МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«РОЗРОБКА СКРАПЕРУ РЕСУРСІВ З ПОШУКУ РОБОТИ»**

Виконав: студент 2 курсу, групи 8.1212-іпз-1
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми інженерія програмного забезпечення
(назва освітньої програми)
Д.А. Братющенко
(ініціали та прізвище)

Керівник завідувач кафедри програмної інженерії, доцент, к.ф.-м.н. Лісняк А.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук, доцент, к.т.н. Решевська К.С.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення
(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.
(підпис)

“ ____ ” _____ 2023 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Братющенку Дмитру Андрійовичу
(прізвище, ім’я та по-батькові)

1. Тема роботи Розробка скраперу ресурсів з пошуку роботи

керівник роботи Лісняк Андрій Олександрович, к.ф.-м.н, доцент
(прізвище, ім’я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 01 » травня 2023 року № 642-с

2. Строк подання студентом роботи 27.11.2023 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Практична частина з розробки сервісу.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 03.05.2023 р.

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	17.05.2023	
2.	Збір вихідних даних.	07.06.2023	
3.	Обробка методичних та теоретичних	28.06.2023	
	джерел.		
4.	Розробка першого та другого розділу.	30.08.2023	
5.	Розробка третього розділу.	08.11.2023	
6.	Оформлення та нормоконтроль	20.11.2023	
	кваліфікаційної роботи магістра.		
7.	Захист кваліфікаційної роботи.	14.12.2023	

Студент _____ Д.А. Братющенко
(підпис) (ініціали та прізвище)

Керівник роботи _____ А.О. Лісняк
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ А.В. Столярова
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка скраперу ресурсів з пошуку роботи»: 46 с., 17 рис., 14 джерел.

ДОДАТОК, ІНСТРУМЕНТ, ПОШУК РОБОТИ, РОЗРОБКА, СКАРАПЕР, GOOGLE, JAVA.

Об’єкт дослідження – вебресурси та інформаційні технології.

Мета роботи: створення інструменту для оптимального пошуку роботи згідно з навичками користувача.

Метод дослідження – методи збору та аналізу вимог до програмного забезпечення, методи моделювання, проєктування, конструювання та тестування програмного забезпечення.

При розробці додатку був проведений аналіз предметної області, обрано та спроєктовано архітектуру системи за допомогою UML у вигляді діаграм. Реалізовано сервісну частину за допомогою Java та пов’язаних з пошуком та сортуванням данних інструментів.

В результаті роботи отримано програмне забезпечення для пошуку роботи з вебресурсів.

SUMMARY

Master’s qualifying paper «Development of the Job Search Resources Scraper»: 46 pages, 17 figures, 14 references.

APP, TOOL, JOB SEARCH, DEVELOPMENT, SCRAPER, GOOGLE, JAVA.

The object of the study is web resources and information technologies.

The aim of the study is creation of a tool for optimal job search according to the user’s skills.

The method of research are methods of collecting and analyzing software requirements, methods of modelling, designing, constructing, and testing software.

When developing the application, an analysis of the subject area was carried out, the system architecture was selected and designed using UML in the form of diagrams. The service part was implemented using Java and tools related to searching and sorting data.

As a result of the work, software for searching for work from web resources was obtained.

ЗМІСТ

Завдання на кваліфікаційну роботу……..…..2
Реферат…….…….…….…….……4
Summary…….…….…….…….……...5
Вступ…….…….…….…….……...7
1 Аналіз архітектури сервісів з пошуку роботи…….…….…….... 9
1.1 Аналіз існуючих сервісів з пошуку роботи…….…….……..… 9
1.2 Аналіз актуальних потреб для сервісу з пошуку поботи…… 12
1.3 Постановка задачі…….…….…….…….……16
1.3.1 Функціональні вимоги..……....16
1.3.2 Технічні вимоги……16
1.3.3 Тести та валідація……..17
2 Проєктування скраперу …….……..……..18
2.1 Визначення вимог до скраперу з пошуку роботи…….……18
2.2 Розробка архітектури скраперу з пошуку роботи……..……. 23
2.3 Діаграма варіантів використання…….……...27
2.4 Діаграма послідовності…….…….……...29
2.5 Діаграма розгортання…….…….…….....31
3 Реалізація скраперу ……...……...…….……33
3.1 Створення бекенд функціоналу за допомогою Java...……..33
3.2 Створення бази даних за допомогою PostgreSQL…….……...40
3.3 Розробка UI за допомогою React…….……...42
Висновки…….…….…….…….……..…… 45
Перелік посилань……...46

ВСТУП

На сьогоднішній день ринок праці постійно змінюється у вимогах, попиті та пропозиціях, що ускладнює пошук роботи для кандидатів і заповнення вакансій для роботодавців.

У світі, де швидкість, точність та доступність інформації визначають успішність, розробка програмного забезпечення для ефективного пошуку роботи має величезне значення. З розвитком Інтернету з’являється безліч сайтів з вакансіями, але вибір найбільш підходящої роботи може займати багато часу і зусиль. Для полегшення цього процесу було розроблено парсер з пошуку роботи, який дозволяє автоматизувати процес збору та аналізу інформації з різних сайтів з вакансіями. Цей інструмент спрямований на надання користувачам зручного та ефективного механізму для пошуку та аналізу вакансій, забезпечуючи швидкий доступ до актуальної та релевантної інформації.

Кількість користувачів, що визначають обсяг наборів даних у системах пошуку роботи, представлена числовими оцінками. Наприклад, сайт www.jobs.ua на сьогодні є одним із лідерів на українському інтернет-ринку праці, де надають комплекс рекламних та інформаційних послуг [3]. Основні користувачі цього сайту – це кандидати, які шукають роботу, та роботодавці, що розглядають кандидатів для вакансій. Щодня понад 70 000 претендентів переглядають вакансії на сайті Jobs.ua.

На сьогодні вже існує чимало готових інструментів, які дозволяють збирати, аналізувати дані і приводити їх до необхідного вигляду для подальшої обробки або зберігання. Однак більшість з них виконують автоматичні запити користувачів і не надають рекомендацій на основі аналізу ефективності пошуку, проведеного попередніми користувачами.

Цей проєкт не лише передбачає розробку програмного забезпечення, але й включає в себе вивчення передових технологій у сфері обробки даних та

алгоритмів машинного навчання. Основними завданнями є створення ефективного інструменту для користувачів, що дозволить швидко та точно аналізувати актуальні вакансії на ринку праці.

Цей дипломний проєкт є спробою поєднати технологічну експертизу та потреби ринку праці, надаючи інструмент, що сприятиме оптимізації пошуку роботи та допоможе індивідуальним користувачам отримати доступ до робочих можливостей, які найкраще відповідають їхнім професійним потребам та уподобанням.

Об’єктом задачі є розробка програмного забезпечення, яке буде здійснювати збір інформації про робочі місця з вебсайтів різних компаній, робочих платформ та ресурсів з пошуку роботи. Це включає в себе створення алгоритмів для обробки отриманих даних, їх аналізу та подальше представлення користувачам у зручному форматі.

Мета: розробка та впровадження програмного скрапера, призначеного для збору та обробки інформації про робочі можливості з різних джерел.

Задачі:

– сформувати вимоги до системи;
– спроєктувати та побудувати архітектуру системи;
– реалізувати систему скрапера з пошуку роботи;
– протестувати роботу системи.

Об’єкт дослідження: аналіз методів парсингу вебсторінок, розробка алгоритмів для збору та структурування інформації про вакансії, а також вивчення технологій обробки даних для підвищення ефективності та точності результатів.

Методи дослідження: моделювання, проєктування, програмний, аналітичний.

1 АНАЛІЗ АРХІТЕКТУРИ СЕРВІСІВ З ПОШУКУ ПРОБОТИ

1.1 Аналіз існуючих сервісів з пошуку роботи

Було проведно дослідження існуючих інформаційних систем і був зроблен аналіз за показниками цільової аудиторії, спеціалізації, актуальності,переваг і недоліків.

На сьогоднішній день існують багато різних сервісів з пошуку роботи: соціальні мережі, вебсервіси. Одні з найбільш поширених:

– LinkedIn;
– jobs.ua;
– work.ua;
– dou.ua;
– ua.jooble.org.

Спільні риси різних сервісів: надання доступу до баз даних з вакансіями та резюме, організування зустрічі кандидатів та менеджерів з підбору персоналу, фільтрування вакансій та шукачів роботи за різними показниками.

Отже, наприклад, сайт пошуку праці “work.ua” – найбільший український ресурс для знаходження працівників [1]. Ця платформа володіє розширеними можливостями пошуку вакансій за різними категоріями, містами, компаніями, посадами, що дозволяє швидко отримувати результати, що найбільше відповідають поставленому запиту (див. рис. 1.1).

Сайт пропонує зручну форму для додавання резюме, що допомагає кожному, хто шукає роботу, ефективно представити себе потенційному роботодавцю. Є можливість перегляду новин та статей, які стосуються сфери зайнятості.

Користувачі, які переходять до розділу «Для роботодавців», можуть переглядати наявні резюме за різними категоріями або містами. У разі необхідності вони можуть розмістити власну вакансію.

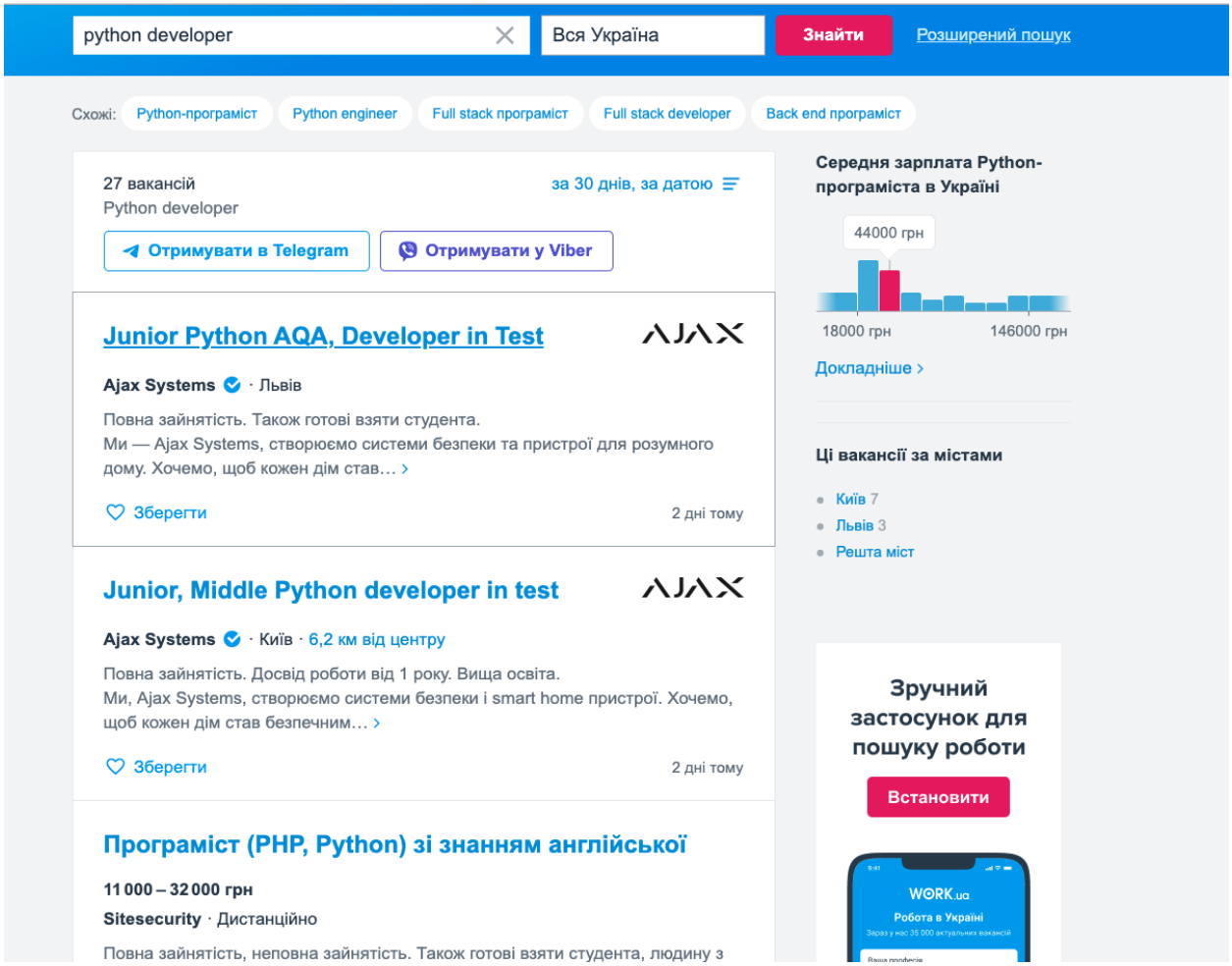

Рисунок 1.1 – Зразок пошуку роботи на сайті work.ua

Щодо зручності використання цієї платформи для студентів, то вони можуть перейти за посиланням «Робота для студентів», щоб переглянути вакансії, призначені для пошуку працівників-студентів. Однак, у більшості таких вакансій вказується, що для працевлаштування потрібний досвід роботи протягом 1 року. На жаль, спеціальних модулів на сайті для пошуку робочих місць студентами без досвіду роботи немає.

