

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

УДК 004.9:331.5

«ПОГОДЖЕНО»

Декан факультету

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук

інформаційних технологій

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

2022р.

2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

на тему Програмне забезпечення системи підтримки прийняття рішень для
агенції надання послуг в сфері працевлаштування

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Програмне забезпечення інформаційних систем»

НУБІП України

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми

(науковий ступінь та вчене звання)

(підпис)

(ПІБ)

Керівник магістерської кваліфікаційної роботи

НУБІП України

старший викладач
(науковий ступінь та вчене звання)

(підпис)

Ящук Дар'я Юріївна
(ПІБ)

доц., к.т.н.
(науковий ступінь та вчене звання)

(підпис)

Дудник Алла Олексіївна
(ПІБ)

Виконав

(підпис)

Захарченко Олександр Петрович

(ПІБ студента)

НУБІП України

КИЇВ-2022

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри

Доц., к.т.н.

Голуб Б.Л.

(науковий ступінь, вчене звання)

(підпис)

(ПІБ)

2021 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Захарченку Олександр Петровичу

(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи «Програмне забезпечення системи підтримки прийняття рішень для агенції надання послуг в сфері працевлаштування» затверджена наказом ректора НУБіП України від 01.11.2021р. № 1861"С"

Термін подання завершеної роботи на кафедру

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи

1. Автоматизована система надання послуг в сфері працевлаштування за кордоном.
2. Аналіз даних, що накопичені в процесі надання послуг в сфері працевлаштування.
3. Дані про шуканців роботи та роботодавця

Перелік питань, що підлягають дослідженню

1. Дослідження системи надання послуг в сфері працевлаштування за кордоном
2. Створення моделей предметної області
3. Моделювання та проектування системи аналізу
4. Дослідження використання технологій OLAP та Data Mining

Дата видачі завдання

2021 р.

Керівник магістерської кваліфікаційної роботи

(підпис)

Яшук Д.Ю.

(прізвище та ініціали)

Завдання прийняв до виконання

(підпис)

О. О. Дудник А.О.

(прізвище та ініціали)

Захарченко О.П.

(прізвище та ініціали студента)

ЗМІСТ	
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Загальна характеристика предметної області	9
1.2 Огляд інформаційних джерел та існуючих рішень	9
1.3 Постановка завдання	16
2 МОДЕЛЮВАННЯ СИСТЕМИ	18
2.1 Діаграма прецедентів	18
2.2 Діаграма послідовності	20
2.3 Діаграма активності	24
2.4 Діаграма пакетів	25
2.5 Діаграма компонентів	27
2.6 Діаграма класів	29
3 РОЗРОБКА СИСТЕМИ	31
3.1 Архітектура системи	31
3.2 Логічна модель даних	32
3.3 Фізична модель даних	34
3.4 Алгоритмізація та програмування програмних модулів	35
3.5 Проектування сховища даних	40
3.5.1 Логічна модель сховища даних	40
3.5.2 Фізична модель сховища даних	42
3.6 Створення OLAP-кубу	48
3.7 Реалізація отримання даних за допомогою Data Flow	52

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....	55
4.1 Побудова звітності в середовищі ВІ.....	55
4.2 Розрахунок КРІ.....	58
4.3 Алгоритм дерева рішень.....	59
4.4 Спрощений алгоритм Байєса.....	63
4.5 Алгоритм кластеризації.....	65
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

НУБІП України

ПЗ – програмне забезпечення

БД – база даних

СД – сховище даних

OLAP - онлайн-аналітична обробка

UML - Unified Modeling Language

SSAS - SQL Server Analysis Services

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВСТУП

Актуальність теми. Внаслідок повномасштабної війни росії проти України звичне життя мільйонів українців змінилось, адже вони змушені покинути свої домівки, змінити місце проживання та роботу. З підрахунків Міжнародної організації праці зроблено висновок, що через війну в Україні втрачено майже 5 мільйонів робочих місць. В той же час, за даними Державної служби зайнятості, нині зареєстровано менше безробітних, ніж за минулий рік.

Існує декілька причин, які впливають на зменшення кількості безробітних, а саме: виїзд великої кількості українців за кордон, зменшення підконтрольної території Україною, відправка людей в неоплачувані відпустки на період воєнного стану.

З оприлюдненого звіту Державної служби зайнятості можна побачити, що послугами працевлаштування скористалися 560 тисяч осіб, з яких 495 тисяч зареєстровані як безробітні.

Станом на 1 травня 283 тисячі людей мають статус безробітного. Найбільше безробітних було зареєстровано у таких вікових категоріях:

- 34-44 роки - 29%;
- 45-55 - 28%;
- до 35 років - 26%;
- понад 55 - 17%.

Для рішення проблеми роботодавців в пошуку високо-кваліфікованого працівника та шукача в пошуку роботи виникла потреба у створенні програмного забезпечення для агенції, які надають послуги працевлаштування.

Об'єкт та предмет дослідження. Об'єктом дослідження є процес надання послуг у сфері працевлаштування, що реалізований за допомогою програмного забезпечення. Предметом дослідження виступає – програмне забезпечення системи підтримки прийняття рішень для агенції надання послуг в сфері працевлаштування.

Мета роботи. Мета роботи полягає в підвищенні рівня працевлаштування за рахунок створення системи підтримки прийняття рішень.

Методи дослідження. Для проведення дослідження використовувались стандарти програмні продукти, такі як: ERwin Data Modeler, SQL Server Business Intelligence Development Studio, OLAP- технології та алгоритми Data Mining.

Зміст поставлених завдань. Для досягнення мети необхідно:

- провести системний аналіз;
- сформулювати вимоги до системи підтримки прийняття рішень в сфері працевлаштування;
- розробити та реалізувати структуру програмного забезпечення системи;
- розробити алгоритми для аналізу даних, провести дослідження системи та протестувати систему.

Наукова новизна. У процесі дослідження розроблені методи системи підтримки прийняття рішення агенції з надання послуг в сфері працевлаштування, що дозволяє покращити процес працевлаштування.

