

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
СПЕЦИФІКАЦІЯ ПРОЕКТУ.....	9
1. ТЕХНІЧНЕ ЗАВДАННЯ.....	10
1.1. Підстави для розробки.....	10
1.2. Призначення розробки.....	10
1.3. Аналіз вимог до програмного забезпечення.....	10
1.3.1. Функціональні вимоги.....	10
1.3.2. Вимоги до складу та параметрів технічних засобів.....	11
1.3.2.1. Вимоги до інтерфейсу.....	11
1.3.2.2. Вимоги до інформаційної та програмної сумісності.....	13
1.3.2.3. Вимоги до тестування програмного забезпечення.....	13
1.4. Стадії та етапи розробки.....	15
1.5. Порядок контролю і приймання.....	15
2. АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІ ПОКАЗНИКИ.....	16
2.1. Призначення та область застосування.....	16
2.2. Опис та обґрунтування обраної архітектури.....	16
3. ОПИС ПРОГРАМИ.....	17
3.1. Опис та обґрунтування обраних програмних засобів.....	17
3.2. Опис логічної структури.....	19
3.3. Опис модулів та класів.....	19
3.4. Опис і обґрунтування вхідних і вихідних даних.....	20
3.5. Опис інтерфейсу користувача.....	20
3.6. Використані технічні засоби.....	21
4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ.....	23
4.1. Об'єкт випробувань.....	23
4.2. Порядок та методика випробувань.....	23
4.3. Результати випробувань та використані технічні засоби.....	23
ВИСНОВКИ.....	25

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	26
---------------------------------	----

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ОС – операційна система.

БД – база даних.

ІС – інформаційна система.

ПК – персональний комп'ютер.

ІСБО – інформаційна система будівельної організації.

СУБД – система управління базами даних.

## СПЕЦИФІКАЦІЯ ПРОЕКТУ

Призначенням даного проекту являється створення інформаційної бази даних для фотоцентру. Програма призначена для перегляду розроблених об'єктів фотоцентром: офісів, магазинів, кіосків, проявлення та друк фотографій і т.д. по бажанню замовників (професіонали і аматори). Кожна з перерахованих категорій об'єктів має характеристики, властиві тільки цієї або декільком категоріям: наприклад, до характеристик кіосків здійснюється тільки прийом замовлень, тому кожен кіоск закріплений за певною філією, у якій ці замовлення виконуються. Для розробки даного проекту ми використовуємо БД. Тож для чого потрібні бази даних?

Бази даних – це частина інформаційних систем – програмно-апаратних комплексів, що здійснюють зберігання та обробку величезних інформаційних масивів. База даних являє собою певним чином структуровану сукупність даних, що зберігаються разом і обробляються відповідно до деякими правилами. Як правило, база даних моделює деяку предметну область або її фрагмент. Дуже часто в якості постійного сховища інформації баз даних виступають файли.

Інформація, яка зберігається в базі даних, може постійно поповнюватися. Від того, як часто це робиться, залежить її актуальність. Інформацію про об'єкти також можна змінювати і доповнювати. Бази даних, як спосіб зберігання великих обсягів інформації та ефективного маніпулювання нею, використовуються практично у всіх областях людської діяльності.

Даний проект, що розробляється – інформаційна система (ІС) фотоцентру, що дозволяє автоматизувати роботу даної організації. Основна ціль фотоцентру – створювати об'єкти, які замовили клієнти, а також структурно розділити організацію на категорії робітників та виконану роботу. За результатами виконання робіт складається звіт із вказівкою строків виконання робіт і фактичних витрат матеріалів.

## **1. ТЕХНІЧНЕ ЗАВДАННЯ**

### **1.1 Підстави для розробки**

Розробка програмного забезпечення виконується на підставі рішення засідання кафедри ПЗКС про затвердження тем курсових робіт.

*Повна назва розробки:* «ІС фотоцентру» для ПК на основі ОС Windows».

*Коротка назва:* ІСФ, програма.