Тим часом, багато людей щомісяця відвідують вебсайт “rabota.ua”, щоб знайти роботу, а провідні компанії України розміщують свої вакансії й знаходять співробітників (див. рис. 1.2) [2]. Він пропонує різноманітні робочі вакансії для кандидатів різного рівня кваліфікації та спеціалізацій. “Robota.ua” ставить своєю метою полегшення пошуку роботи для користувачів, надаючи зручний інструмент для взаємодії між кандидатами та роботодавцями на українському ринку праці. Цей майданчик для пошуку роботи обирають 75%

українських інтернет-користувачів. Сайт постійно оновлюється новими вакансіями, що дозволяє користувачам бути в курсі найсвіжіших пропозицій на ринку праці. Кожна вакансія супроводжується інформацією про компанію-роботодавця, її описом, рейтингом та відгуками. Сайт надає можливість користувачам використовувати різні фільтри (місцезнаходження, зарплатні очікування, тип роботи тощо) для точного вибору вакансій.

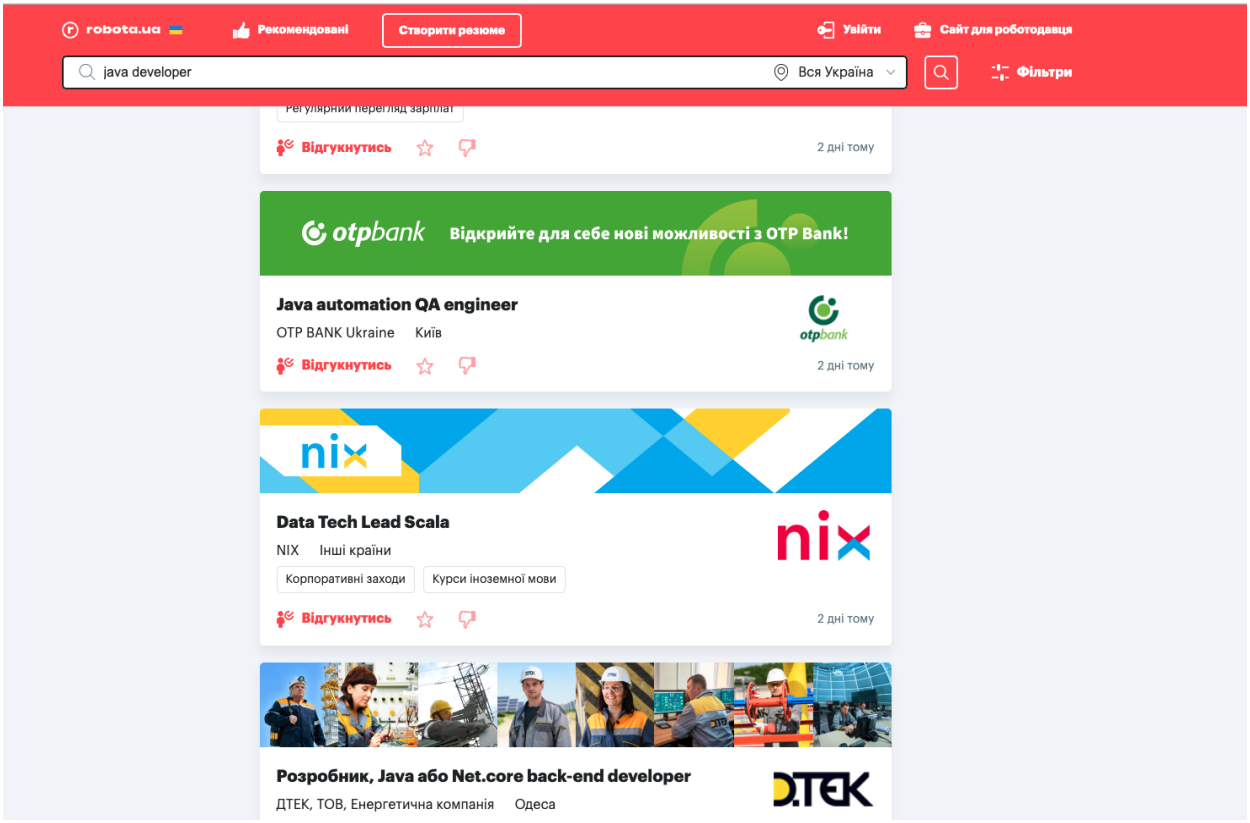

Рисунок 1.2 – Зразок пошуку роботи сайту rabota.ua

На цьому сайті доступні всі стандартні функції: заповнення анкети, сортування вакансій за категоріями, датами публікації, вибором регіону.

Розробники передбачили наявність логотипів відомих компаній на головній сторінці для спрощення процесу. Це допомагає знайти потрібну вакансію на потрібну посаду ще швидше.

Для студентів, які шукають роботу, є спеціальний модуль «Робота для студентів», який дозволяє переглядати відібрані вакансії та використовувати фільтри для цих вакансій.

Даний вебсайт має аналітичні можливості й надає наступну інформацію: статистику по розподілу зарплати за вакансіями та резюме, середню заробітну плату, кількість вакансій та активних кандидатів, пропозиції компаній, окрім зарплати, такі як бонуси, розташування офісу, знижки для працівників тощо, а також інформацію про топові сфери та професії на ринку праці.

LinkedIn.com є комбінацією найкращих характеристик двох типів пошукових систем: можливість шукати роботу в Інтернеті за допомогою власної системи пошуку праці та можливість взаємодіяти з друзями-колегами та приватними особами, щоб розширити пошукові можливості [5].

Оголошення про вакансії на LinkedIn мають високу якість, і якщо ви маєте зв’язки з кимось, хто вже ознайомлений з цією конкретною роботою, ви можете вступити до розмови ще до того, як надіслати резюме.

Наполегливо створіть вражаючий профіль на LinkedIn. Для пошуку роботи на цій платформі доступний широкий спектр фільтрів. Серед них дата публікації, простий звернення, тип роботи, місцезнаходження, компанія, галузь, робоча посада, списки з менше ніж 10 кандидатами, робота на відстані (також віддалена робота), звання, переваги та рівень досвіду.

Переваги використання:

– міцні зв’язки у професійній спільноті;
– додатки, доступні в один клік;
– деякі списки показують кількість поточних заявників;
– велика кількість опцій фільтрації.

Розкриття вашого профілю перед рекрутерами може викликати зацікавленість рекрутерів у вашій поточній роботі.

1.2 Аналіз актуальних потреб для сервісу з пошуку роботи

Щодо отримання актуальності потреб можна виділити два основних підходи: опитування (роботодавців, кандидатів на роботу) і аналіз статистики

сайтів. Опитування охоплює аналіз структури цільової аудиторії користувачів за такими характеристиками: сфера діяльності, посада, регіон, робочий стаж і може проводитись за різними критеріями:

– опитування з огляду на джерело успішного пошуку (див. рис. 1.3);
– за результатами пошуку: за посадою, за регіоном, за рівнем досвіду (див. рис. 1.4, 1.5).

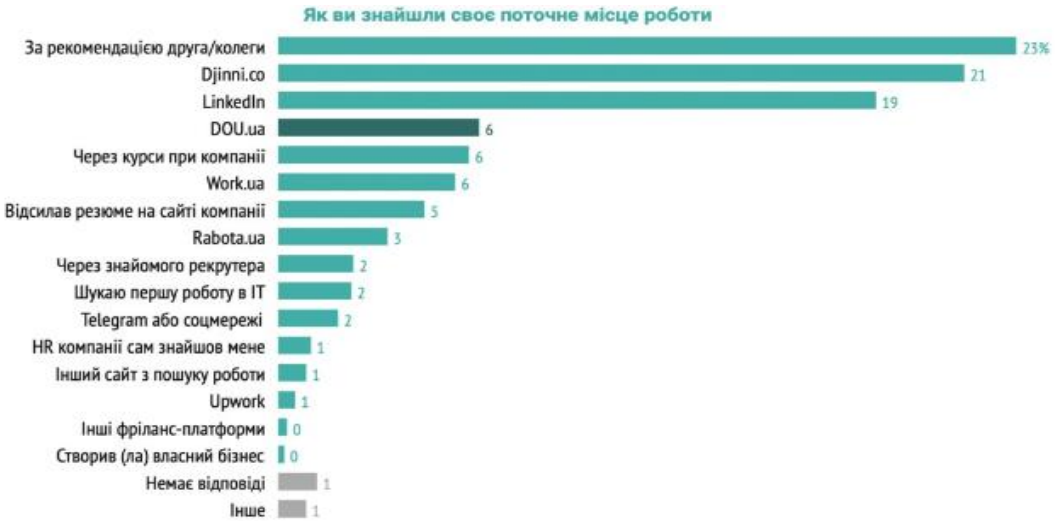

Рисунок 1.3 – Структура відповідей по результативних системах пошуку роботи

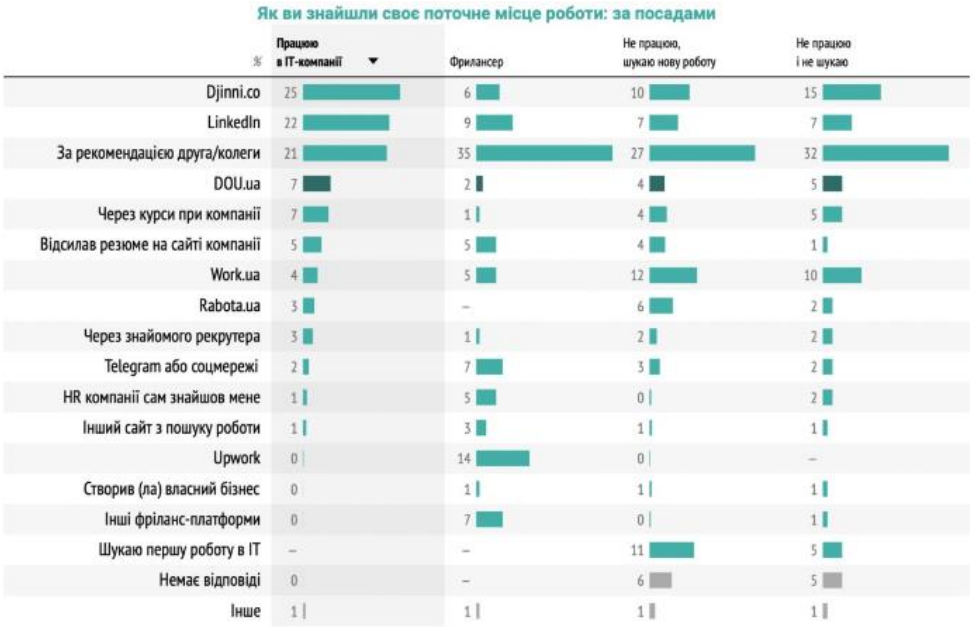

Рисунок 1.4 – Структура відповідей по результативних системах пошуку роботи по посадах у сфері IT

%	Програміст	QA	Analyst	PM	Дизайнер	DevOps	Інше (технічна)	Інше (нетехнічна)
Djinni.co	73	75	71	65	63	66	60	61
LinkedIn	66	55	62	61	50	66	53	48
Через знайомих	46	44	46	51	46	44	47	46
DOU.ua	37	46	48	44	38	38	44	49
Розсилка резюме в компанії	24	28	32	26	32	22	33	34
Work.ua	6	9	12	10	16	4	16	17
Robota.ua	4	8	13	9	13	5	13	15
Skyworker	0	0	–	–	–	–	–	0
Upwork	0	0	–	1	1	1	0	–
Соцмережі	0	0	–	–	–	1	1	1

Рисунок 1.5 – Структура відповідей по результативних системах пошуку роботи по стажу роботи

Опитування з метою вивчення поведінки користувачів: наприклад, чи вони відстежують сайти пошуку роботи та з якою метою.

