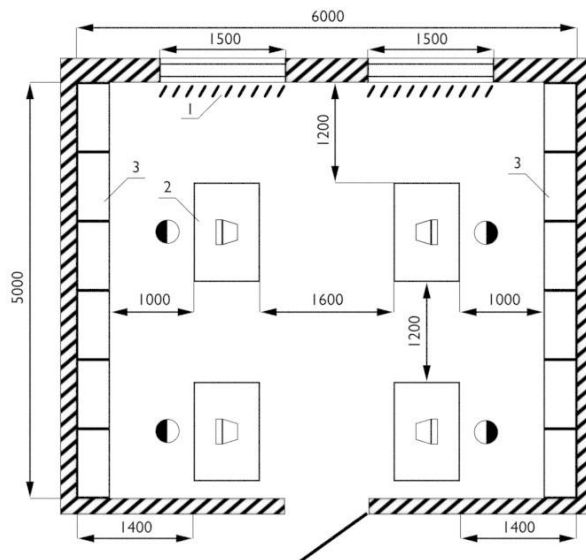


Рисунок 4.1 – План приміщення

Умовні позначки на рисунку: 1 – комп'ютеризоване робоче місце з ВДТ; 2 – сонцезахисні жалюзі; 3 – шафи для зберігання.

Розглянемо відповідність характеристик комп'ютеризованого робочого місця розробника нормативним [25]. Площа, на якій розташовується одне робоче місце з ВДТ, повинна становити не менше 6,0 кв м. В нашому випадку площа становить 7,5 кв м.

При розміщенні робочих столів з персональними комп'ютерами слід дотримувати:

- відстань між бічними поверхнями персональних комп'ютерів 1,2 м;
- відстань від тильної поверхні одного персонального комп'ютера до екрана іншого – 2,5 м;
- робочі місця з ВДТ розміщуються на відстані не менше 1 м від стіни зі світловими прорізами (вікнами);
- прохід між рядами робочих місць має бути не меншим за 1 м.

В даному випадку робочі місця розміщено так, що в поле зору робітника не може попасти екранна сторона дисплею, окрім його

власного. Також відстань між бічними поверхнями, відстань від стін із світловими прорізами та проходи між рядами (рис. 4.1) задовольняють нормам.

Рекомендовані розміри столу для робочого місця з ВДТ становлять: висота – 725 мм, ширина – 600-1400 мм, глибина 800-1000 мм. В даному приміщенні використовуються робочі столи з розмірами: ширина – 1200 мм, глибина – 800 мм.

Комп'ютеризовані робочі місця розміщені рядами вздовж стіни з вікнами. Це дає змогу виключити дзеркальне відбиття на екрані ВДТ джерел природного світла (вікон) та потрапляння останніх у поле зору операторів, що погіршує їх зорову роботу.

Відповідно до ДСН 3.3.6.042-99 роботи, що виконуються користувачами ЕОМ, відносяться до легких фізичних робіт – категорії Іа. У виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату.

Згідно НПАОП 0.00-1.28-10 приміщення, що розглядається, повинне мати природне і штучне освітлення.

Природне освітлення приміщення відбувається за системою одnobічного бічного освітлення. Природне світло проникає у приміщення через два світлові віконні отвори, які мають регульовальні пристрої для відкривання. Також наявні штори (жалюзі) з можливістю захисту працюючих від прямого попадання сонячних променів і регулювання рівня освітленості в приміщенні. Вікна приміщення орієнтовані на північний схід. Оскільки будинок розташований у відносній віддаленості від прилеглих будівель, то які-небудь перешкоди природному освітленню розглянутого приміщення відсутні. Всередині приміщення стіни обклеєні світлими шпалерами, стеля

побілена, у якості підлогового покриття використаний лінолеум світло-жовтого кольору.

В досліджуваному приміщенні використовується система загального рівномірного штучного освітлення. Для освітлення використані люмінесцентні лампи типу ЛД 80, які розміщено в три ряди (рисунок 4.2).