### **1.2 Призначення розробки**

Призначенням даної розробки є створення інформаційної бази фотоцентру.

Реалізована версія програми призначена для робітників організації, щоб вони мали можливість переглядати структуру організації розподілену по таблицях, а також виконаний звіт по виконаній роботі.

### **1.3 Аналіз вимог до програмного забезпечення**

#### **1.3.1 Функціональні вимоги**

До програмного забезпечення висуваються наступні вимоги:

- Створення бази даних.
- Редагування таблиць бази даних.
- Перегляд інформації по таблицях.
- Забезпечити пошук згідно критеріїв.
- Створення інтерфейсу.

### 1.3.2 Вимоги до складу та параметрів технічних засобів

Для нормальної роботи розробленого програмного продукту необхідний персональний комп'ютер з наступними мінімальними характеристиками:

Параметр	Значення
Процесор	Intel Core 2 Duo 1.8 Ghz
Пам'ять	10 GB RAM
Жорсткий диск	4 GB

#### 1.3.2.1 Вимоги до інтерфейсу

Програма повинна мати зрозумілий, зручний та візуально приємний інтерфейс.

Програмне забезпечення повинно бути оформлене у вигляді html-сторінки, що відображається в браузері з приємно підбраною кольоровою гамою.

Додатково передбачено кнопки переходу між таблицями.

Доступ до програмних засобів має бути забезпечено за допомогою web-браузера Chrome.

Програма повинна складатися з:

- Системи вибору будь-якої таблиці.
- Системи запитів.
- Системи додавання та редагування нових полів.

Основними властивостями, яким повинні задовольняти інтерфейси, є такі:

1. Адаптованість означає, що інтерфейс повинен бути:

- сумісним з потребами та можливостями користувача;
- забезпечувати простоту переходу від виконання однієї функції до іншої;

- забезпечувати користувача на високому рівні вказівками стосовно його можливих дій, а також генерувати належний зворотний зв'язок на його запити;
- надавати користувачу можливість відчувати себе повноправним керівником ситуації при розв'язанні всіх типів задач, тобто, забезпечувати його всією необхідною інформацією; користувач повинен бути впевненим, що він сам розв'язує поставлену задачу;
- забезпечувати користувача різними, взаємно доповнюючими формами представлення результатів в залежності від типу запиту або від характеру отриманого рішення;
- враховувати особливості користувачів різних рівнів.

2. Достатність інтерфейса означає таке:

- допустимі запити користувача повинні бути чіткими і однозначними для користувачів всіх рівнів, а також для прикладних задач всіх типів;
- реакція системи на всі типи запитів також повинна бути однозначною і зрозумілою і, по можливості, простою.

3. Дружність інтерфейсу – це максимальна простота його використання і готовність в повній мірі задовольнити запити користувача при розв'язанні визначеного класу задач.

4. Гнучкість інтерфейсу. Інтерфейс повинен бути максимально простим навіть у випадку, коли розв'язується дуже складна задача. При цьому простота означає таке:

- інтерфейс не повинен бути перевантажений деталями щодо представлення розв'язку поставленої задачі;
- користувач може не охопити всіх подробиць;
- тобто нічого зайвого, крім того, що необхідно для розуміння результату;
- він не повинен містити зайвих декоративних деталей, які відволікають від головної задачі;

- інтерфейс повинен бути консистентним, тобто, ґрунтуватись на використанні відомих, загальноприйнятих методів і засобів представлення інформації;
- в ідеалі процес взаємодії користувача з системою не повинен представляти ніяких труднощів.

#### **1.3.2.2 Вимоги до інформаційної та програмної сумісності**

Необхідною вимогою до програмного забезпечення є встановлена операційна система класу Windows 7/8/10, Linux, MacOS.

Налаштування таблиць зберігається у таблиці БД.

Окремих вимог не висувається.