Отже, у цьому прикладі, щодо каналів пошуку роботи в IT-сфері, найбільш результативними виявилися особисті рекомендації: 26% опитаних повідомили, що знайшли своє поточне місце роботи через пораду друга або колеги; ще 4% отримали запрошення від знайомого рекрутера. Наступним за ефективністю є LinkedIn: за допомогою цієї мережі працевлаштувались 16% опитаних.

Профільні сайти також активно допомагають у пошуку роботи в IT: на Джині та DOU роботу знайшли 12% та 10% опитаних відповідно [4, 6]. Трохи менш результативним способом пошуку роботи в IT виявилися сайти пошуку праці Work.ua (7%) та Rabota.ua (7%).

Курси та навчальні центри, пов’язані з компаніями, забезпечили роботу для 6% опитаних. Ще 6% отримали пропозицію роботи після надсилання резюме на сайті компанії [7]. Також, 2% повідомили, що їх зв’язав незнайомий рекрутер з запрошенням на співбесіду через Skype, електронну пошту або телефонний дзвінок.

Основним інструментом для пошуку роботи та відбору кандидатів на

роботу є спеціалізовані соціальні мережі та вебсайти.

Досліджено зв’язок поведінки під час пошуку роботи з результатами успішного працевлаштування за такими показниками.

Загальна активність у пошуку роботи:

– кількість співбесід;
– кількість пропозицій роботи;
– зайнятість;
– якість отриманої роботи.

Активний пошук роботи:

– кількість співбесід;
– кількість пропозицій роботи;
– зайнятість;
– якість отриманої роботи.

Підготовчий пошук роботи:

– кількість співбесід;
– кількість пропозицій роботи;
– зайнятість;
– якість отриманої роботи.

Неформальний пошук роботи:

– кількість співбесід;
– кількість пропозицій роботи;
– зайнятість;
– якість отриманої роботи.

Офіційний пошук роботи:

– кількість співбесід;
– кількість пропозицій роботи;
– зайнятість;
– якість отриманої роботи.

Якість пошуку роботи:

– кількість співбесід;

– кількість пропозицій роботи;
– зайнятість;
– якість отриманої роботи.

1.3 Постановка задачі

1.3.1 Функціональні вимоги

На базі вищенаведених показників в межах цієї роботи необхідно розробити сервіс з пошуку роботи. Цей сервіс – скрапер, який має збирати інформацію з різних веб джерел по пошуку роботи і зберігати у базі даних. Збережені результати мають відібражатися користувачу. Фільтрація скраперу буде відбуватися за рахунок запиту користувача, який буде мати змогу записати його у вігляді запросу з критерій роботи, яка цікавить корустувача, у поле web UI скрапера.

Перелік функціональних вимог:

– розробка парсера для збору даних з вебсайтів, що пропонують робочі можливості;
– отримання інформації про вакансії: назва вакансії, опис, місцезнаходження, умови праці, заробітна плата тощо;
– процес аналізу та структурування даних для подальшого зберігання;
– система фільтрації та сортування даних за різними параметрами для користувача (місцезнаходження, зарплата, тип роботи тощо).

1.3.2 Технічні вимоги

Для будь-якого продукту, системи або послуги правильно сформульовані технічні вимоги є ключовим етапом у процесі розробки,

оскільки вони визначають параметри, якими повинен керуватися процес розробки для досягнення певного результату або функціоналу. Наведемо нижче основні технічні вимоги скраперу:

- мова програмування: Java;
- використання бібліотек для парсингу вебсторінок;
- використання бази даних для зберігання зібраної інформації;
- розробка алгоритмів для оптимального та швидкого збору даних з різних джерел;
- реалізація інтерфейсу користувача через вебсайт або десктопну програму для взаємодії та відображення результатів.

1.3.3 Тести та валідація

Тести спрямовані на виявлення помилок і несуттєвих недоліків, в той час як валідація підтверджує, що результат розробки відповідає потребам і очікуванням користувачів та специфікаціям проєкту. Обидва процеси важливі для забезпечення якості продукту. А саме:

- розробка тестових сценаріїв для перевірки функціональності та коректності роботи програми;
- валідація роботи програми з реальними вебсторінками та перевірка правильності збору та обробки даних.

2 ПРОЄКТУВАННЯ СКРАПЕРУ

2.1 Визначення вимог до скраперу з пошуку роботи

Скрапер з пошуку роботи повинен об’єднувати найкращі практики існуючих вебдодатків з пошуку роботи, та містити нові функції яких немає у існуючих сервісах з пошуку роботи. Також він повинен бути зрозумілим для користування.

Загальні риси вебсервісів з пошуку роботи включають в себе різноманітні функції та можливості, які спрощують процес пошуку роботи для кандидатів і полегшують пошук кваліфікованих працівників для роботодавців. Ось деякі загальні риси цих сервісів:

- соціальні функції;
- мобільність;
- пошук та фільтрація вакансій;
- можливість швидкого та зручного пошуку вакансій за різними критеріями, такими як регіон, галузь, рівень зарплати, тип роботи тощо;
- розміщення резюме та вакансій;
- функція для кандидатів розміщувати свої резюме та для роботодавців – вакансії, що спрощує знаходження та відбір потенційних кандидатів чи робочих місць;
- система рекомендацій;
- алгоритми, які рекомендують вакансії або кандидатів на основі історії пошуку або попередньої взаємодії користувача з платформою;
- управління профілем;
- можливість створювати та редагувати профілі користувачів з деталями про навички, досвід роботи, освіту тощо;
- сповіщення та аналітика;

– система сповіщень щодо нових вакансій, відповідей на заявки, а також збір та аналіз даних про активність користувачів;
– система відгуків та рейтингів;
– можливість залишати відгуки та оцінки про роботодавців або кандидатів, що допомагає іншим користувачам у прийнятті рішень;
– адаптація до мобільних платформ зручна для використання на смартфонах та планшетах;
– безпека та конфіденційність;
– забезпечення захисту особистої інформації користувачів та даних, які збираються платформою;
– можливість зв’язку між кандидатами та роботодавцями через соціальні мережі або чати.

Ці риси є загальними та можуть варіюватися залежно від конкретної платформи та її функціональних можливостей. Однак вони спрямовані на полегшення та оптимізацію процесу пошуку роботи для кандидатів та роботодавців.

Загальні риси вебсервісів з пошуку роботи включають в себе різноманітні функції та можливості, які спрощують процес пошуку роботи для кандидатів і полегшують пошук кваліфікованих працівників для роботодавців. Ось деякі загальні риси цих сервісів:

– пошук та фільтрація вакансій;
– розміщення резюме та вакансій;
– система рекомендацій;
– управління профілем;
– сповіщення та аналітика;
– система відгуків та рейтингів;
– мобільність;
– безпека та конфіденційність;
– соціальні функції.

Пошук та фільтрація вакансій. Можливість швидкого та зручного

пошуку вакансій за різними критеріями, такими як регіон, галузь, рівень зарплати, тип роботи тощо.

Розміщення резюме та вакансій. Функція для кандидатів розміщувати свої резюме та для роботодавців – вакансії, що спрощує знаходження та відбір потенційних кандидатів чи робочих місць.

Система рекомендацій. Алгоритми, які рекомендують вакансії або кандидатів на основі історії пошуку або попередньої взаємодії користувача з платформою.

Управління профілем. Можливість створювати та редагувати профілі користувачів з деталями про навички, досвід роботи, освіту тощо.

Сповіщення та аналітика. Система сповіщень щодо нових вакансій, відповідей на заявки, а також збір та аналіз даних про активність користувачів.

Система відгуків та рейтингів. Можливість залишати відгуки та оцінки про роботодавців або кандидатів, що допомагає іншим користувачам у прийнятті рішень.

*Мобільність. А*даптація до мобільних платформ зручна для використання на смартфонах та планшетах.

Безпека та конфіденційність. Забезпечення захисту особистої інформації користувачів та даних, які збираються платформою.

Соціальні функції. Можливість зв’язку між кандидатами та роботодавцями через соціальні мережі або чати.

Ці риси є загальними та можуть варіюватися залежно від конкретної платформи та її функціональних можливостей. Однак вони спрямовані на полегшення та оптимізацію процесу пошуку роботи для кандидатів та роботодавців.

У вебдодатках з пошуку роботи можна виявити різний спектр користувачів, які шукають нові робочі можливості чи прагнуть покращити свою кар’єру. Ось кілька ключових груп користувачів, які активно використовують вебдодатки для пошуку роботи:

– роботодавці та підприємства;

– фрілансери та самозайняті особи;
– кар’єрні консультанти та рекрутери;
– стартапи та підприємці;
– професіонали, які шукають можливості для розвитку;
– інші.

Роботодавці та підприємства. Люди, які шукають робочі місця, вакансії або можливості для свого кар’єрного зростання. Це можуть бути студенти, випускники, досвідчені спеціалісти, люди, які прагнуть змінити професію тощо.

Фрілансери та самозайняті особи. Люди, які шукають проєкти чи роботу на фріланс-платформах, зокрема вебдодатки можуть мати функціонал для пошуку не лише повноцінних постійних робочих місць, але і короткострокових проєктів чи фріланс-замовлень.

Кар’єрні консультанти та рекрутери. Професіонали, які працюють у сфері кадрового підбору та консультування по кар’єрному зростанню. Вони також можуть використовувати вебдодатки для підбору кандидатів або надання консультацій.

Стартапи та підприємці. Люди, які створюють свої власні бізнеси та шукають співробітників або співзасновників для розвитку своїх проєктів.

Професіонали, які шукають можливості для розвитку. Люди, які хочуть знайти нові можливості для розвитку своєї кар’єри, отримати нові знання або змінити область своєї діяльності.

Вебсервіси з пошуку роботи можуть відрізнятися за функціональністю, цільовою аудиторією, особливостями та інтерфейсом. Ось деякі відмінності, які можна зустріти між різними вебсервісами з пошуку роботи:

– спеціалізація;
– функціонал;
– цільова аудиторія;
– розмір ринку;
– інтерфейс та користувацький досвід;

– доступність мобільних додатків;

– додаткові можливості.

Спеціалізація. Деякі вебсервіси можуть бути спрямовані на конкретні сфери (наприклад, технології, медицина, маркетинг) чи типи робіт (повна зайнятість, фріланс, тимчасова робота).

Функціонал. Різні сервіси можуть надавати різноманітні функції: від базового пошуку вакансій до розміщення резюме, систем рекомендацій, можливості спілкування між кандидатами та роботодавцями тощо.

Цільова аудиторія. Деякі вебсервіси можуть бути спрямовані на певні групи користувачів: студенти, досвідчені спеціалісти, люди, які шукають роботу на частковий робочий час тощо.

Розмір ринку. Деякі платформи можуть мати широкий географічний охоплення та покривати багато країн або ринків праці, тоді як інші можуть бути спрямовані на певні регіони або місцевість.