Апробація результатів дослідження. За темою магістерської опубліковані тези «Програмне забезпечення системи підтримки прийняття рішень для агенції надання послуг в сфері працевлаштування», що ввійшли до «Збірник матеріалів Міжнародної науково-практичної конференції молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ. ЕКОНОМІКА, ТЕХНІКА, ОСВІТА '2022» (26-27 жовтня 2022 року, НУБіП України, Київ, 2022).

Структура магістерської роботи описана далі.

Кількість сторінок – 72.

Кількість використаних джерел – 21.

Кількість розділів – 4.

У першому розділі розглянуто аналіз предметної області, проведено огляд та аналіз існуючих рішень, та представлена постановка завдання для створеної системи.

У другому розділі – моделювання системи, написані загальні відомості про інструмент моделювання системи, змодельовано діаграми для предметної області та виділені абстракції предметної області.

У третьому розділі побудовано архітектуру системи, побудовано логічну та фізичну модель БД, показано алгоритмізацію та програмування програмних модулів, спроектовано сховище даних, створено OLAP-куб.

У четвертому розділі проведено дослідження за допомогою OLAP-технологій та Data Mining, а саме побудовано звітність за допомогою BI, розраховано KPI та застосовані алгоритми Data Mining дерева рішень, спрощений Байєса та кластеризації.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області

Ріст попиту на послуги працевлаштування за кордоном тісно пов'язане з всесвітнім розвитком інтернет-ресурсів. На сьогоднішній день важко уявити роботу організації без власного веб-сайту. Для бюро, які надає послуги з працевлаштування – це ще один спосіб для швидкого пошуку клієнтів [1].

Важливим кроком для підняття рейтингу та поширення інформації про бюро в інтернеті є створення власного веб-сайту. Сьогодні, майже кожне бюро працевлаштування має свій власний сайт. Це необхідно для розширення свого поля діяльності.

Тому була розроблена веб система надання послуг в сфері працевлаштування за кордоном. Інформація, яка розміщена на сайті надає змогу потенційним клієнтам отримати представлення про бюро та його діяльність. За допомогою інтерфейсу та зручної навігації, клієнт може самостійно вибрати інформацію, в якій він зацікавлений. Розроблений сайт є доповненням до рекламної компанії бюро. Також він забезпечує можливість зворотного зв'язку та інтерактивну роботу з організацією.

Надання послуг працевлаштування за кордоном є складною та відповідальною справою. Воно тісно пов'язане з обробкою даних клієнтів, створенням та зберіганням документації. Врахувавши ці фактори, можна зробити висновок, що є велика необхідність в створенні програмного забезпечення системи підтримки прийняття рішень для агенції надання послуг в сфері працевлаштування для того, щоб зробити роботу агенції ефективнішою.

Огляд інформаційних джерел та існуючих рішень

В ході дослідження був проведений аналіз існуючих рішень.

На першому етапі були розглянуті декілька агентств, які працюють в цій сфері та платформ з пошуку роботи.

GRC.ua – це міжнародний кадровий портал який допомагає знайти роботу та персонал в Україні.

Головна сторінка GRC.ua зображена на рис. 1.1.

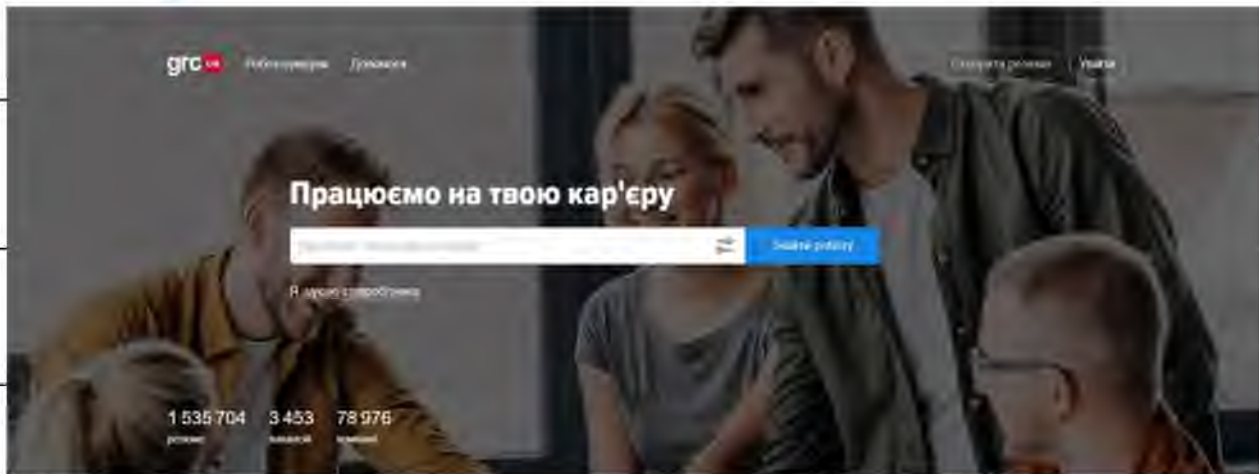


Рис.1.1 Головна сторінка веб-сайту “GRC.ua”

Веб-сайт “GRC.ua” пропонує користувачу для працевлаштування велику кількість міст в Україні. Основними розділами на сайті є:

- роботодавцю;
- допомога пошуку;
- створити резюме;
- увійти;
- контакти.

Вся потрібна інформація, яка цікавить потенційного клієнта легко знаходиться. Сайт має зручну та зрозумілу навігацію, яка полегшує пошук вакансії. На сайті є можливість переглянути напрямки для працевлаштування. На головній сторінці можна переглянути перелік вакансії, які є на сайті та загальну

кількість по кожній з вакансій. Також в категорії вакансії відображається середня заробітна плата.

Кожна з вакансій містить опис, який включає в себе:

- назва вакансії;
- місто;
- заробітна плата;
- досвід роботи;
- освіта;

Також можна відфільтрувати вакансії за такими критеріями:

- категорія;
- місто;
- досвід роботи;
- тип зайнятості;
- заробітна плата.

Для розміщення вакансії роботодавцю потрібно зареєструватись в системі та оплатити послуги, після чого він отримає доступ до бази даних резюме.

Веб-сайт work.ua – сайт пошуку роботи та працівників.

Головна сторінка порталу work.ua зображена на рис. 1.2.