#### **1.3.2.3 Вимоги до тестування програмного забезпечення**

Для тестування програмного забезпечення необхідно виконати наступні дії:

##### **1) Дії CRUD:**

- запустити програму;
- додати нове замовлення;
- заповнити всі необхідні поля;
- редагування полів таблиці;
- видалення полів;
- закрити програму.

##### **2) Реалізація специфічних запитів:**

1. Одержати перелік і загальне число пунктів прийому замовлень на фотороботи по філіях, по кіосках прийому замовлень у цілому, по фото, які не були забрані замовником центру.
2. Одержати перелік і загальне число замовлень на фотороботи по філіях, кіоскам прийому замовлень, у цілому по фотоцентру, що



надійшли протягом деякого періоду часу.

3. Одержати перелік і загальне число замовлень (окремо простих і термінових) на окремі види фоторобіт по зазначеній філії, кіоску прийому замовлень, що надійшли протягом деякого періоду часу.

4. Одержати суму виторгу із замовлень (окремо простих і термінових) на окремі види фоторобіт по зазначеній філії, кіоску прийому замовлень, що надійшли протягом деякого періоду часу.

5. Одержати кількість надрукованих фотографій у рамках, простих і термінових замовлень по зазначеній філії, кіоску прийому замовлень, фотоцентру в цілому за деякий період часу.

6. Одержати кількість проявлених фотоплівок у рамках простих і термінових замовлень по зазначеній філії, кіоску прийому замовлень, фотоцентру в цілому за деякий період часу.

7. Одержати перелік постачальників у цілому по фотоцентру, постачальників окремих видів фототоварів, що зробили поставки в деякий період, поставки певного обсягу.

8. Одержати список клієнтів у цілому по фотоцентру, клієнтів зазначеної філії, що мають знижки, що зробили замовлення певного обсягу.

9. Одержати суму виторгу від реалізації фототоварів у цілому по фотоцентру, по зазначеній філії, проданих протягом деякого періоду часу.

10. Одержати перелік фототоварів і фірм, що їх виробляють, які користуються найбільшим попитом у цілому по фотоцентру, у зазначеній філії.

11. Одержати перелік реалізованих фототоварів й обсяги їхньої реалізації в цілому по фотоцентру, по зазначеній філії, проданих протягом деякого періоду часу.

12. Одержати перелік робочих місць фото центру в цілому та зазначеної кваліфікації.

Після успішного виконання цих дій тестування програми може вважатися завершеним.

#### 1.4 Стадії та етапи розробки

№ з/п	Назва етапів випускної роботи	Термін виконання етапів роботи
1	Одержання технічного завдання	21/02/2018
2	Опрацювання літературних джерел	28/02/2018
3	Проектування структури	07/03/2018
4	Написання програмного коду	14/03/2018
5	Розробка баз даних	04/04/2018
6	Тестування та відлагодження програми	25/04/2018
7	Оформлення програмної документації	09/05/2018
8	Представлення готової роботи	25/05/2018
9	Захист роботи	Згідно графіку

#### 1.5 Порядок контролю і приймання

Програмне забезпечення повинне бути випробуване та протестоване на наявність помилок.

Порядок контролю і прийому проекту полягає в наступному:

- перевірка правильного функціонування програми керівником проекту;
- проект має бути розгорнений на зовнішньому сервері (посилання на нього видається усім викладачам);
- перевірка відповідності стандартам оформлення документації (перевірка нормоконтролером);
- перевірка правильного оформлення програмної документації керівником проекту;

- документація повинна бути вложена в проект в папку «documentation»;
- попередній захист курсової роботи.

## 2. АРХІТЕКТУРА, ФУНКЦІОНАЛЬНІ ПОКАЗНИКИ

### 2.1 Призначення та область застосування

Призначення даної програми є створення інформаційної бази даних для фотоцентру.

### 2.2 Опис та обґрунтування обраної архітектури

Для даного проекту була вибрана об'єктно-зорієнтована архітектура, оскільки вона є найпоширенішою та найкраще підходить для розробки програм складного рівня. Середовищем для розробки було обрано JetBrains IntelliJ IDEA, а мовою програмування було обрано Java, так як це середовище є зручним, а мова Java однією з найпопулярніших для використання мов.