Інтерфейс та користувацький досвід. Різні сервіси можуть мати різні інтерфейси та способи візуалізації інформації, що може вплинути на зручність користування та привабливість для користувачів.

Доступність мобільних додатків. Деякі сервіси мають мобільні додатки, які розширюють їх доступність та зручність використання на мобільних пристроях.

Додаткові можливості. Окрім пошуку роботи, деякі сервіси можуть надавати додаткові ресурси, такі як поради щодо пошуку роботи, блоги з корисною інформацією про ринок праці, курси для саморозвитку тощо.

Ці відмінності можуть впливати на те, який вебсервіс з пошуку роботи буде більш відповідати потребам конкретного користувача. Кожен сервіс може мати свої унікальні особливості, які роблять його привабливим для певної аудиторії.

Як результат, актуальним є скрапер з пошуку роботи, який об’єднує функціонал різних існуючих вебсервісів з пошуку роботи, таких як:

– job.ua;

– LinkedIn;
– robota.ua.

Також скрапер повинен враховувати відмінності існуючих сервісів, що можуть бути об’єднані у скрапері з пошуку роботи, що зробить його актуальнішим за існуючі окреми вебсервіси з пошуку роботи.

2.2 Розробка архітектури скраперу з пошуку роботи

Архітектура скраперу буде поділена на 3 частини:
– бекенд, де буде реалізація функціоналу скраперу;
– фронтенд частина, що позволить користуватися бекенд функціоналом через інтерфейс для користувача;
– база даних, де буде зберігатися результати та історія з пошуку роботи скраперу.

Бекенд буде написаний мовою Java. Java є однією з найпопулярніших мов програмування завдяки своїй простоті, надійності та широкому спектру застосування [10]. Це об’єктно-орієнтована мова програмування, що відома своєю портативністю, надійністю та широким застосуванням у різних областях програмування, від веброзробки до розробки мобільних додатків та підприємницьких застосунків [11]. Ось деякі ключові характеристики мови програмування Java:
– об’єктно-орієнтована мова;
– платформонезалежність;
– багатопоточність;
– безпека;
– широке застосування;
– сучасна бібліотека;
– строга система типів;
– робота з великими проєктами.

Об'єктно-орієнтована мова. Java підтримує парадигму об'єктно-орієнтованого програмування, що означає, що програми створюються за допомогою класів та об'єктів, які взаємодіють один з одним.

Платформонезалежність. Код, написаний на Java, може запускатися на будь-якій системі, яка має встановлену відповідну віртуальну машину Java (JVM). Це досягається завдяки концепції "Write Once, Run Anywhere" (пиши один раз, виконуй скрізь).

Багатопоточність. Мова Java підтримує роботу з багатьма потоками виконання, що дозволяє розвивати програми, які одночасно виконують кілька завдань.

Безпека. Java використовує механізми безпеки, включаючи відсічення пам'яті та перевірку меж масивів, щоб запобігти помилкам у виконанні програм.

Широке застосування. Використовується для розробки вебдодатків (наприклад, за допомогою фреймворків Spring, Hibernate), мобільних додатків (за допомогою Android SDK), підприємницьких застосунків, ігор, систем управління базами даних тощо.

Сучасна бібліотека. Java має широкий набір бібліотек та API, що полегшують розробку різних програм та забезпечують багато готових рішень для різних завдань.

Строга система типів. Мова має статичну систему типів, що дозволяє виявляти помилки під час компіляції програми.

Робота з великими проєктами. Ідеально підходить для великих та складних проєктів завдяки своїм можливостям управління пам'яттю, багатопоточності та структурованості коду [12].

Фронтенд частина буде написана на React. Він є потужним інструментом для створення вебінтерфейсів та надає зручність, ефективність та гнучкість у розробці високоякісних вебдодатків. React – це JavaScript бібліотека для створення користувацьких інтерфейсів (UI), розроблена компанією Facebook. React надає зручні та ефективні інструменти для

створення вебінтерфейсів, основаних на компонентах [13]. Ось деякі ключові особливості React:

– компонентний підхід;
– віртуальний DOM;
– JSX;
– односторінкові додатки;
– компонентний стан;
– екосистема та спільнота;
– підтримка мобільних платформ.

Компонентний підхід. React базується на концепції компонентів. Все в React є компонентами – від простих кнопок і текстових полів до складних блоків вебсторінок. Це спрощує розробку, підтримку та перевикористання код.

Віртуальний DOM. React використовує віртуальний DOM для оптимізації швидкості відображення змін в інтерфейсі. Замість безпосередньої модифікації реального DOM, React спочатку вносить зміни до віртуального DOM та потім визначає оптимальний спосіб оновлення реального DOM.

JSX. React використовує JSX (розширений синтаксис JavaScript), що дозволяє вписувати HTML-подібний код безпосередньо в JavaScript. Це спрощує роботу з UI-компонентами та полегшує розуміння коду.

Односторінкові додатки. React часто використовується для створення односторінкових додатків (Single Page Applications – SPA), які забезпечують більш плавність взаємодії користувача з додатком завдяки динамічному оновленню контенту без перезавантаження сторінки.

Компонентний стан. Компоненти в React можуть мати внутрішній стан, що дозволяє зберігати та змінювати дані відображення без прямого впливу на інші елементи.

Екосистема та спільнота. React має широку екосистему з багатьма корисними бібліотеками, такими як Redux для управління станом додатків, React Router для навігації, Axios для HTTP-запитів тощо. Крім того, React має

активну та велику спільноту розробників, яка надає багато ресурсів та підтримки.

Підтримка мобільних платформ. React може бути використаний для розробки мобільних додатків через React Native, що дозволяє використовувати React для створення нативних мобільних додатків для iOS та Android.

Бази даних обрані PostgreSQL. PostgreSQL є потужною СКБД, яка підходить для багатьох різних застосувань, від невеликих проєктів до великих підприємств, завдяки своїм функціональним можливостям, надійності та гнучкості [8]. PostgreSQL (часто скорочується до “Postgres”) – це потужна об’єктно-реляційна система керування базами даних (СКБД), яка відома своєю надійністю, розширюваністю та додатковими можливостями [14]. Ось деякі ключові характеристики PostgreSQL:

– відкритість та безкоштовність: PostgreSQL є вільним програмним забезпеченням з відкритим вихідним кодом, що дозволяє безкоштовно використовувати його та змінювати за необхідності;
– об’єктно-реляційна модель: PostgreSQL підтримує об’єктно-реляційну модель даних, що дозволяє працювати зі складними зв’язками між об’єктами та таблицями в базі даних;
– розширюваність та розширені можливості: PostgreSQL має велику кількість розширень та додаткових модулів, які дозволяють розширювати його можливості та пристосовувати під конкретні потреби;
– підтримка стандартів SQL: PostgreSQL підтримує багато стандартів SQL та має високий рівень сумісності з ANSI SQL;
– транзакції та безпека: підтримується система транзакцій, що дозволяє забезпечити цілісність даних, також є можливість налаштування прав доступу до даних для забезпечення безпеки і конфіденційності;
– широкі можливості індексування та оптимізації: PostgreSQL підтримує різні типи індексів, що дозволяють оптимізувати запити та прискорювати роботу з великими обсягами даних;

– підтримка JSON та робота з нереляційними даними: PostgreSQL надає можливості для роботи з документними типами даних, такими як JSON та JSONB, що дозволяє зберігати та опрацьовувати нереляційні дані;
– масштабованість: PostgreSQL здатний працювати з великими обсягами даних та має можливості для масштабування як по горизонталі, так і по вертикалі.

Комбінація обраних технологій, наведених вище, гарно сполучаються між собою для створення повноцінного та завершеного скраперу з пошуку роботи. Схематично архітектура сервісу буде мати наступний вігляд (див. рис. 2.1).

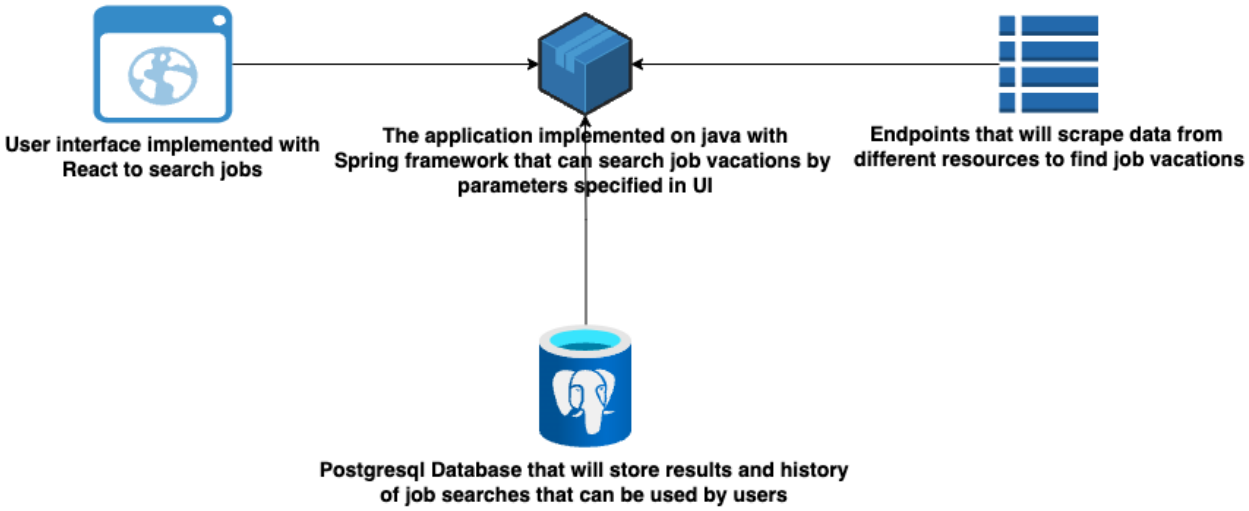

Рисунок 2.1 – Архітектура скраперу з пошуку роботи

2.3 Діаграма варіантів використання

Діаграма варіантів використання (Use Case Diagram) – це один із видів діаграм UML, який використовується для моделювання функціональності системи та взаємодій між системою та її користувачами або іншими зовнішніми сутностями.

Діаграма варіантів використання допомагає визначити, як система

реагує на зовнішні події або взаємодіє з користувачами та іншими системами.

На рисунку 2.2 представлена діаграма варіантів використання.