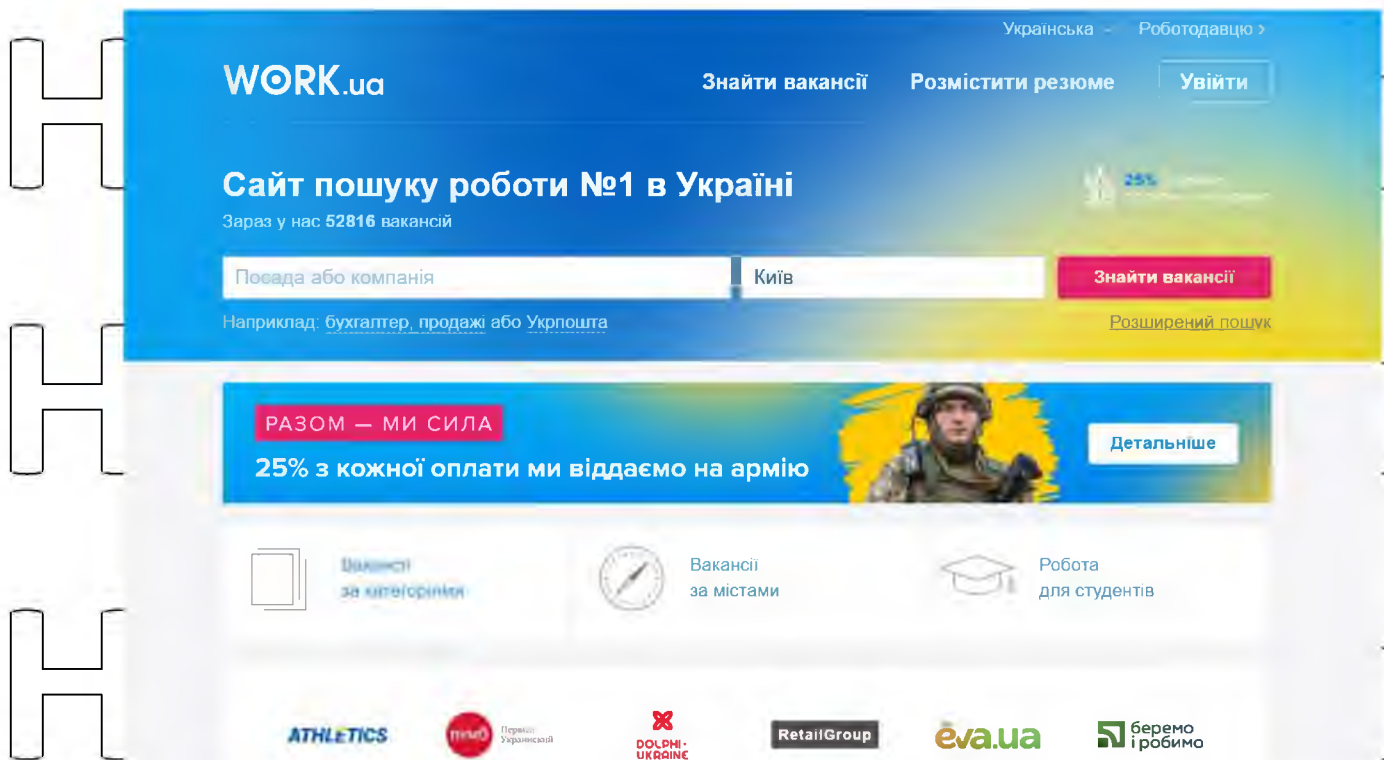


Рис. 1.2 Головна сторінка веб-сайту “work.ua”

Веб-сайт “work.ua” пропонує користувачу пошук роботи в будь якій сфері в залежності від його навичок та знань. Основними розділами на сайті є:

- знайти вакансії
- розмістити резюме;
- увійти;
- роботодавцю.

В шапці сайту є можливість вибрати мову для відображення інформації:

- російська;
- українська;
- англійська;

На головній сторінці розміщується інформація про вакансії за містами та категоріями, робота для студентів, компанії які взаємодіють з сайтом.

Кожна з вакансій містить опис який включає в себе:

- назва вакансії;
- місто;

• заробітна плата;
• досвід роботи;
• освіта;

НУБІП України

Також можна відфільтрувати вакансії за такими критеріями:

• категорія;
• місто;
• досвід роботи;
• тип зайнятості;
• заробітна плата.

НУБІП України

На сайті присутня форма для реєстрація як для шукача так і роботодавця.
Шукач в свою чергу може створити резюме яке потім розміститься на сайті.
Роботодавець може зареєструватись та створити вакансію.

НУБІП України

Кожне з резюме містить опис який включає в себе:

• назва вакансії;
• місто;
• заробітна плата;
• досвід роботи;
• освіта.

НУБІП України

Також можна відфільтрувати вакансії за такими критеріями:

• категорія;
• місто;
• досвід роботи;

НУБІП України

• тип зайнятості;
• заробітна плата;
• стать.

НУБІП України

На сайті також присутня статистика яка зображена за допомогою різних

діаграм. Статистика є по вакансіях, порівняння кількості резюме та вакансії за місяць.

НУБІП України

Проаналізувавши веб-сайти агентств, які працюють в сфері працевлаштування, робимо висновок, що розроблена система повинна містити такі функції:

- зрозуміла та зручна навігація;
- вивід вакансій по критеріям;
- вивід резюме по критеріям;
- форма для замовлення консультації;
- форма реєстрації для шукача та роботодавця;
- форма створення резюме;
- форма розміщення вакансій.
- візуалізація даних.

На другому етапі були проаналізовані останні дослідження та публікації предметної області.

В статті Літвінчук Л. на тему «Світові практики цифровізації надання послуг з працевлаштування» представлені основні тенденції впровадження цифрових технологій в практику надання послуг з працевлаштування. Науково-технічний прогрес та інноваційний прорив в розвитку інформаційних технологій зумовили зміни в усіх сферах суспільного життя. Це стало основою усвідомлення необхідності реінжинірингу економічних і організаційних моделей і процесів, переорієнтації освітньої системи і ринку праці зокрема [2].