Було проаналізовано декілька систем керування базами даних, таких як: MySQL, MongoDB, PostgreSQL, MS SQL. На мою думку, для цілей, які виконуватиме ця програма, більш за все відповідає вимогам та підходить система керування базами даних MySQL.

Архітектура складається з таких підрівнів:

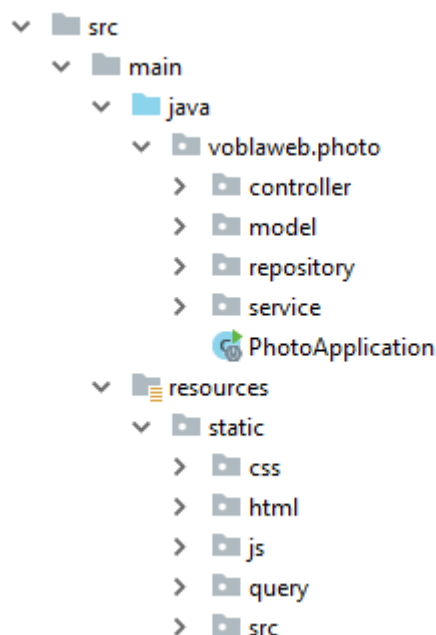


Рисунок 1 – Підрівні системи

### 3. ОПИС ПРОГРАМИ

#### 3.1 Опис і обґрунтування обраних програмних засобів

Розробка програмного забезпечення здійснена в середовищі JetBrains IntelliJ IDEA. Дане середовище розробки – зручний засіб розробки віконних програм для операційної системи Windows 7/8/10. JetBrains IntelliJ IDEA являє собою розширені рефакторинг і форматування для Java, Scala, HTML, CSS, JavaScript, XML і багатьох інших мов. Присутні інструменти для роботи з базами даних і SQL файлами, включаючи зручний клієнт і редактор для схеми бази даних. В ньому поєднані потужні можливості мови програмування Java які, в свою чергу, дозволяють розробнику легко, швидко та якісно створювати програмні продукти.

Я обрав мову програмування Java, бо:

1. Використання об'єктно-орієнтованого підходу дозволяє легко використовувати готові об'єкти в своїх програмах. Для запуску програми необхідна установка JRE, що містить повний набір бібліотек, навіть якщо всі вони не використовуються в додатку.

2. Присутній суворий контроль типів, обов'язковий контроль виняткових ситуацій.

3. Механізм автоматичного генерування документації на основі коментарів, розміщених в тексті програм.

4. Можливо реалізувати абсолютно різні за способом функціонування і сфері використання програми.

Framework SpringBoot:

1. Реалізує модель розробки, засновану на кращих стандартах індустрії, і робить її доступною в багатьох областях Java.

2. Працює з системами керування базами даних на Java-платформі, використовуючи JDBC- і ORM-засоби і забезпечуючи вирішення завдань, які повторюються в великому числі Java-based environments.

3. Надає свій шар доступу до баз даних і підтримує всі популярні СУБД.

Базу даних MySQL, бо:

1. Може підтримувати роботу БД значних розмірів (берігають до 50 млн. записів).

2. Працює на різних платформах.

3. Має систему контролю доступу до даних, забезпечує шифрування даних при передаванні.

4. Швидкість функціонування.

Архітектуру Proxu, бо:

1. Писати і підтримувати невеликі сервіси завжди простіше, ніж великі.

2. Неважко відстежити залежності між сервісами.

ORM – Hibernate, бо:

1. Усуває безліч повторюваного коду.

2. Підтримує XML так само як і JPA анотації, що дозволяє зробити реалізацію коду незалежною.

3. Легко інтегрується з іншими Java EE фреймворками.

4. Підтримує ледачу ініціалізацію використовуючи проху об'єкти і виконує запити до бази даних тільки по необхідності.