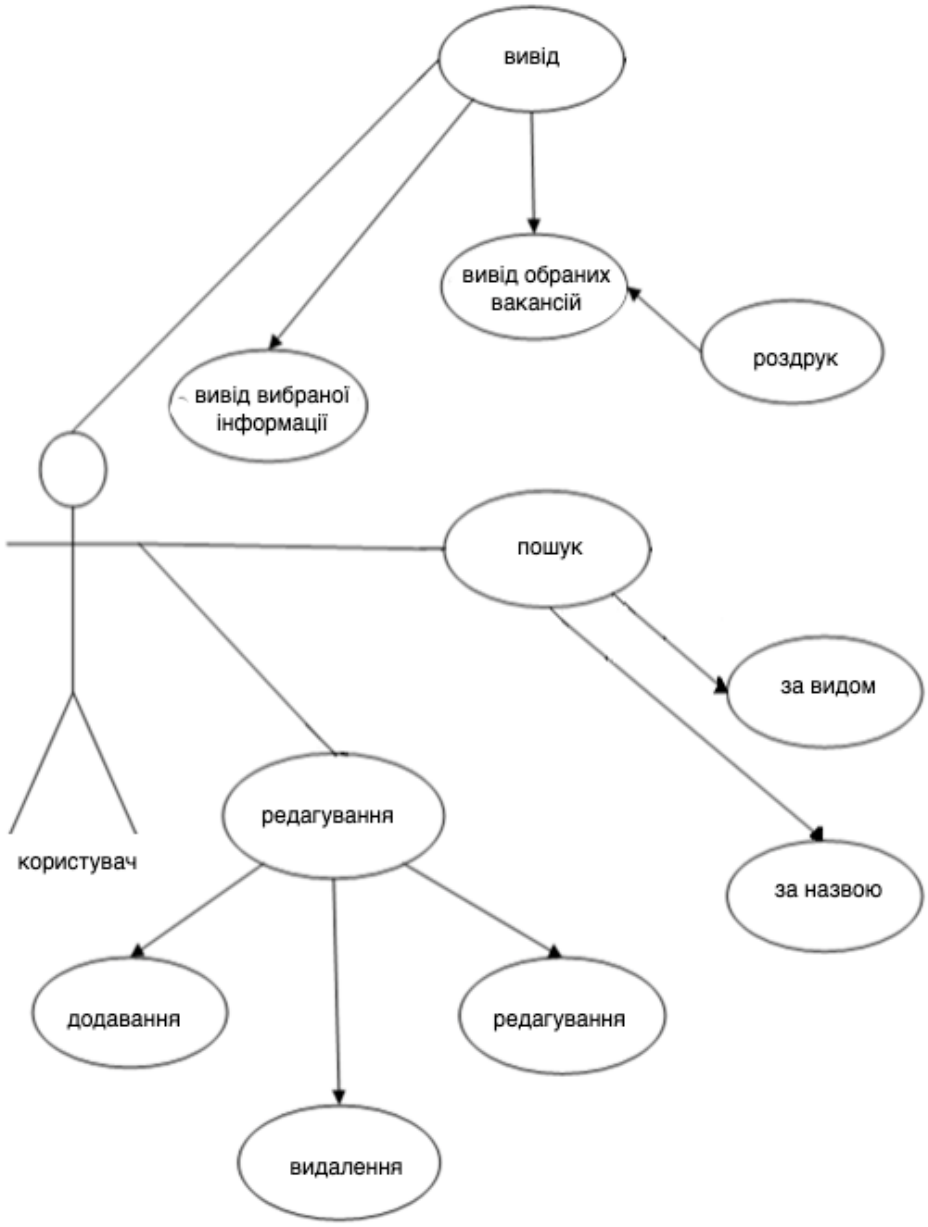

Рисунок 2.2 – Діаграма варіантів використання

Розглянемо основні елементами діаграми варіантів використання.

Актори: актори представляють ролі, які взаємодіють з системою. Це можуть бути користувачі, інші системи, зовнішні програми або будь-які інші сутності, які мають певні вимоги до системи.

Варіанти використання (Use Cases): варіанти використання описують конкретні функціональність системи або дії, які система може виконувати для

задоволення потреб акторів. Кожен варіант використання представляє собою одну конкретну функцію або можливість.

Зв'язки між акторами та варіантами використання: зв'язки показують, які актори взаємодіють з якими варіантами використання. Наприклад, актор може викликати певний варіант використання.

Система: зазвичай на діаграмі є один об'єкт «Система», який представляє саму систему, що розробляється.

Діаграма варіантів використання використовується для:

– опису функціональних вимог до системи;

– визначення взаємодій користувачів із системою;

– уточнення, як система реагує на певні події або сценарії;

– встановлення основних варіантів використання системи.

Варіанти використання можуть бути узагальнені до більш детальних сценаріїв, наприклад, за допомогою діаграм секвенцій або діаграм діяльності.

Ця діаграма допомагає команді розробників і стейкхолдерам краще зрозуміти, як система взаємодіє зі своїми користувачами та іншими системами, і є корисним інструментом для аналізу та специфікації вимог до програмного забезпечення.

2.4 Діаграма послідовності

Діаграма послідовності (Sequence Diagram) – це один із видів діаграм UML (Unified Modeling Language), який використовується для моделювання взаємодій між різними об'єктами або компонентами в системі в часовій послідовності.

Діаграма послідовності допомагає візуалізувати, як об'єкти взаємодіють один з одним у певному сценарії або взаємодій.

На рисунку 2.3 представлена діаграма послідовності.

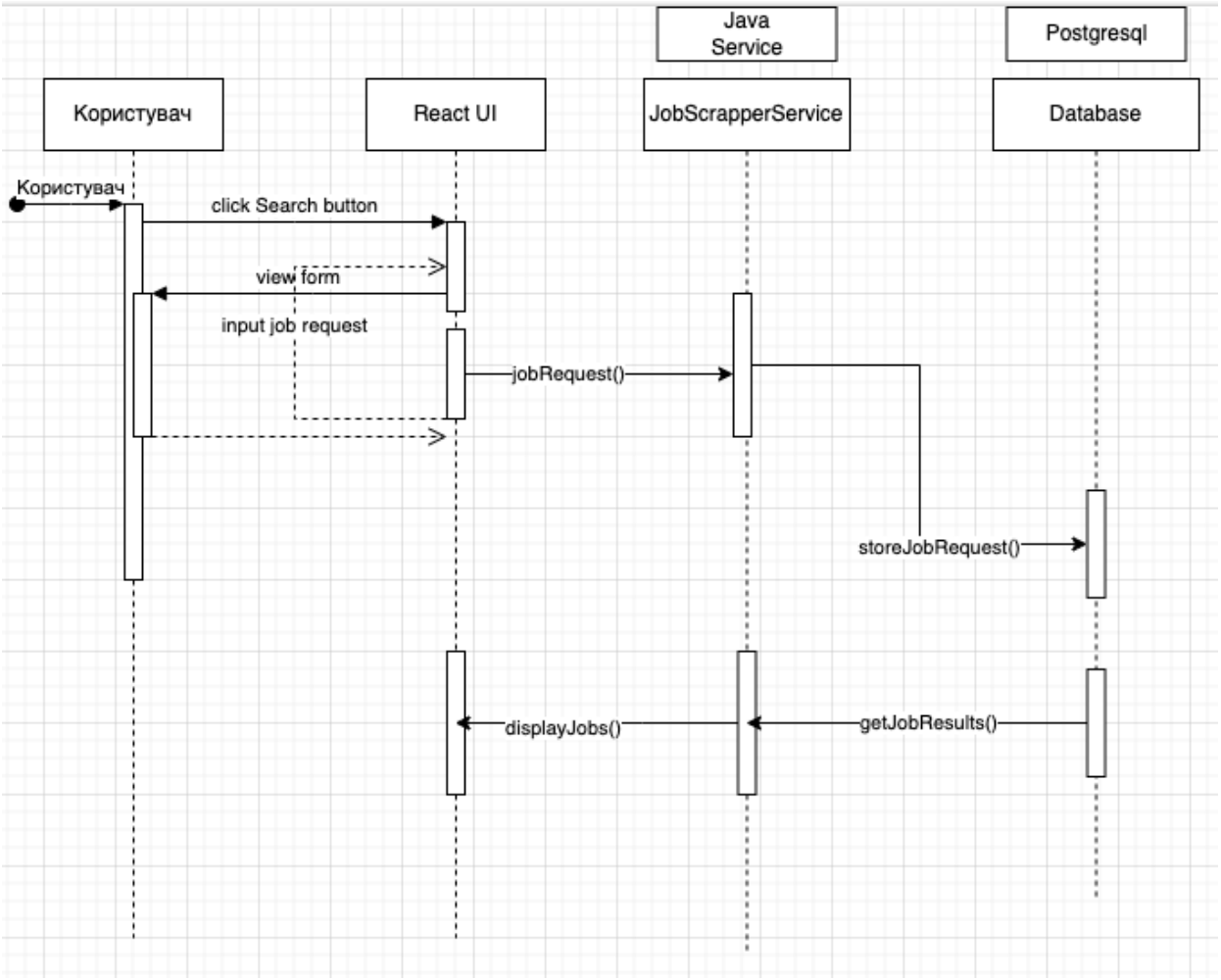

Рисунок 2.3 – Діаграма послідовності

Основними елементами діаграми послідовності представлені нижче.

Об’єкти (Objects): об’єкти або екземпляри класів, які беруть участь у взаємодії, представляються на діаграмі. Кожен об’єкт має ім’я та опціонально – тип (клас або компонент, до якого він належить).

Лінії життя (Lifelines): лінії життя відображають «життя» об’єкта на діаграмі та вказують на той час, коли об’єкт існує та бере участь у взаємодії. Лінії життя зв’язані з об’єктами та показують, коли об’єкт активний або пасивний.

Послідовності повідомлень (Message Sequences): послідовності повідомлень показують взаємодію між об’єктами. Це може бути виклик методу, повідомлення або сповіщення від одного об’єкта іншому. Вони представляються стрілками, які зєднують об’єкти та мають імена та вміст, що

описують, що сталося в процесі взаємодії.

Опціональні фрагменти (Optional Fragments): діаграми послідовності можуть також включати опціональні фрагменти, такі як умовні або циклічні фрагменти, які показують альтернативи або повторювані дії в процесі взаємодії.

Діаграми послідовності використовуються для:

– моделювання взаємодій між об’єктами в системі;
– визначення послідовності дій в рамках конкретного сценарію;
– відображення викликів методів, обміну повідомленнями та інших видів взаємодій між об’єктами;
– аналіз та специфікація функціональності системи.

Діаграми послідовності є потужним інструментом для розуміння та моделювання взаємодій в системі та можуть бути використані для аналізу, проєктування та документування програмних систем.

2.5 Діаграма розгортання

Діаграма розгортання (Deployment Diagram) у рамках мови моделювання UML (Unified Modeling Language) використовується для відтворення фізичної архітектури системи та розташування компонентів програмного забезпечення на апаратному забезпеченні або в інфраструктурі.

Ці діаграми використовуються для:

– моделювання фізичної інфраструктури системи та взаємодії компонентів з апаратним забезпеченням і іншими середовищами;
– аналізу та документування архітектури системи;
– уточнення розташування та конфігурації компонентів системи;
– визначення вимог до інфраструктури та процесу розгортання системи на реальних вузлах.

На рисунку 2.4 зображено діаграму розгортання розроблюваної системи.

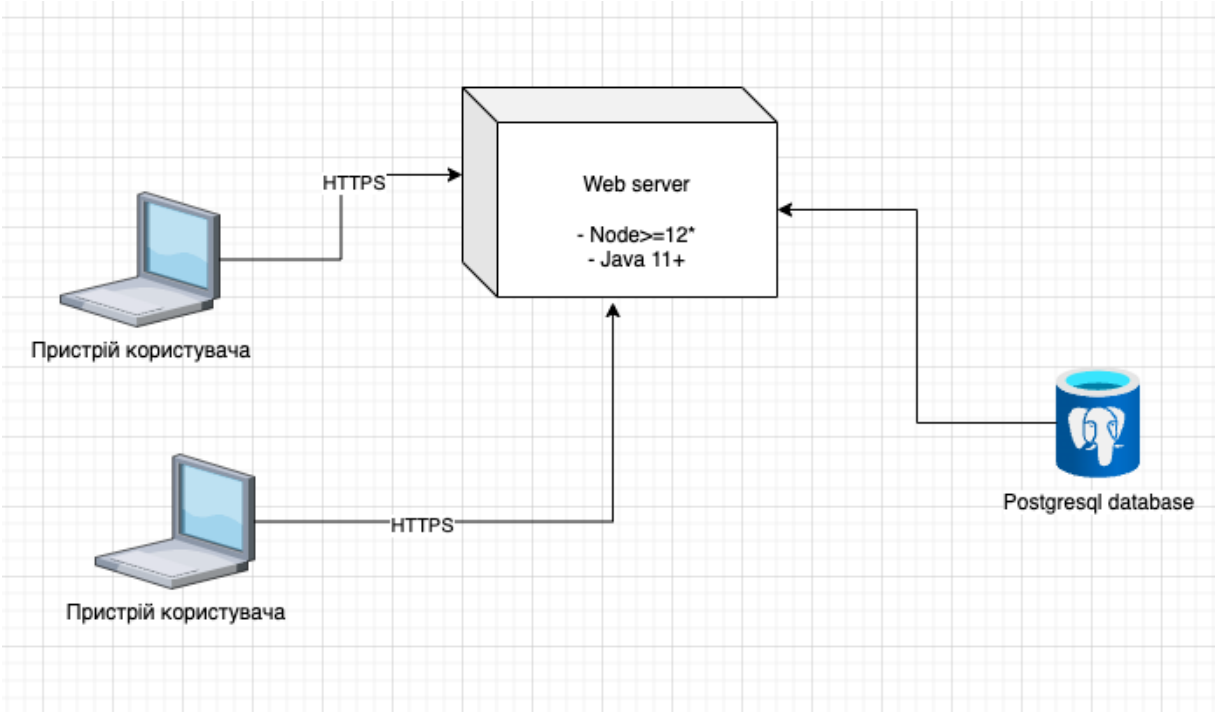

Рисунок 2.4 – Діаграма розгортання

Діаграми розгортання надають можливість командам розробників та системним архітекторам краще зрозуміти взаємодію компонентів системи з фізичною архітектурою та інфраструктурою, сприяючи ефективному проєктуванню та розгортанню програмних систем.