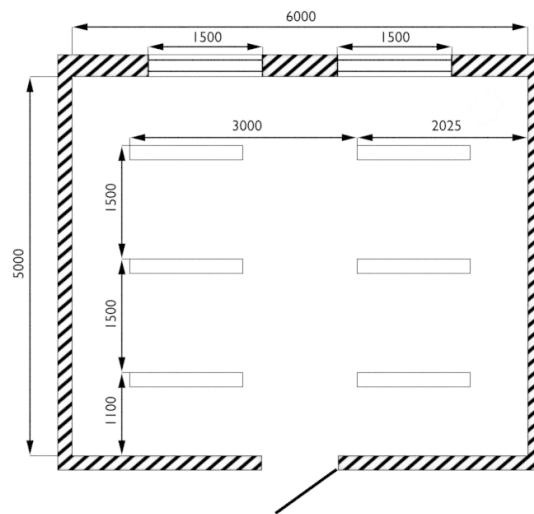


Рисунок 4.2 – Розміщення світильників у приміщенні

Визначимо фактичну освітленість в приміщенні з формули світлового потоку:

$$E_{\phi} = \frac{\Phi_{\lambda} \cdot \eta \cdot N}{S \cdot K_3 \cdot Z} \quad (4.1)$$

де  $N$  – число світильників у приміщенні,  $N \in \mathbb{N}$ ;

$\eta$  – коефіцієнт використання світлового потоку;

$\Phi_{\lambda}$  – світловий потік лампи, для лампи типу ЛД 80

$\Phi_{\text{л}} \approx 4300 \text{лк}$ ;

$K_z$  – коефіцієнт запасу,  $K_z \approx 1.5$ ;

$Z$  – коефіцієнт нерівномірності,  $Z \approx 1.1$  для люмінесцентних ламп;

$S$  – площа приміщення.

Підвісна стеля білого кольору має коефіцієнт відбиття  $\rho_{\text{стели}} \approx 70\%$ , стін  $\rho_{\text{стін}} \approx 50\%$ . Розрахуємо параметр  $i$ , який необхідний для визначення коефіцієнта освітленості.

$$i \approx a \cdot b \cdot 5 \cdot 6 \cdot 0.85 \cdot h \cdot (a \cdot b) \cdot 3.2 \cdot (5 \cdot 6)$$

Таким чином, знаючи коефіцієнти відбиття та параметр  $i$ ,  $\approx 0.42$ . Тоді, за формулою (4.1):

$$E_{\phi} \approx \frac{4300 \cdot 0.42 \cdot 6}{30 \cdot 1.5 \cdot 1.1} \approx 218 \text{лк.}$$

Нормативне значення освітленості для кабінету, де працюють з використанням ВДТ, згідно ДБН В.2.5-28-2006 становить 200 лк, допустимі межі відхилення  $180 \text{лк} \leq E_H \leq 220 \text{лк}$ . Отже, фактичне значення освітленості потрапляє в допустимі 10% відхилення від нормативного показника, тобто – відповідає нормам.

У приміщенні маються внутрішні джерела постійного шуму – вентилятори блоків ЕОМ, дисководи. Зовнішніми джерелами шуму і вібрації в приміщенні є проїжджаючі транспортні засоби. Наявність постійного шуму в робочій зоні приводить до розладу центральної



нервової системи і до таких захворювань як неврози, однак фактичний обмірюваний рівень шуму в робочій зоні склав 37 дБА, що задовольняє нормативному рівню шуму (не повинний перевищувати 50 дБА згідно з ДСН 3.3.6.037- 99).

Джерелом електромагнітного випромінювання в сучасному офісі є візуальні дисплейні термінали. Нормування електромагнітного випромінювання ВДТ здійснюється згідно положень ДСанПіН 3.3.2007-98.

Приміщення відноситься до зони П-Па згідно з ДНАОП 0.001.31-99 і до категорії пожежної небезпеки В. Горючі рідини, пил та волокна у приміщенні не використовуються і не виділяються.

Ймовірними причинами виникнення пожежі можуть бути несправність електрообладнання, короткі замикання внаслідок виходу з ладу електроустаткування, порушення правил протипожежної безпеки тощо.

Для своєчасного попередження пожеж та підвищення оперативності реагування у приміщенні використовується такий комплекс заходів:

- обов'язковий інструктаж персоналу з питань охорони праці;
- зокрема, правила пожежної безпеки у приміщеннях з ЕОМ;
- заборона використання відкритого вогню у приміщенні;
- наявність системи автоматичної пожежної сигналізації з димовими пожежними оповіщувачами;
- наявність шляхів евакуації при виникненні пожежі;
- розміщення схеми евакуації людей при пожежі і ознайомлення з нею персоналу.