Впровадження цифрових технологій для покращання сервісів з працевлаштування в Україні потребує:

- створення цілісної системи вебсервісів з працевлаштування;
- впровадження системи аналізу структурних змін попиту та пропозицій робочої сили на місцевому та регіональному рівнях;
- розроблення «дорожньої карти» успішного працевлаштування на освітньому порталі служби зайнятості з урахуванням індивідуального рівня обізнаності клієнта та його професійного досвіду;

• запровадження уніфікованих форм подання запитів/заяв/звітів та забезпечення системи використання електронного підпису на електронні форми документів та можливостей перенесення їх на інші носії інформації, отримання зворотного зв'язку щодо отримання/прийняття, друку документів тощо [2].

В науковій роботі авторів Н. А. Хрущ, О. С. Корпан, М. В. Желіховська представлені результати дослідження проблеми прийняття управлінських рішень в системі стратегічного управління підприємствами. В роботі наведено обґрунтування процесу прийняття рішень та побудова моделі розробки, прийняття та оцінки управлінського рішення в системі стратегічного управління підприємствами [3].

В ході дослідження систематизовано ознаки класифікації управлінських рішень і запропоновано модель розробки, прийняття та оцінки управлінського рішення. Запропоновані пропозиції можуть бути застосовані при дослідженні багатьох економічних процесів і в той же час є можливість їх подальшого удосконалення [3].

На третьому етапі були розглянуті існуючі системи аналізу ведення бізнесу.

BAS Комплексне управління підприємством (рис.1.3) – комплексним рішенням, що охоплює основні контури управління та обліку. Рішення дозволяє організувати єдину інформаційну систему для управління різними аспектами діяльності підприємств малого і середнього бізнесу. BAS Комплексне управління підприємством дозволяє збільшити продуктивність праці всіх служб підприємства та покращити процеси взаємодії підрозділів [4].

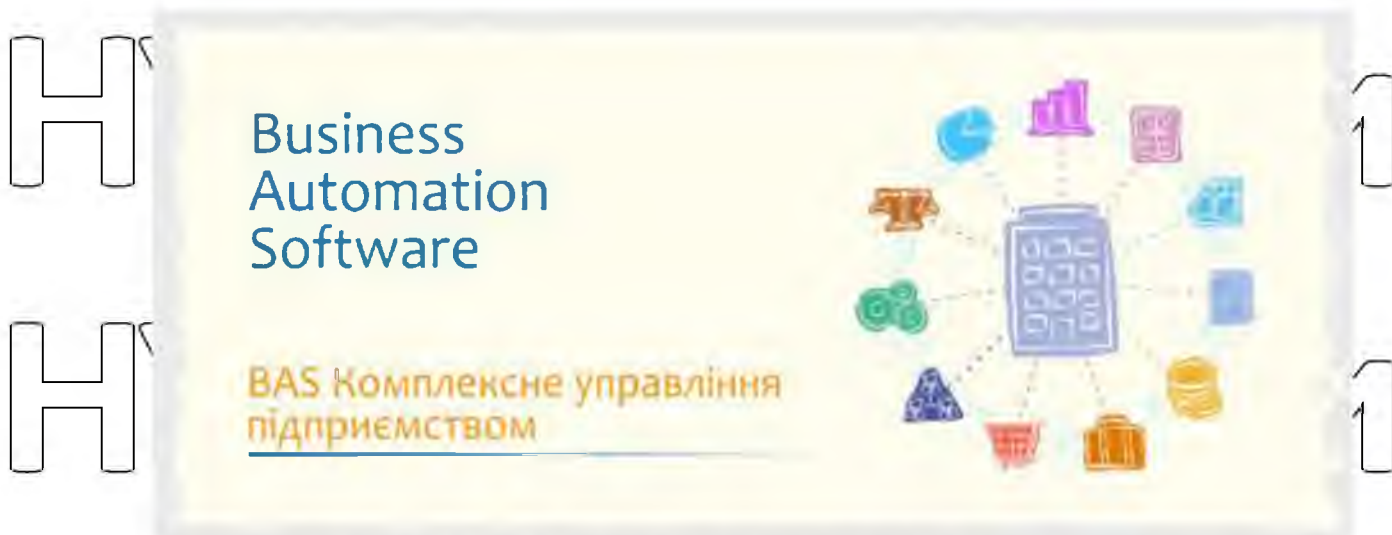


Рис. 1.3 BAS Комплексне управління підприємством

Постановка завдання

Проаналізувавши предмету область та існуючі рішення, доходимо до висновку, що необхідно розробити систему підтримки прийняття рішень, яка дозволить збільшити продуктивність праці всіх служб агентства по працевлаштуванню. Буде організована система для управління різними аспектами діяльності агентства, які якісно та ефективно будуть співпрацювати між собою.

На основі цього постають завдання для розроблюваної системи підтримки прийняття рішень:

- 1) Які вакансії популярні серед кандидатів/фірм?
- 2) На скільки швидко відбувається підбір на посаду?
- 3) Яка кількість кандидатів на одну вакансію?
- 4) Яка кількість реалізованих запитів за період?

Для вирішення цих завдань необхідно розробити систему, яка буде одним з головним джерелом даних для системи підтримки прийняття рішень. До розроблюваної системи, можна сформулювати список вимог, які вона буде виконувати, а саме:

• розробити форму заявки на консультацію. Форма заявки повинна бути зручною та практичною. Форма заявки повинна складатись з 4 полів: «Ваше ім'я», «Електронна адреса», «Номер телефону», та блок з вибором типу консультації. В залежності від вибраного типу консультацію дані будуть відправлятися в конкретну таблицю в БД;

• кожен блок з інформацією про вакансію повинен містити назву вакансії і складатись з таких полів: «Назва вакансії», «Заробітна плата», «Тип зайнятості», «Назва компанії», «Вимоги до досвіду роботи», «Опис вакансії»

• реалізувати зручну навігацію. У користувача не повинно виникати проблем з пошуком інформації, яка його цікавить.

• розробити вивід інформацію по критеріям;

- спроектувати базу даних з можливістю подальшої модифікації;

• розробити систему адміністрування для працівника, роботодавця та шукача роботу;

• швидкість. Розроблюваний сайт повинен бути швидким;

- адаптивність. Сайт повинен коректно відображати інформацію на всіх пристроях;

• розробити можливість додавання, видалення та оновлення інформації про вакансії для роботодавця;

- розробити можливість створення резюме для шукача;

- розробити можливість відклику на вакансії шукачу.