3 РЕАЛІЗАЦІЯ СКРАПЕРУ

3.1 Створення бекенд функціоналу за допомогою Java

Мову програмування для проєкту було обрано Java. Java є однією з найпопулярніших мов програмування, і вона має численні переваги, які сприяють її широкому застосуванню в різних сферах програмування. Ось деякі з основних переваг мови програмування Java (див. рис. 3.1):

– платформено-незалежна;
– об’єктно-орієнтована мова;
– безпека;
– збирання сміття;
– багатобібліотечна підтримка;
– широке співтовариство та підтримка;
– мультипоточність;
– розширені можливості безпеки.

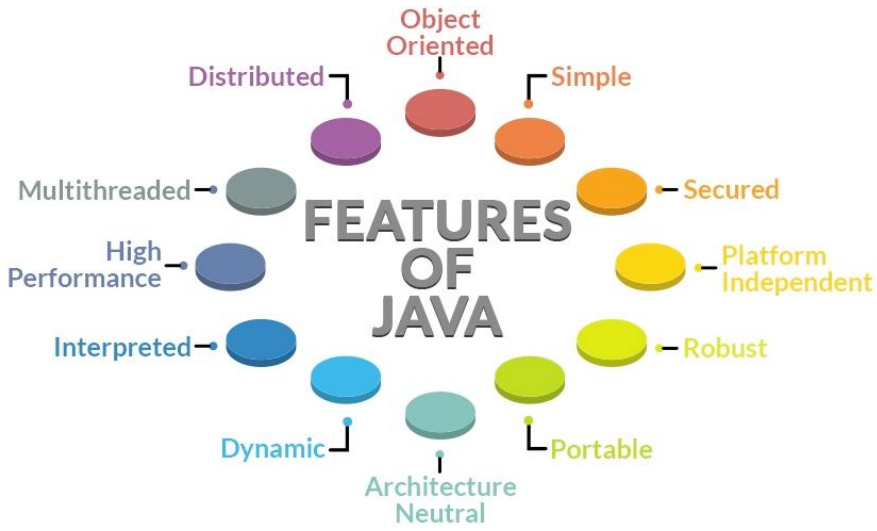

Рисунок 3.1 – Основні переваши роботи на Java

Платформено-незалежна. Java працює на основі концепції “Write Once, Run Anywhere” (WOAR), що означає, що програми, написані на Java, можуть

працювати на будь-якій платформі, яка підтримує віртуальну машину Java (JVM). Це робить Java ідеальним вибором для розробки крос-платформенного програмного забезпечення.

Об'єктно-орієнтована мова. Java базується на об'єктно-орієнтованому програмуванні, що сприяє покращенню модульності, повторному використанню коду та полегшує роботу з складними програмними системами.

Безпека. Java використовує систему безпеки на рівні мови програмування, включаючи механізми перевірки типів, що дозволяє уникнути багатьох типових помилок, таких як витоки пам'яті, дивні вказівники та інші проблеми безпеки.

Збирання сміття. Java має вбудований механізм збирання сміття, який автоматично видаляє непотрібні об'єкти з пам'яті, що спрощує роботу з управлінням пам'яттю та уникненням витоків пам'яті.

Багатобібліотечна підтримка. Java має широкий спектр вбудованих бібліотек для різноманітних завдань, що дозволяє розробникам прискорити процес розробки, використовуючи готові рішення для багатьох типових задач.

Широке співтовариство та підтримка. Java є однією з найпопулярніших мов програмування, тому вона має велике співтовариство розробників. Це означає наявність великої кількості документації, уроків, форумів та бібліотек, які полегшують вивчення та використання мови.

Мультипоточність. Java має вбудовану підтримку мультипоточності, що дозволяє розробникам створювати програми, які можуть одночасно виконувати кілька завдань, що покращує продуктивність та швидкодію програм.

Розширені можливості безпеки. У Java існують інструменти та технології для розробки безпечних додатків, включаючи криптографію, автентифікацію, авторизацію та інші заходи безпеки.

Загалом, Java є потужною мовою програмування, яка має багато переваг і підходить для широкого спектру застосувань, від мобільних додатків до великих корпоративних систем.

Інструмент для розробки скраперу було обрано IntelliJ IDEA. IntelliJ IDEA – це інтегроване середовище розробки (Integrated Development Environment, IDE) для роботи з різними мовами програмування, такими як Java, Kotlin, Scala, Groovy та іншими. Воно розроблене компанією JetBrains і володіє широким набором інструментів для розробки програмного забезпечення. Ось деякі ключові особливості та функції IntelliJ IDEA:

- інтуїтивний інтерфейс користувача: IntelliJ IDEA має добре організований та інтуїтивно зрозумілий інтерфейс, що полегшує розробку програм та навчання користувачів;
- підтримка різних мов програмування: інструмент надає можливість розробки програм на різних мовах, зокрема Java, Kotlin, JavaScript, Python, PHP, HTML, CSS, SQL тощо;
- розширені можливості рефакторингу: IntelliJ IDEA має потужні інструменти для рефакторингу коду, які допомагають оптимізувати та переглядати код для поліпшення його читабельності та продуктивності;
- автоматичне доповнення коду та підказки: IDE надає функцію автодоповнення коду, відстежує помилки та надає підказки, що спрощує процес розробки і допомагає уникнути помилок;
- інтеграція зі сховищами коду: IntelliJ IDEA інтегрується з різними системами контролю версій, такими як Git, Subversion, Mercurial, що дозволяє командам програмістів працювати спільно над проєктами;
- широкий вибір плагінів та розширень: є можливість встановлювати плагіни для розширення функціональності інструменту під конкретні потреби розробки;
- підтримка тестування: IntelliJ IDEA допомагає виконувати автоматичне тестування програм та підтримує роботу з різними фреймворками для тестування;
- інструменти для візуального профілювання та аналізу коду: дозволяє виявляти проблемні місця в коді, оптимізувати його та виявляти

ефективніше роботу програм.

IntelliJ IDEA (див. рис. 3.2) є потужним та універсальним інструментом для розробки програмного забезпечення, що надає широкий набір інструментів та функціональності для комфортної та продуктивної роботи розробників.

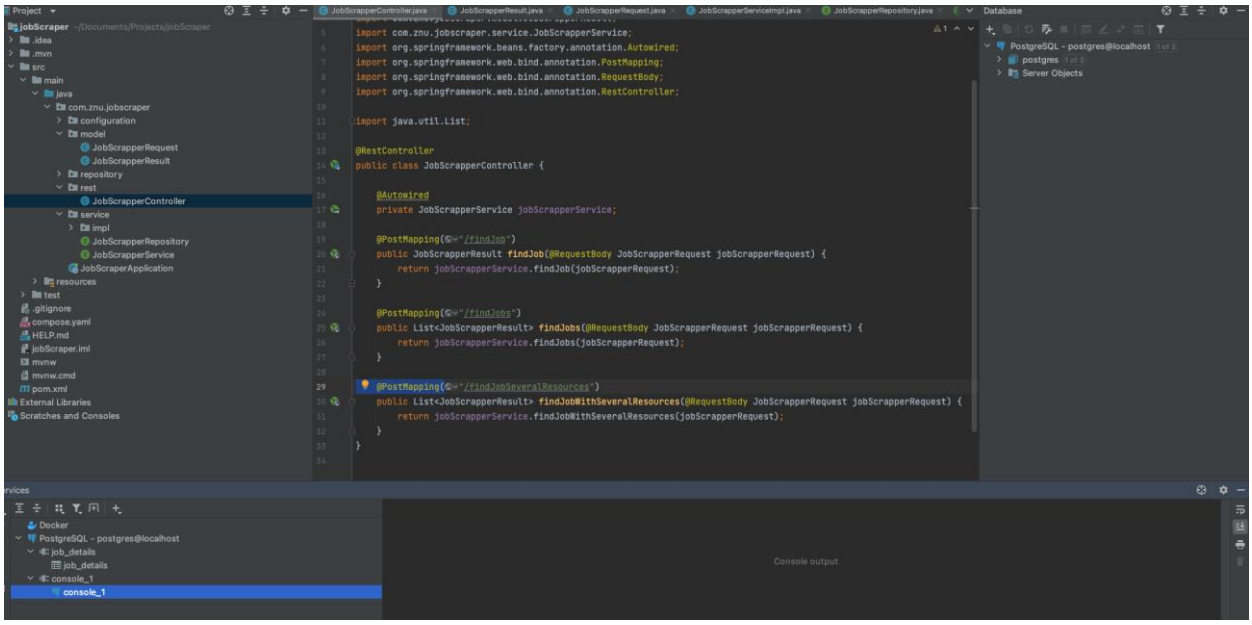

Рисунок 3.2 – Приклад роботи в Intellij IDEA

Для поліпшення над роботу з проєкту будемо використовувати Spring Framework. Spring Framework – це потужний і популярний фреймворк для розробки програмного забезпечення на Java [9] (див. рис. 3.3). Цей фреймворк надає широкий спектр інструментів та можливостей для створення різноманітних додатків – від простих вебдодатків до складних корпоративних рішень. Ось деякі ключові особливості Spring Framework:

– Inversion of Control (IoC) / Dependency Injection (DI): Spring використовує IoC-контейнер для управління об’єктами та їх залежностями (DI дозволяє внедрювати залежності в об’єкти, що полегшує тестування та підтримку коду);
– модульність: фреймворк розділений на різні модулі, що дозволяє розробникам використовувати лише необхідні компоненти для свого

проєкту, зменшуючи навантаження на додаток;

– Spring Boot: це підфреймворк Spring, який надає швидкий спосіб створення самостійних, готових до використання додатків (Spring Boot автоматизує конфігурацію та надає стандартні параметри за замовчуванням для різних компонентів);
– аспекти (Aspects) та AOP (Aspect-Oriented Programming): Spring підтримує AOP, що дозволяє використовувати аспекти для розширення функціональності додатку (це дозволяє виділяти перехоплення певних методів та внедрювати спеціалізовані функції);
– підтримка для розробки вебдодатків: Spring Framework має модулі для розробки вебдодатків, такі як Spring MVC, що дозволяють створювати вебдодатки з підтримкою MVC-архітектури;
– інтеграція з різними технологіями: Spring Framework надає інтеграцію з різними технологіями, такими як Hibernate для роботи з базами даних, Spring Security для забезпечення безпеки, Spring Data для спрощення роботи з базами даних тощо;
– тестування: Spring добре підтримує тестування, надаючи можливості для unit-тестування, інтеграційного тестування та тестування компонентів;

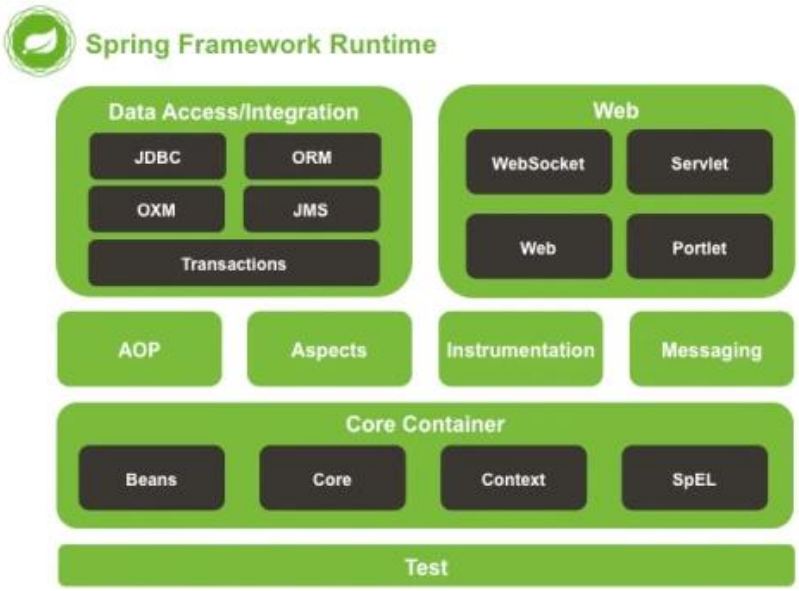

Рисунок 3.3 – Модулі структури фреймворку Spring

Spring Framework забезпечує простий та ефективний спосіб розробки Java-додатків, спрощуючи багато рутинних завдань та надаючи гнучкість у виборі компонентів для власних потреб.