В даному розділі було проаналізовано основні вимоги, які ставляться до приміщення в цілому, робочого місця користувача комп'ютером, освітлення, шуму, електромагнітних випромінювань, пожежної безпеки.

Можна стверджувати, що при розробці даної програмної системи було дотримано усіх необхідних вимог та норм охорони праці.

#### **4.2 Створення метеорологічних умов виробничого середовища користувачів ВДТ, ЕОМ, ПЕОМ**

Як фактор виробничого середовища, мікроклімат впливає на теплообмін організму людини з цим середовищем. Необхідною умовою життєдіяльності людини є терморегуляція, тобто здатність організму регулювати віддачу тепла в оточуюче середовище. Цей процес визначається параметрами мікроклімату. Метеорологічні умови визначаються такими параметрами:

- температурою повітря в приміщенні, С;
- відносною вологістю повітря, %;
- рухливістю повітря, м/с;
- тепловим випромінюванням, Вт/м<sup>3</sup>.

Принцип нормування мікроклімату – створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Параметри мікроклімату, вміст шкідливих речовин на робочих місцях, оснащених моніторами, відповідають вимогам ДСН 3.3.6.04299 «Державні санітарні норми мікроклімату виробничих приміщень», ГОСТ 12.1.005-88 «ССБТ. Воздух рабочей зоны. Общие санитарногигиенические требования к воздуху рабочей зоны».

Обчислювальна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні.

В приміщеннях, де встановлені комп'ютери, повинні дотримуватися оптимальні параметри мікроклімату, які визначають комфортні умови. Ці параметри залежать від періоду року, категорії робіт за важкістю, і від теплових характеристик виробничого приміщення (табл. 4.1).

Робота за комп'ютером характеризується малими фізичним навантаженнями, цей вид діяльності належить до категорії легких робіт-1а за критерієм загальних енерговитрат організму (ДСН 3.3.6.042-99).

Системи опалення та кондиціювання повітря в приміщеннях з ПК повинні відповідати СНиП 2.04.05-91 «Отопление, вентиляция и кондиционирование».

Таблиця 4.1. Параметри мікроклімату для приміщень, де встановлені комп'ютери (ГОСТ 12.1.005-88; ДСН 3.3.6.042-99)

Період року	Категорія робіт	Параметр мікроклімату	Величина
Холодний	Роботи легкі 1 а	Температура повітря в приміщенні	22'24 °С
		Відносна вологість	40'60 %
		Швидкість руху повітря	до 0,1 м/с
Теплий		Температура повітря в приміщенні	23'25 °С
		Відносна вологість	40'60 %
		Швидкість руху повітря	0,1,0,2 м/с

Під час роботи комп'ютерної техніки в повітряному середовищі відбувається суттєва трансформація іонного складу, істотно знижується концентрація легких, середніх та важких негативно

зарядних частинок. Така зміна балансу іонного складу призводить до негативного впливу на здоров'я працюючих [24].

Рівні іонізації повітря приміщень при роботі на персональних комп'ютерах визначені в таблиці 4.2 (відповідно до НПАОП 0.03-3.0680).

Таблиця 4.2. Рівні іонізації повітря в приміщенні з ПК

Рівні	Кількість іонів в 1 см <sup>3</sup> повітря	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

### **4.3 Особливості роботи та розлади здоров'я користувачів комп'ютерів, що формується під впливом роботи за комп'ютером**

У операторів ВДТ частіше зустрічаються порушення органів зору, опорно-рухового апарату, центральної нервової, серцевосудинної, імунної та статеві систем, захворювання шкіри. Зафіксована значна кількість скарг операторського персоналу на загальне недомогання, передчасне стомлювання, головний біль, порушення функцій органів зору, які здійснювали несприятливий психофізіологічний вплив на самопочуття та працездатність операторів.

Сучасна професія користувача ВДТ належить до розумової праці, яка характеризується: високою напруженістю зорових функцій; одноманітною позою; великою кількістю стереотипних висококоординованих рухів, що виконуються лише м'язами кистей рук

на фоні малої загальної рухової активності; значним нервово-емоційним компонентом, особливо в умовах дефіциту часу; роботою з великими масивами інформації, що викликає активізацію уваги та інших вищих психічних функцій. Крім того, при роботі з дисплеями на електронно-променевих трубках виникає вплив на користувача цілої низки факторів фізичної природи – електростатичні поля, радіочастотне та рентгенівське випромінювання тощо.