НУБІП України

НУБІП України

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Діаграма прецедентів зображена на рис. 2.1 На даній діаграмі зображено відношення між екторами та прецедентами в системі.

Діаграма прецедентів – це діаграма поведінки, яку використовують для опису дій (варіантів використання) які система може виконувати в співпраці з зовнішніми користувачами система. Дана діаграма показує можливості системи за допомогою об'єктів і сценаріїв використання [5].

Ектор – це об'єкт, який взаємодіє з системою ззовні та ініціює варіант використання. Ектором може виступати як користувач, так і якась система. Один ектор може бути пов'язаний з кількома варіантами використання в системі [5].

Варіанти використання – це основна функціональність робочої системи.

При будівні діаграми прецедентів для системи необхідно дотримуватися таких правил:

- кожен ектор має мати ім'я;
- взаємодія ектора з прецедентом має бути зрозуміло і чітко визначено;
- граничні межі системи повинні охоплювати всі прецеденти [5].

В створеній діаграмі використовується зв'язок включення. У термінах UML зв'язок включення говорить про те, що перший прецедент включає другий.

Розроблена діаграма дозволяє зробити висновок, що в кожного працівника агенції є свої обов'язків. Одним із завдання розроблюваного програмного забезпечення є оптимізація роботи агенції.

В змодельованій діаграмі містяться такі ектори, як:

- «Кандидат»;
- «Фірма»;
- «Аналітик»;
- «HR-менеджер».

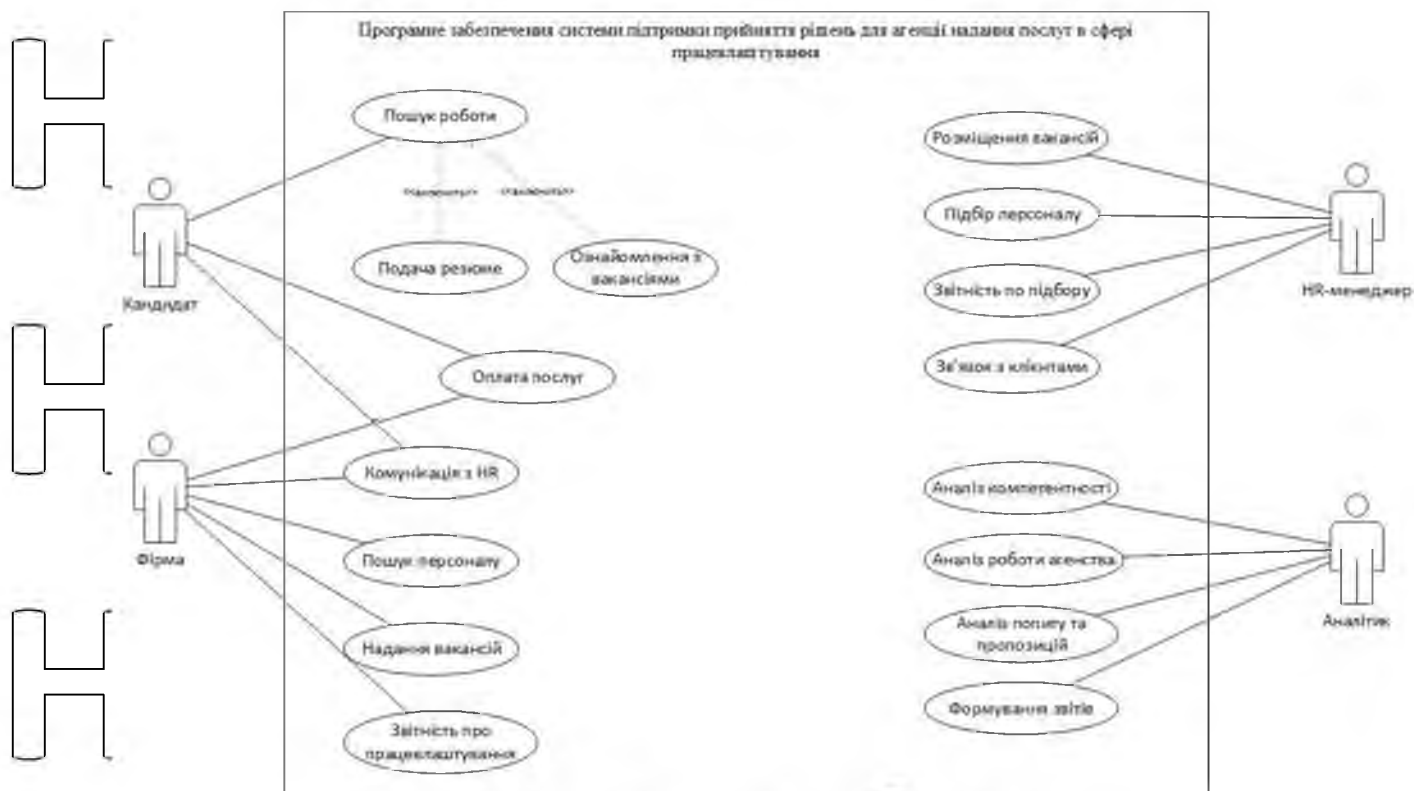


Рис.2.1 Діаграма прецедентів

Для детального розуміння роботи системи нижче представлено сценарій виконання прецеденту «Пошук роботи».

Назва прецеденту використання. «Пошук роботи»

Мета прецеденту використання. клієнт входить на веб-сайт та залишає заявку на пошук роботи.

Оптимістичний сценарій:

- А. Кандидат відвідує веб-сайт.
- Б. Кандидат заповнює форму для зв'язку з HR-менеджер.
- В. HR-менеджер зв'язується з клієнтом і узгоджує деталі.
- Г. Кандидат оплачує послуги агенства та заключає договір з агенством.
- Д. Кандидат відправляє своє резюме HR-менеджеру.
- Е. HR-менеджер розглядає вакансії які підходять кандидату.
- Є. HR-менеджер знаходить вакансію та відправляє резюме фірмі.
- Ж. Фірма погоджується на кандидати.
- З. HR-менеджер відправляє інформацію про співбесіду в фірмі кандидату.
- И. Кандидат проходить співбесіду та працевлаштовується.