Збірка проєкту будемо робити за допомогою maven. Apache Maven – це інструмент для управління проєктами та збірки програмного забезпечення в середовищі розробки Java. Maven дозволяє автоматизувати процес збірки, залежностей, тестування, документування та розгортання Java-проєктів. Ось декілька ключових понять та функціональностей Maven:

- POM (Project Object Model): Maven використовує файл POM для опису проєкту (POM – це XML-файл, який містить інформацію про проєкт, його залежності, налаштування збірки та плагіни);
- управління залежностями: Maven дозволяє описувати залежності проєкту в файлі POM та автоматично завантажувати відповідні бібліотеки (JAR-файли) з репозиторіїв Maven;
- цикл життя збірки (Build Lifecycle): Maven визначає стандартний цикл життя для збірки проєкту, який складається з фаз (phases), таких як compile, test, package, install, deploy (кожна фаза виконує певні дії в процесі збірки);
- плагіни: Maven підтримує використання плагінів, що дозволяє розширювати можливості і функціональність збірки проєкту, наприклад, виконання додаткових завдань під час збірки;
- конвенція над конфігурацією (Convention over Configuration): Maven використовує стандартні конвенції та структуру каталогів для проєктів (це дозволяє зменшити необхідність конфігурувати кожен проєкт з нуля);
- репозиторії: Maven використовує репозиторії для зберігання та керування бібліотеками, що використовуються в проєкті (Maven має центральний репозиторій та можливість налаштувати власні локальні або віддалені репозиторії).

Maven (див. рис. 3.4) дозволяє розробникам створювати проєкти за

стандартними шаблонами, автоматизувати збірку, керувати залежностями та спрощувати процес розробки великих та складних Java-проєктів.

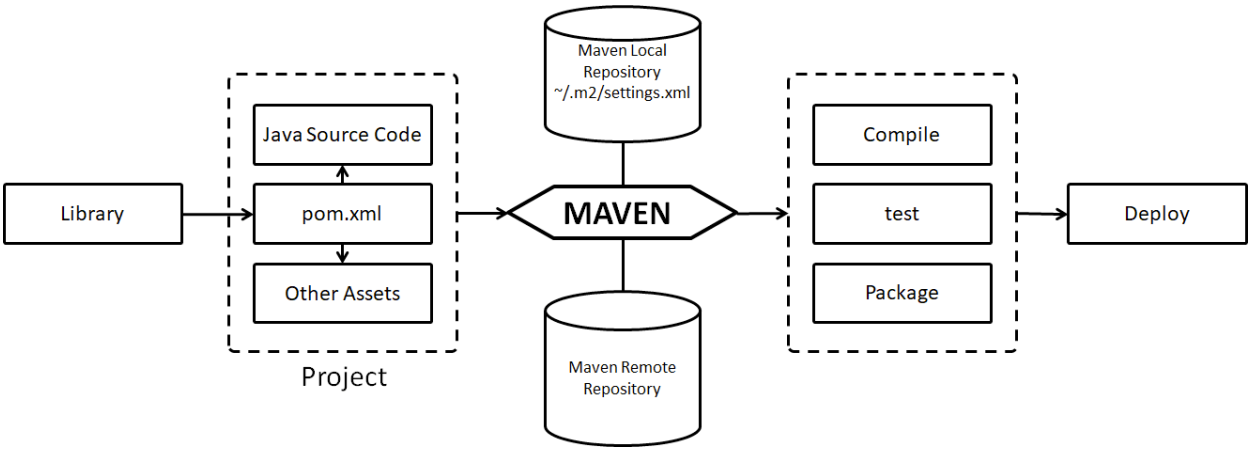

Рисунок 3.4 – Концепція життєвого циклу Maven

Після створення проєкту в Intellij Idea треба структурувати проєкт (див. рис. 3.5). Загальний підхід до створення пакетів полягає в логічному групуванні класів та компонентів вашого проєкту, що допомагає зрозуміти структуру вашого коду та спрощує подальше управління проєктом.

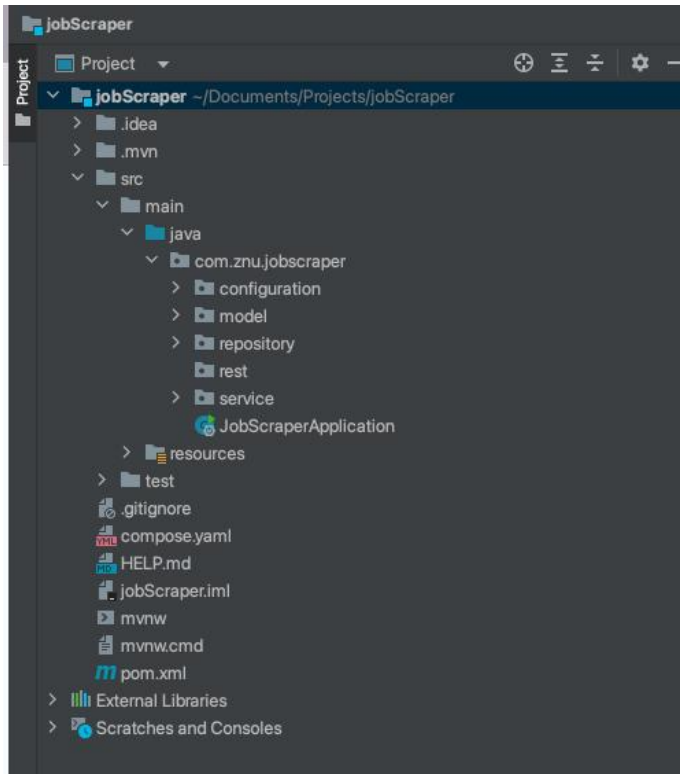

Рисунок 3.5 – Структура проєкту

Створення пакетів в проєктах IntelliJ IDEA має важливе значення для організації та структурування вашого коду. Правильне створення пакетів сприяє чистоті, зручності та ефективності вашого проєкту.

Головна логіка буде заключатись у відправці користувачем запитів на пошук роботи до працюючого скраперу. Скрапер буде обробляти запроси і шукати варіанти з різних вебджерел, та відображати збережені у базі данних скраперу результати (див. рис. 3.6).

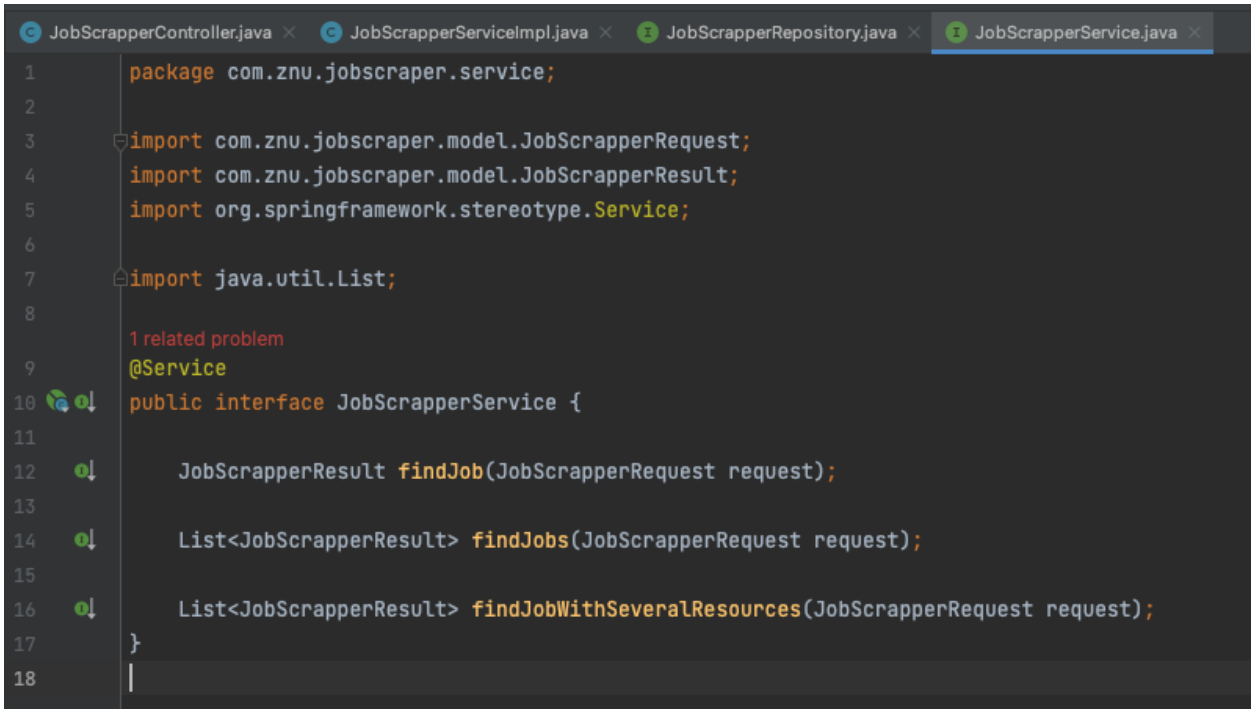

Рисунок 3.6 – Головні ендпоінти з пошуку роботи скраперу

3.2 Створення бази даних за допомогою PostgreSQL

PostgreSQL – це потужна система керування базами даних (СКБД) з відкритим вихідним кодом, яка використовується для зберігання та керування великими обсягами даних. Вона відома своєю надійністю, розширюваністю, а також високим рівнем функціональності та сумісності зі стандартами SQL.

Основні характеристики PostgreSQL:

– відкритий вихідний код: PostgreSQL є вільним програмним

забезпеченням з відкритим вихідним кодом (це означає, що ви можете використовувати, змінювати та розповсюджувати його безкоштовно);

– об’єктно-реляційна модель даних: PostgreSQL підтримує об’єктно-реляційну модель даних, що дозволяє зберігати дані у вигляді таблиць зі зв’язками між ними;
– можливості розширення: велика кількість розширень та додаткових модулів дозволяють розширити функціональність PostgreSQL та адаптувати її під конкретні потреби;
– стандартизований мова запитів: підтримка багатьох стандартів SQL забезпечує сумісність з існуючими інструментами та дозволяє зручно взаємодіяти з базою даних;
– транзакційна безпека: механізми транзакцій дозволяють зберігати цілісність даних, навіть у випадку виникнення помилок під час виконання операцій;
– широкий вибір типів даних: PostgreSQL підтримує багато типів даних, включаючи вбудовані та користувацькі типи, що дозволяє зберігати та обробляти різноманітні дані;
– масштабованість: PostgreSQL здатний працювати з великими обсягами даних та має можливості для масштабування як по горизонталі, так і по вертикалі.