5. Підтримує різні рівні cache, а отже може підвищити продуктивність.

Front-end (HTML5, CSS, angular1,5) бо:

1. Створено нові елементи й атрибути, за допомогою яких стало значно легше просувати сайти.

2. Розроблено теги, що дозволять розбивати сторінки сату на інтуїтивно-зрозумілі блоки (header, footer, section, nav, article, aside, figure).

3. Завантаження сторінок сайту відбувається швидше.

4. Не потрібно підключати додаткові розширення до браузера, щоб відтворити аудіо чи відео ( достатньо всього вставити тег video або audio).

### 3.2 Опис логічної структури

Логічну структуру бази даних зручно подати у вигляді наступної діаграми варіантів використання (рис.2).

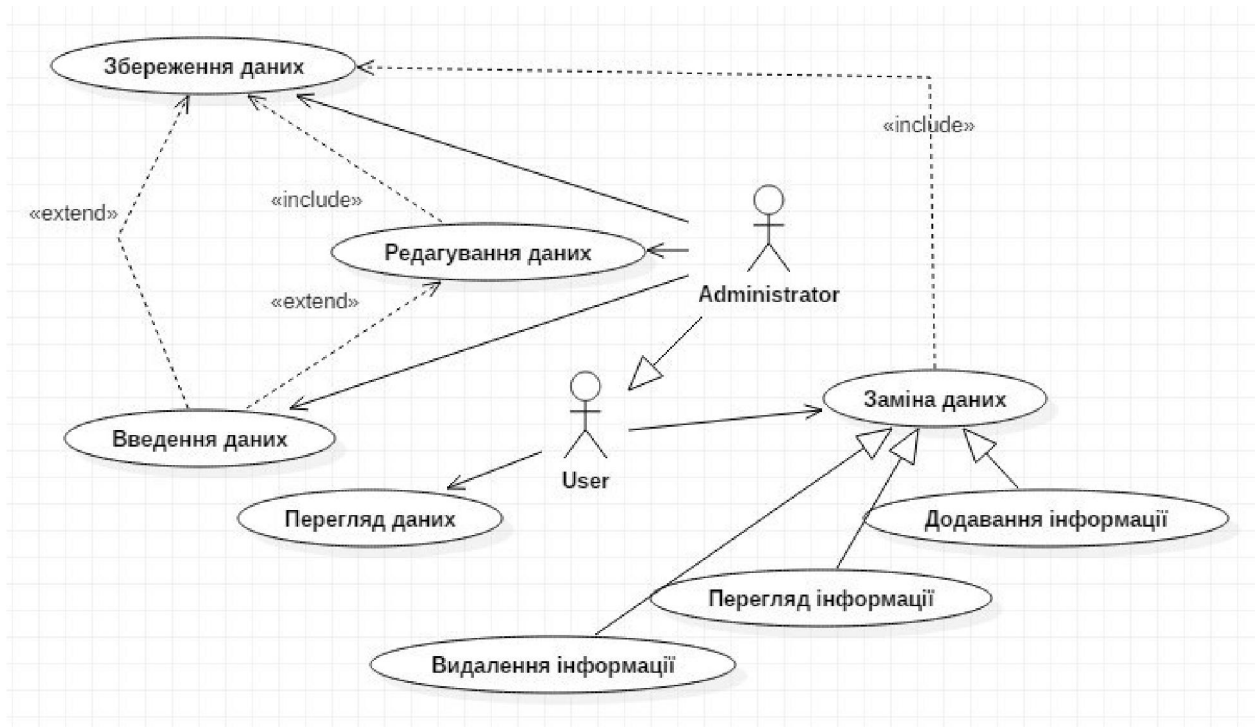


Рисунок 2 – Логічна структура бази

### 3.3 Опис модулів класів

Робота розробленого програмного забезпечення реалізується наступними файлами класами:

Клас Branches – це клас для опису об’єкту «віток» фотоательє:

```

public class Branches {
    private int id;
    private String address;
    private int amountOfWorkplaces;
}
  
```

Клас Clients – це клас для опису об’єкту «клієнтів», що користуються послугами фотоательє:

```
public class Clients {
    private int id;
    private String name;
    @ManyToOne(fetch=FetchType.EAGER)
    @JoinColumn(name = "branchId")
    private Branches branchId;
    private String type;
    private int discount;
```

Клас OfficeOrders – це клас для опису об’єкту «офісні замовлення», що використовується для офісних замовлень:

```
public class OfficeOrders {
    private int id;
    private PhotoGoods photoGoodId;
    private Kiosks kioskId;
    private Date dateOfSupply;
    private int amountOfGoods;
    private int price;
    private Date dateOfSelling;
```

Клас PhotoGoods – це клас для опису об’єкту «фототовари», що використовується для отримання фототоварів:

```
public class PhotoGoods {
    private int id;
    private String name;
    private GoodType goodTypeId;
    private String supplierName;
```

Клас Services – це клас для опису об’єкту «сервісів», що використовується для отримання сервісів:

```
public class Services {
    private int id;
    private String name;
    private int price;
```

Клас ServiceSupplies – це клас для опису об’єкту «сервісні замовлення», що використовується для сервісних замовлень:

```
public class ServiceSupplies {
    private int id;
    private Date date;
    private Kiosks kioskId;
    private Services serviceId;
    private Clients clientId;
```

Клас Supplies – це клас для опису об’єкту «замовлення», що використовується для створення замовлень:



```
public class Supplies {
    private int id;
    private Kiosks kioskId;
    private TypeOfSupplies typeOfSupplyId;
    private int price;
    private int amountOfPhotosPerFrame;
    private int totalAmountOfPhotos;
    private String format;
    private String paperType;
    private Date supplyDate;
    private String done;
    private Clients clientId;
```

Клас TypeOfSupplies – це клас для опису об’єкту «тип замовлень», що використовується для отримання типів замовлень:

```
public class TypeOfSupplies {
    private int id;
    private String name;
```

Клас GoodType – це клас для опису об’єкту «тип товару», що використовується для отримання типів товару:

```
public class GoodType {
    private int id;
    private String name;
```

Клас HurrySupplies – це клас для опису об’єкту «термінові замовлення», що використовується для створення термінових замовлень:

```
public class HurrySupplies {
    private int id;
    private Branches branchId;
    private TypeOfSupplies typeOfSupplyId;
    private int price;
    private int amountOfPhotosPerFrame;
    private int totalAmountOfPhotos;
    private String format;
    private String paperType;
    private Date supplyDate;
    private String done;
    private Clients clientId;
```

Клас Kiosks – це клас для опису об’єкту «кіоски», що використовується для отримання кіосків:

```
public class Kiosks {
    private int id;
    private String address;
    private int amountOfWorkplaces;
    private Branches branchId;
```

### **3.4 Опис і обґрунтування вхідних і вихідних даних**

Вхідними даними для розробленого продукту є таблиці в базі даних, такі як мости, будинки, робітники, бригади, будівельне керування, техніка, бригадир. А також при відкритті кожної з таблиць ми можемо маніпулювати з нею та маємо кнопку повернення назад.

### **3.5 Опис інтерфейсу користувача**

Робота з програмою вимагає лише базових знань та навичок роботи з web-браузером Chrome.

Розроблене програмне забезпечення – набір класів, які об'єднані з графічним інтерфейсом користувача, а також базою даних.

На (рис.3) представлено головну форму програми. Дана форма складається із списку кнопок по яких можна перейти, щоб відобразились таблиці бази даних. В майбутньому в програмі буде передбачений механізм реалізації різних запитів. Поки що тестово користувач може переглянути таблиці БД, оперувати з ними, виконувати такі команди з полями таблиці, як видалення, редагування, створення та зчитування інформації.