Прагматичний сценарій:

Умова 1: Кандидат не має можливості оплатити повний пакет послуг.

A1. HR-менеджер пропонує часткову оплату.

A2. Кандидат робить перший внесок.

A3. HR-менеджер заключає договір з кандидатом про надання послуг.

A4. Кандидат доплатує всю суму.

A5. HR-менеджер видає документи.

Умова 2: Кандидат відмовляється від послуг агенства після оплати послуг.

B1. HR-менеджер телефонує клієнту по телефону та дізнається чому кандидат відмовляється від послуг.

B2. HR-менеджер розриває договір з клієнтом.

B3. HR-менеджер повертає частку коштів клієнту.

B4. HR-менеджер видаляє резюме з бази кандидатів.

2.2 Діаграма послідовності

Діаграма послідовності – це діаграма, головною метою якої є показати в послідовному порядку взаємодію між об'єктами системи. Діаграма послідовності описує в якому саме порядку взаємодіють об'єкти в системі.

Діаграма послідовності зазвичай використовують розробники для моделювання взаємодії між різними об'єктами системи в одному з варіантів використання. Дані діаграми потрібні для ілюстрації того, як різні частини системи взаємодіють між собою в послідовному порядку для виконання певної функції. При створенні цього типу діаграм використовуються ектори.

Ектор – це об'єкт який зображається на діаграмі. Він представляє тип ролі, в якій взаємодіє з системою та іншими об'єктами системи. Ектором в свою чергу може виступати підсистема, інший об'єкт або людина. Об'єкт завжди зображується за межами системи. На діаграмі послідовності може бути зображено декілька об'єктів.

Кожен зображений об'єкт має власну лінію яку називають лінією життя.
Лінія життя потрібен для зображення кожного окремого учасника на діаграмі.
Для розроблюваної системи було побудовано дві діаграми послідовності які зображені на рис.2.2 та рис.2.3.

Діаграма на рис.2.2 показує як відбувається замовлення на пошук кандидата для роботодавця, а на рис.2.3 показує як відбувається пошук роботи.

Діаграма послідовності на рис.2.2 містить такі об'єкти:

- Роботодавець;
- Веб-сайт;
- Працівник агенції.

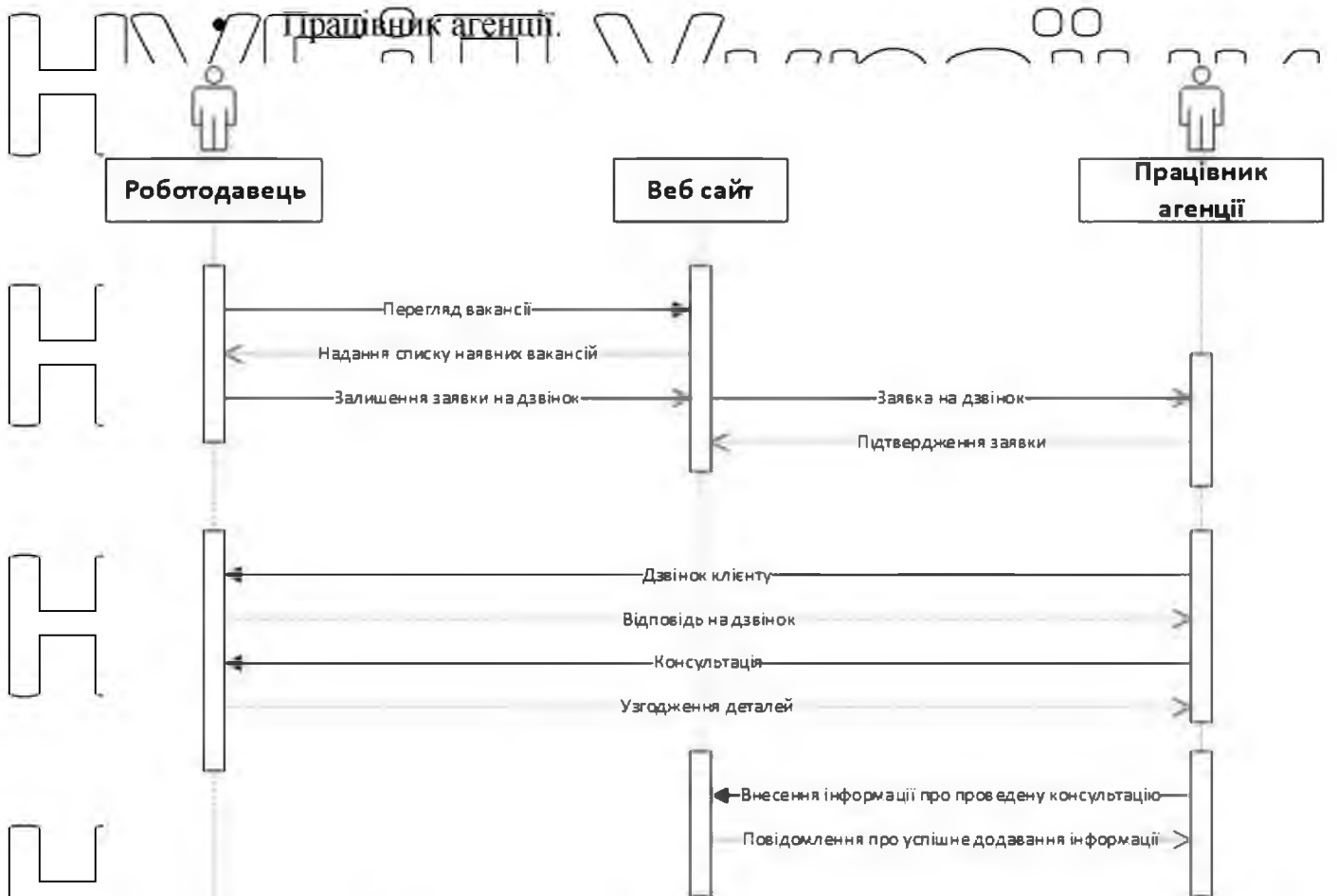


Рис.2.2 Діаграма послідовності (Комунікація з HR)

Об'єкт «Роботодавець» надсилає запит на перегляд резюме які є на сайті до об'єкту «Веб-сайт». Після чого отримує список наявних резюме які є на веб-

сайті. Після перегляду резюме, «Роботодавець» надсилає запит на залишення заявки для дзвінку.