PostgreSQL використовується у різних сферах, від малих проєктів до великих підприємств, завдяки своїм функціональним можливостям, надійності та гнучкості.

За допомогою БД ми будемо зберігати та відібражати результати пошуку скраперу (див. рис. 3.7).

В таблиці job_information ми будемо зберігати з якого ресурсу ми шукали пропозиції з роботи (колонка source), також за допомогою foreign key job_details_fk, яка відноситься до id таблиці job_details, будемо зберігати інформацію з деталей пошуку (колонка job_details). Також будемо знаходити посилання на ресурс з окремої таблиці(job_source), в якій будемо вести перелік інтернет ресурсів, з яких будемо скрейпити дані.

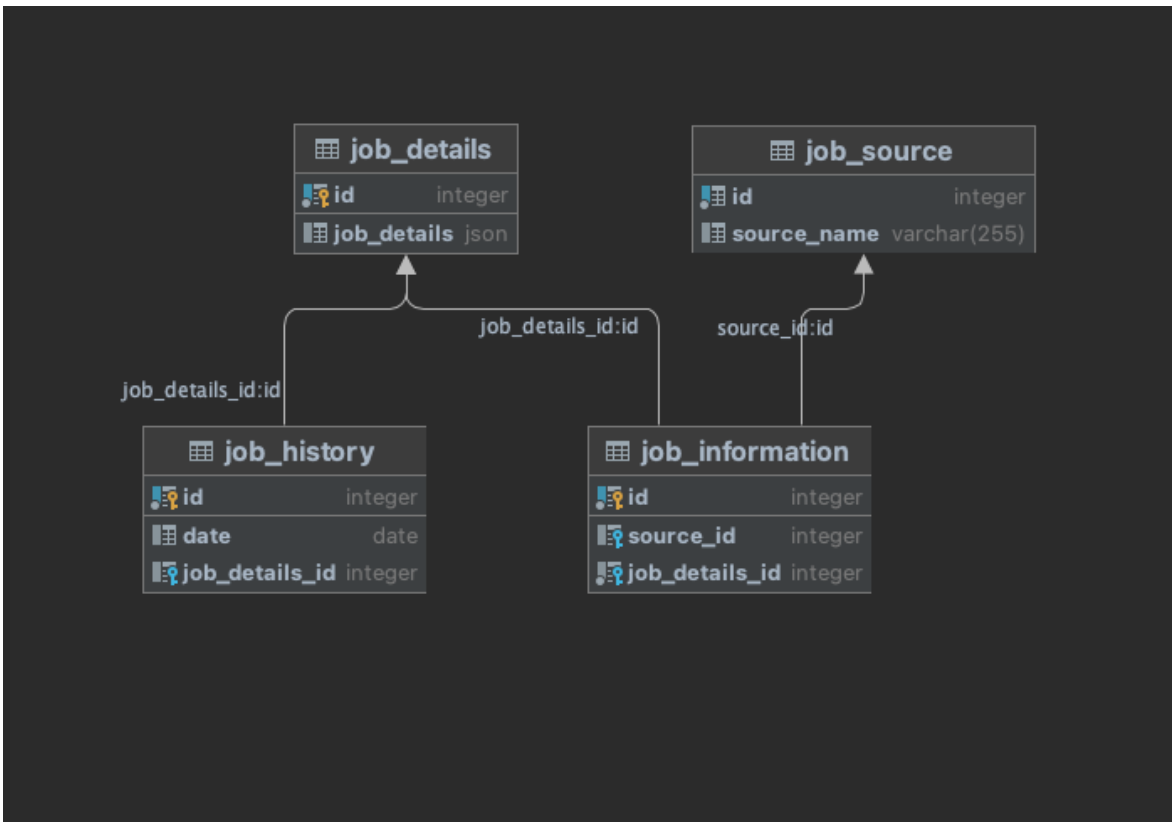

Рисунок 3.7 – Архітектура бази даних

3.3 Розробка UI за допомогою React

React – це бібліотека для створення інтерфейсів користувача у вебдодатках. Вона дозволяє розробникам створювати ефективні, масштабовані та переважно інтерактивні вебінтерфейси. Основна ідея React полягає в тому, що інтерфейс користувача представляється у вигляді компонентів.

Розглянемо основні концепції та можливості React.

Компоненти. Все у React є компонентами. Компонент – це невеликий фрагмент користувацького інтерфейсу, який може бути перевикористаний та вкладений в інші компоненти. Компоненти розділяються на класові та функціональні.

Віртуальний DOM. React використовує віртуальний DOM для оптимізації швидкодії. Замість безпосередньої маніпуляції реальним DOM,

React працює з віртуальною копією DOM, що забезпечує швидку перерендерізацію та оновлення сторінок.

Односторінкові додатки (Single Page Applications, SPA, див. рис. 3.8). React часто використовується для створення SPA, де весь контент завантажується разом з першою сторінкою, а подальша навігація відбувається без перезавантаження сторінок.

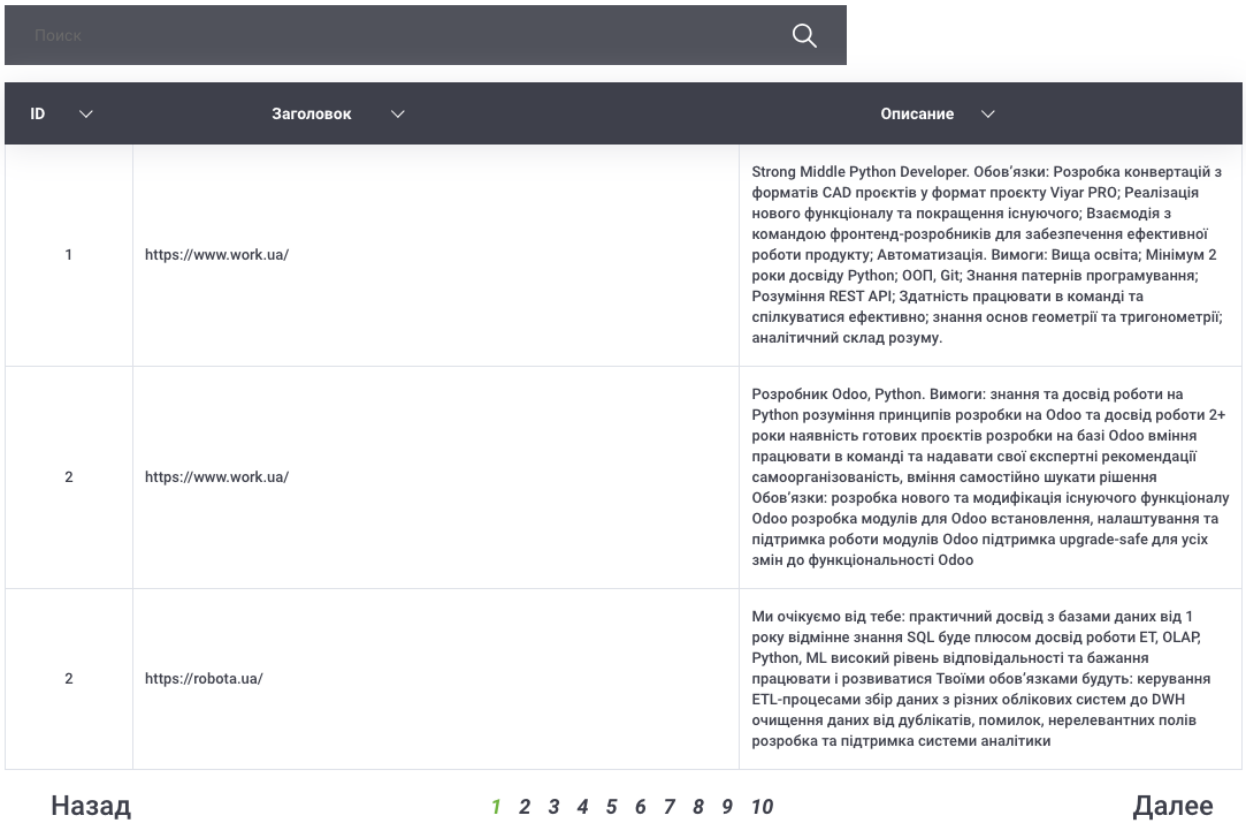

Рисунок 3.8 – UI сторінка результату пошуку

Композиція компонентів. React дозволяє компонувати компоненти разом, щоб створювати складніші структури інтерфейсу.

Стан та властивості (State and Props). Стан використовується для зберігання даних в компонентах, тоді як властивості передаються в компоненти ззовні.

React Hooks. Це функціональні конструкції, які дозволяють функціональним компонентам використовувати стан та інші можливості, які раніше були доступні лише класовим компонентам.

React використовується широкою групою розробників для створення динамічних та інтерактивних вебдодатків. Вона пропонує зручний спосіб організації користувацького інтерфейсу та забезпечує ефективність та швидкодію завдяки своїм оптимізаціям.

ВИСНОВКИ

Предметом дослідження даної роботи фільтрована добірка посилань з пошуку роботи за допомогою скраперу, яка може використовуватися для оцінювання ефективності систем пошуку роботи, та для пошуку роботи.

Визначені показники, які характеризують працевлаштування для кандидата на роботу і роботодавця в автоматизованій системі пошуку роботи за різними наборами факторів, а саме.

Показники для пошуку роботи: регіон; тип зайнятості; галузь/вид діяльності; компанія; дата публікації оголошення роботодавцем; мова; заробітна плата середня; заробітна плата мінімальна; стаж роботи; рівень посади; тип роботи (он-лайн/оф-лайн); вік претендента мінімальний; вік претендента максимальний; режим роботи; термінове оголошення; гаряча пропозиція; кольоровий блок в оголошенні; VSP-компанія; автопоновлення оголошення; платні публікації; банерна реклама в оголошенні. Показники ефективності оголошення про вакансію: закрита/ не закрита вакансія за певний період (тиждень/місяць); час від появи оголошення до закриття вакансії; кількість заяв на вакансію від претендентів за певний період (тиждень/місяць)

Кількісні характеристики процесу пошуку роботи на сайті: дата публікації оголошення роботодавцем; заробітна плата середня; заробітна плата мінімальна; вік претендента мінімальний; вік претендента максимальний.

Характеристики компанії: рейтинг компанії; розмір компанії; рік заснування; локація; назва компанії.

Характеристики виду оголошення: термінове оголошення; гаряча пропозиція; кольоровий блок в оголошенні; VSP-компанія; автопоновлення оголошення; платні публікації; банерна реклама в оголошенні.

ПЕРЕЛІК ПОСИЛАНЬ

1. Work.ua. URL: https://www.work.ua/ (дата звернення: 20.09.2023).
2. Rabota.ua. URL: http://www.rabota.ua/ (дата звернення: 18.09.2023).
3. Jobs.ua. URL: https://jobs.ua/ (дата звернення: 18.09.2023).
4. Dou.ua. URL: https://dou.ua/ (дата звернення: 16.09.2023).
5. LinkedIn. URL: https://linkedin.com/ (дата звернення: 12.09.2023).
6. Джинні. URL: https://djinni.co/ (дата звернення: 18.09.2023).
7. USAJobs. URL: https://www.usajobs.gov/ (дата звернення: 18.09.2023).
8. PostgreSQL. PostgreSQL. URL: https://www.postgresql.org/ (дата звернення: 20.09.2023).
9. Walls C. Spring in Action, 5th ed. Manning Publications Co, 2018. 520 p.
10. Bloch J. Effective Java, 3rd ed. Boston : Addison-Wesley Professional, 2017. 392 p.
11. Niemeyer P., Leuck D., Loy M. Learning Java: An Introduction to Real-World Programming with Java, 6th ed. O'Reilly, 2023. 552 p.
12. Шилдт Г. Java: руководство для начинающих, 7-е издание. Київ : Діалектика-Київ, 2020. 816 с.
13. Scott A. D. JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native, and Electron. O'Reilly Media, 2020. 341 p.
14. Rogov E. PostgreSQL 15 Internals. DMK Press, 2022. 662 p.