Діяльність професіоналів можна поділити на три групи:

1. Діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження. Наприклад, робота операторів комп'ютерного набору, працівників довідкових служб;
2. Діяльність, яка пов'язана із здійсненням логічних операцій, що постійно повторюються. Це робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва;
3. Діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму. Наприклад, робота інженера-програміста, диспетчерів руху залізничного транспорту, аеропортів тощо.

У користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень досить часто (40—70%) виникають психологічні та поведінкові порушення (нервозність, роздратування, тривога, нерішучість, замкнутість тощо). Серед користувачів ВДТ в США і Європі значного поширення набуло специфічне захворювання, яке отримало назву синдром

комп'ютерного стресу (СКС). СКС супроводжується головним болем, запаленням очей, алергією, роздратованістю, млявістю і депресією. Інформаційне перевантаження користувачів ВДТ супроводжується низкою специфічних захворювань, які називають інформаційними. Першим симптомом їх є головний біль. Дослідження, проведені в США, Німеччині, Швейцарії та інших країнах, показали, що робота з обслуговування ВДТ супроводжується підвищеним напруженням зору, інтенсивністю і монотонністю праці, збільшенням статичних навантажень, нервово-психічним напруженням, впливом різного виду випромінювань та ін. Внаслідок цього серед операторів ВДТ, як зазначають фахівці Всесвітньої організації охорони здоров'я, частіше, ніж в інших групах працюючих, трапляються такі професійні захворювання, як передчасна стомлюваність, погіршення зору, м'язові і головні болі, психічні й нервові розлади, хвороби серцево-судинної системи, онкологічні захворювання та ін. Вважається, що стан організму операторів ВДТ визначається комплексним впливом факторів трудового процесу і середовища, значення яких є неоднаковим. На операторів з малим стажем роботи на ВДТ домінуючий вплив чинять фактори середовища, а на операторів зі стажем понад 5 років – фактори трудового процесу.

Комп'ютерний зоровий синдром (КЗС) – комплекс порушень здоров'я, який може виникати у користувачів персональних комп'ютерів (ПК). Діагноз ставлять, якщо людина, що працює за ПК протягом двох годин, висловлює хоча б дві з десяти скарг:

- головний біль;
- слезотеча;
- різь;

- туман;
- двоїння;
- свербіж;
- важкість в очах;
- фотофобія;
- миготіння знаків на екрані; - нудота.

У користувачів ПК дуже поширені кон'юнктивіти і блефарити, патогенетично пов'язані з КЗС.

Синдром розвивається при умові, що робоче місце організовано неправильно – у користувача незручне крісло, відсутні пюпітри для паперів, підставки для ніг та кистей рук, не встановлена висота і нахил монітора відносно очей, відстань від очей до екрана. За таких умов тіло людини при роботі займає вимушене положення: спина статично напружена, шия витягнута, плечі жорстко фіксовані. Напружені м'язи погіршують кровотік у сонних артеріях, а недостатнє кровозабезпечення головного мозку веде до очманіння, появи головного болю. На фоні шийного остеохондрозу з'являється відчуття випирання очних яблук, туману в очах, мушок та райдужних кіл у полі зору. Розвитку КЗС сприяє поганий мікроклімат приміщення, значна загальна іонізація та мікробне забруднення, а також куріння.

Національною радою з наукових досліджень США для стану зорового дискомфорту був уведений термін «астенопія», який означає «будь-які суб'єктивні зорові симптоми чи емоційний дискомфорт, що є результатом зорової діяльності». Симптоми астенії були класифіковані на «очні» (біль, печія та різь в очах, почервоніння повік та очних яблук, ломота у надбрівній частині тощо) та «зорові» (пелена перед очима, мерехтіння, швидка втома під час зорової роботи та ін.).

У операторів ВДТ «очні» симптоми трапляються частіше, ніж «зорові», причому частота проявів астенії вища у жінок, ніж у чоловіків і більше виражена в осіб середнього і старшого віку.

Причиною вважається електромагнітне випромінювання від ВДТ.