Наступним кроком буде передача заявки на дзвінок від об'єкту «Веб-сайт» до об'єкту «Працівник агенції», в свою чергу об'єкт «Працівник агенції» передає відповідь в вигляді підтвердження заявки.

Після того як заявку було підтверджено об'єкт «Працівник агенції» телефонує об'єкту «Роботодавець» та отримує відповідь на дзвінок, потім проводиться консультація та відбувається узгодження деталей.

Після узгодження деталей об'єкт «Працівник агенції» звертається до об'єкту «Веб-сайт» та вносить інформацію про проведену консультацію.

В кінці об'єкту «Веб-сайт» надсилає повідомлення об'єкту «Працівнику агенції» про успішне додання інформації про проведену консультацію

Діаграма послідовності на рис.2.3 містить такі об'єкти:

- Шукач;
- Веб-сайт;
- Роботодавець.

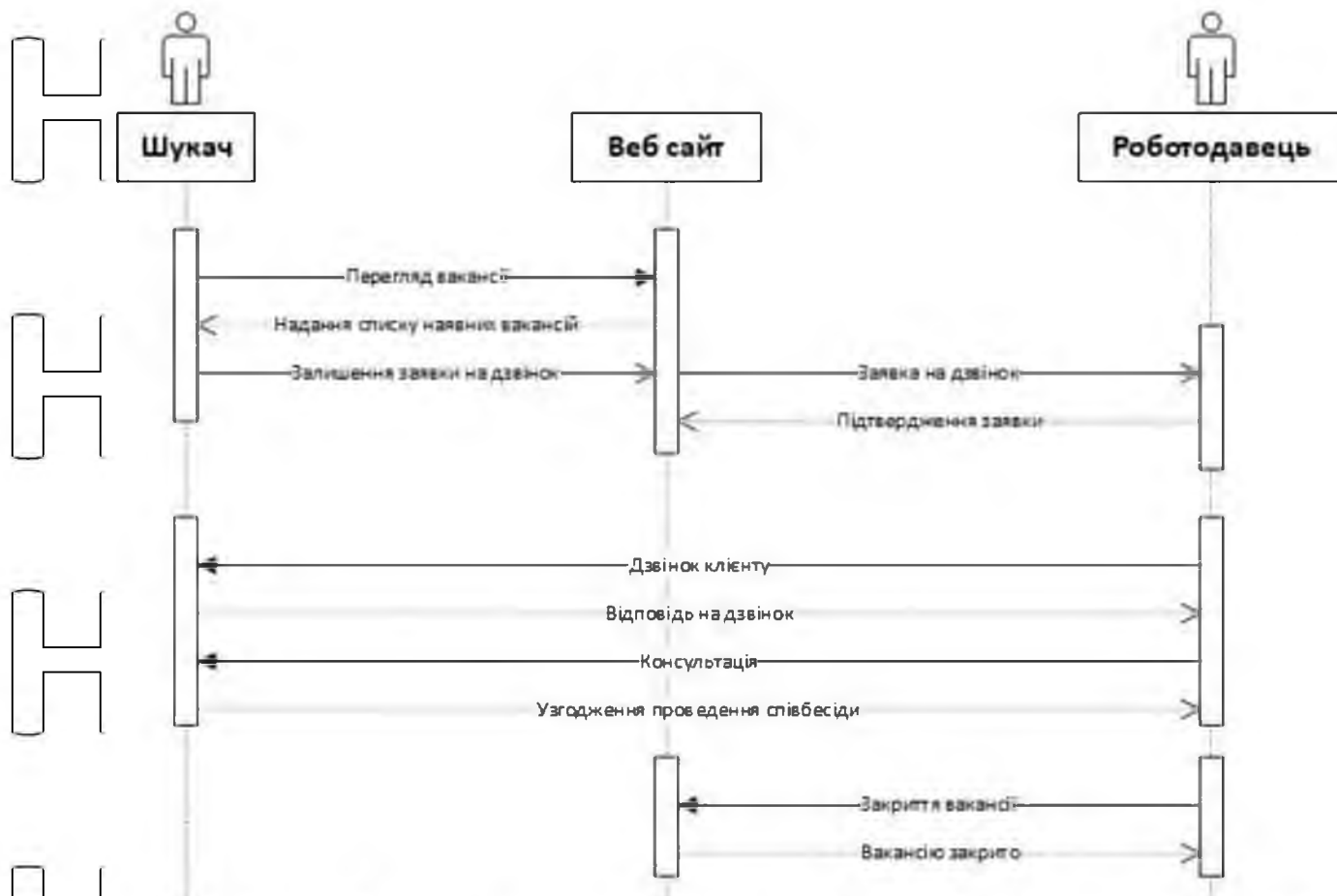


Рис. 2.3 Діаграма послідовності (Пошук роботи)

Об'єкт «Шукач» надсилає запит на перегляд вакансій до об'єкту «Веб-сайт». Після чого отримує список наявних вакансій які є на веб-сайті. Після перегляду та знаходження вакансії, «Шукач» надсилає запит на залишення заявки для дзвінку.

Наступним кроком буде передача заявки на дзвінок від об'єкту «Веб-сайт» до об'єкту «Роботодавець», в свою чергу об'єкт «Роботодавець» передає відповідь в вигляді підтвердження заявки.

Після того як заявку було підтверджено об'єкт «Роботодавець» телефонує об'єкту «Шукач» та отримує відповідь на дзвінок, потім проводиться консультація та відбувається узгодження деталей.

Після узгодження деталей проводиться співбесіда на посаду, якщо кандидат підходить роботодавцю, об'єкт «Роботодавець» звертається до об'єкту «Веб-сайт» та закриває вакансії.

В кінці об'єкт «Веб-сайт» надсилає повідомлення об'єкту «Роботодавець» про успішне закриття вакансії.