# KOKOS. inc

Сторінка записів

Титульна сторінка

Сторінка запитів

Рисунок 3 – Головне вікно програми

Тут ми бачимо три основні кнопки, а саме:

- Сторінка записів – кнопка для перегляду таблиць фотоательє.
- Титульна сторінка – кнопка для перегляду титульної сторінки.
- Сторінка запитів – кнопка для перегляду запитів до БД.

### 3.6 Використані технічні засоби

Розробка програмного продукту здійснювалася на персональному комп'ютері з наступною конфігурацією:

- Процесор – Intel(R) Core™ i5-2520M CPU @ 2,50 GHz 2,50 GHz.
- ОЗП – 6 GB.
- Відеоадаптер – Nvidia NV4200.
- Жорсткий диск – 255 GB.
- DVD-RW – TOSHIBA MK2561GSY.

Даний комп'ютер працює під управлінням ОС Windows 10 Professional.

## **4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ**

### **4.1 Об'єкт випробувань**

Об'єктом випробувань є створене в процесі виконання курсової роботи програмне забезпечення. Для перевірки коректності роботи програми було проведено ряд тестів. Мета випробувань – підтвердження задоволення усіх функціональних та не функціональних вимог до програмного забезпечення.

### **4.2 Порядок та методика випробувань**

Порядок проведення випробувань програмного забезпечення:

- запустити програму;
- додати нове замовлення;
- заповнити всі необхідні поля;
- редагування полів таблиці;
- видалення полів;
- перевірка реалізації потрібних запитів;
- закрити програму.

### **4.3 Результати випробувань та використані технічні засоби**

Випробування програмного продукту проводилось шляхом тестування різних введених функцій. При тестуванні помилок не виявлено. Програма працює правильно.

Тестування проводилось на пристроях з наступною конфігурацією:

Таблиця 1 – Характеристика пристроїв

Параметри	Пристрій №1	Пристрій №2
Процесор	Intel(R) Core™ i5-2520M	Intel(R) Core™ i7-2365M
ОЗП	6 GB	8 GB
Версія ОС	Windows 10	Windows 10
Пам'ять	255 ГБ	800 ГБ

Результати випробувань свідчать про те, що розроблене програмне забезпечення повністю задовольняє п.1.3.2.3 *Вимоги до тестування програмного забезпечення* технічного завдання.

## ВИСНОВОК

В даний час бази даних відіграють велику роль у веденні будь-якої документації. Майже кожна структура в наш час обладнана комп'ютером, який допомагає вести облік набагато зручніше ніж на папері. Автоматизація роботи з великими базами даних має велике значення в наш час.

Дана програма призначена для заповнення відповідних структур організації. На даному етапі програма містить в собі лише базові вимоги. Тобто, в базу заносяться лише дані про співробітників та виконану ними роботу. В подальшому планується розширення цієї програми.

Тестування показало працездатність програмного забезпечення та відповідність вимогам.

Дане програмне забезпечення має наступні можливості:

- додавання, редагування та видалення полів будь-якої із таблиць бази даних;
- перегляд результатів.

При розробці даної програми, використав на практиці наступні технології:

- проектування БД;
- моделювання БД;
- програмування мовою Java;
- Framework SpringBoot;
- RESR Service;
- Uml – діаграми;
- Git;
- Object Relation Mapping;
- SQL query;
- Web-дизайн (HTML, CSS, Angular 1,5);
- MOCK Testing;
- Full Stack;

- MySQL;
- Java;
- Framework SpringBoot;
- JDBC;
- Angular 1.6.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Агальцов, В.П. Базы данных. В 2-х т. Т.2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. – М.: ИД ФОРУМ, НИЦ ИНФРА – М, 2013. – 272 с.
2. Кириллов, В.В. Введение в реляционные базы данных. Введение реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб.: БХВ-Петербург, 2012. – 464 с.
3. Философия Java. 4-е полное изд. – СПб.: Питер, 2015 – 1668 с.: ил. (Серия «Классика computer science»).