При роботі з ВДТ основне навантаження припадає на всі елементи зорового аналізатора.

Робота з ВДТ може призвести до розвитку короткозорості, так як у користувачів комп'ютерів, в основному, «працює» ближній зір.

При аналізі зорової роботи операторів ВДТ, встановлено, що через дві години частота флуктуацій акомодативної зорової системи зменшується, а внесок низькочастотної компоненти підвищується. Це може бути причиною скарг на втому зорового аналізатора. Тривала робота на ВДТ може призвести до розвитку короткозорості, оскільки у користувачів ВДТ головним чином «працює» ближній зір.

У 100 пацієнтів із 150, які працювали на ВДТ по шість годин на день протягом чотирьох років, були виявлені проблеми з фокусуванням зору.

Робота за комп'ютером характеризується також тим, що постійний напружений погляд на екран монітора зменшує частоту моргання. При цьому погіршується зволоження поверхні очного яблука слюзовою рідиною, яка захищає рогівку ока від висихання, пилу та інших забруднень. Це може призвести до виникнення так званого синдрому Сікка: рогівка висихає і мутніє, і як наслідок розвивається сліпота [25].

В даному розділі було розглянуто основні метеорологічні умови, які повинні забезпечуватися при роботі із ВДТ, ЕОМ та ПЕОМ (тобто як при розробці системи та і при її використанні). Також було



розглянуто особливості роботи за комп'ютером та розлади здоров'я користувачів, що формуються під його впливом.

Отже, розроблена система не прямим способом приймає участь у електромагнітному забрудненні.

Електромагнітні випромінювання мають значний, негативний вплив на здоров'я людини, і з тенденцією розвитку сучасного суспільства він лише збільшується.

Існує ряд заходів та норм які дозволяють захиститися від шкідливого впливу електромагнітного випромінювання. Також дотримуючись деяких простих правил можна мінімізувати його дію.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи на здобуття освітнього ступеня «магістр» було розроблено автоматизовану інформаційну систему впорядкування та обробки даних з використанням сучасних інформаційних технологій, а саме: мови програмування PHP та IntelliJ IDEA для BuInvest.

Результати за темою кваліфікаційної роботи апробовано через представлення на VIII науково-технічній конференції «Інформаційні моделі, системи та технології» м.Тернопіль, 2020 р.

Робота складається зі вступної частини, розділів в кількості чотири, висновку, списку літературних джерел, а також додатків.

До першого розділу основної частини увійшов аналіз стану справ, та можливі варіанти розв'язання проблеми за матеріалами вітчизняних і закордонних публікацій. Сформульовано основну задачу дослідження та виконано пошук актантів та варіантів використання, представлено опис ключових варіантів використання.

До другого розділу запропоновано проєкт та варіант конструювання програмної системи. Зокрема проаналізовано вибір бази даних, мови програмування та технологій для розробки клієнт-серверних технологій, які використовувались, а також застосований підхід до розробки бази даних. Обґрунтовано технології використані

для розробки, описано програмну систему та спосіб взаємодії з серверами баз даних.

В третьому розділі описано встановлення та налаштування системи: середовища сервера та розгортання.

Четвертий розділ – охорона праці та безпека в надзвичайних ситуаціях, як важливий аспект при розробці програмного забезпечення.

Таким чином в результаті роботи було розроблено, запропоновано алгоритм тестування та можливість впровадження програмного продукту призначеного для ведення діяльності конкретного підприємства враховуючи вимоги сьогодення в період безпрецедентних карантинних обмежень. Хоча робота завершена, проте програмний продукт може додатково вдосконалюватись з врахуванням нових потреб. Також, як варіант можна допрацювати продукт на можливість додаткових графічних представлень по проміжним результатам.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – Інженерія програмного забезпечення (Освітньо-професійна програма - «Програмне забезпечення систем», Освітньо-наукова програма - «Інженерія програмного забезпечення») для студентів усіх форм навчання / Упор.: М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк - Тернопіль:

ТНТУ, 2020- 51с..

2. ДСТУ 7152:2010. Видання. Оформлення публікацій у журналах і збірниках. Київ, 2010. 16 с. (Інформація та документація).

3. Національний стандарт України. Інформація та документація.

Бібліографічне посилання. / Київ ДП «УкрНДНЦ» URL:

<http://lib.pu.if.ua/files/dstu8302-2015.pdf>

4. ДСТУ ГОСТ 7.1:2006 СІБВС. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання (ГОСТ 7.1-2003, IDT).

5. ДСТУ 3582:2013 Інформація та документація. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила (ISO 4:1984, NEQ; ISO 832:1994, NEQ). На заміну ДСТУ 3582-97.

6. Основи інформаційних технологій і систем : підручник / В. А. Павлиш,

Л. К. Гліненко, Н. Б. Шаховська. – Львів : Львівська політехніка, 2018. – 620 с. – [ISBN 966-941-264-5](https://doi.org/10.26907/2542-0406.2018.0620).

7. Основи інформаційних технологій і систем: навч. посіб. / В. А. Павлиш,

Л. К. Гліненко ; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». —

Л. : Вид-во Львів. політехніки, 2013. — 500 с. : іл. — Бібліогр.: с. 486—494 (129 назв). — [ISBN 978-617-607-440-3](#)

8. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. — Х.: ХНАМГ, 2010. — 222 с.

9. James Rumbaugh, Ivar Jacobson, Grady Booch (1999). [The unified modeling language reference manual](#) (англ.). Addison Wesley Longman Inc. [ISBN 0-201-30998-X](#).

10. Організація баз даних: практичний курс: Навч. посіб. для студ. / А. Ю. Берко, О. М. Верес; Нац. ун-т «Львів. політехніка». — Л., 2003. — 149 с. — Бібліогр.: 8 назв.

11. Поняття er-моделі. поняття сутності (entity). атрибути. види атрибутів

[Електронний ресурс] — Режим доступу: URL:

<https://www.bestprog.net/uk/2019/01/24/the-concept-of-er-model-the-concept-ofessence-and-communication-attributes-attribute-types-ua/>

12. Shaw, Mary; DeLine, Robert; Klein, Daniel V.; Ross, Theodore L.; Young, David M.; Zelesnik, Gregory (1995-04). [Abstractions for Software Architecture and Tools to Support Them](#). IEEE Trans. Softw. Eng.: 314–335. [ISSN 0098-5589](#)

13. Эбен М., Таймэн Б. "FreeBSD. Энциклопедия пользователя" -К: ООО

"ТИД "ДС", 2002, - 736 с.

14. James Gosling; Bill Joy, Guy Steele, Gilad Bracha (2005). The Java Language Specification, Third Edition. Addison-Wesley. [ISBN 0-321-24678-0](#).

15. Фрэд Лонг та ін. (2014). Руководство для программиста на Java: 75

рекомендаций по написанию надежных и защищенных программ. «Вильямс». [ISBN 978-5-8459-1897-0](#).

16. Web MVC framework [Електронний ресурс] – Режим доступу:  
URL: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
17. Interface Controller [Електронний ресурс] – Режим доступу: URL:  
<http://docs.spring.io/spring-framework/docs/2.0.8/api/org/springframework/web/servlet/mvc/Controller.html>
18. Java [Електронний ресурс] – Режим доступу:  
URL: <https://metanit.com/java/javaee/3.1.php>
19. Java [Електронний ресурс] – Режим доступу:  
URL: <https://metanit.com/java/javaee/3.9.php>
20. Java DataBase Connectivity [Електронний ресурс] – Режим доступу: URL: <http://java-online.ru/jdbc.xhtml>
21. Сервер Apache, установка та налаштування [Електронний ресурс] – Режим доступу: URL: <https://wiki.merionet.ru/servernye-resheniya/30/ustanovka-inastrojka-veb-servera-apache-i-nginx-v-chem-raznica/>
22. [Електронний ресурс] – Режим доступу: URL: <https://helpx.adobe.com/ua/x-productkb/global/install-java-jre-mac-os.html>
23. Apache Tomcat [Електронний ресурс] – Режим доступу: URL: [https://uk.wikipedia.org/wiki/Apache\\_Tomcat](https://uk.wikipedia.org/wiki/Apache_Tomcat)
24. Дистанційний курс «Основи охорони праці» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <http://dl.tntu.edu.ua/index.php>
25. Про затвердження Правил охорони праці під час експлуатації ЕОМ [Електронний ресурс]. – Режим доступу: URL: <http://zakon4.rada.gov.ua/laws/show/z0382-99>

# ДОДАТКИ