2.3 Діаграма активності

Діаграма активності – дана діаграма використовується для зображення послідовності дій. За допомогою цієї діаграми можна показати як відбувається робочий процес від початку до кінця. В цій діаграмі детально відображені шляхи прийняття рішень, які існують в ході виконання певної події.

Даний вид діаграми є дуже корисний в бізнес-моделюванні адже з її допомогою можна детально зобразити процеси які детально беруть участь у підприємницькій діяльності. Також даний вид діаграми можна використати для деталізації різних ситуацій які можуть виникнути в ході виконання певної події

Діаграма активності, зображена на рис.2.5. На змодельованій діаграмі показується як послідовно відбувається комунікація з HR як з сторони клієнта так і системи та як відбувається пошук персоналу як з сторони клієнта так і системи.

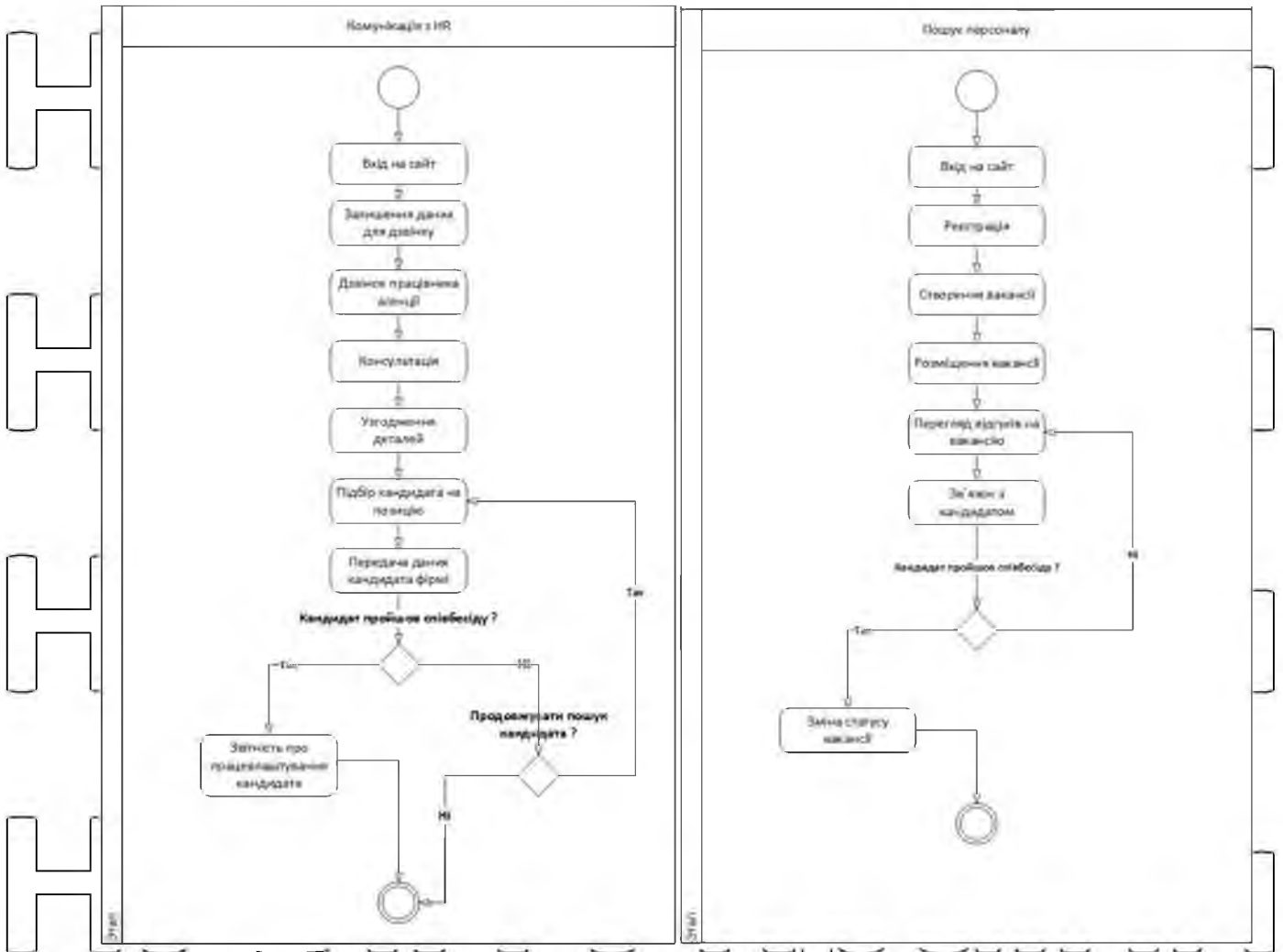


Рис.2.5 Діаграма активності (Комунікація з HR та пошук персоналу)

Діаграма пакетів

Діаграма пакетів – це згруповані пакети моделі з зв'язками між ними.

Дана діаграма в UML використовується для відображення пакетів,

Діаграма пакетів – це набір згрупованих пакетів моделі з визначеними зв'язками між ними. Кожен з пакетів містить в собі під-пакети між якими також встановлені зв'язки. Пакети на створеній діаграмі засновані на логічних відносинах.

Також даний тип діаграми використовується для упорядкування елементів системи та компонентів. Діаграма пакетів зображена на рис.2.6. На даній діаграмі можна виділити три основні пакети.

НУБІП України

- User;
- Server;
- DataBase;

- Analyst;
- Data-Warehouse.

Кожен з перерахованих пакетів містить в собі підпакеги.

Пакет «User» складається таких підпакетів:

- відкритий підпакет «UserGUI»;
- закритий підпакет«UserNetwork».

НУБІП України

Пакет «Server» складається таких підпакетів:

- закритий підпакет«ServerNetwork».
- відкритий підпакет «ServerGUI»;
- відкритий підпакет «Files»;

НУБІП України

- підпакет HTML;
- підпакет CSS;
- підпакет SCRIPT

Пакет «DataBase» складається таких підпакетів:

- закритий підпакет «DataBase access».
- відкритий підпакет «DataBaseGUI»;
- відкритий підпакет «DBRequest»

НУБІП України

Пакет «DataWarehouse» складається таких підпакетів:

- закритий підпакет «DataWarehouse access».
- відкритий підпакет «DWRequest».

НУБІП України

НУБІП України

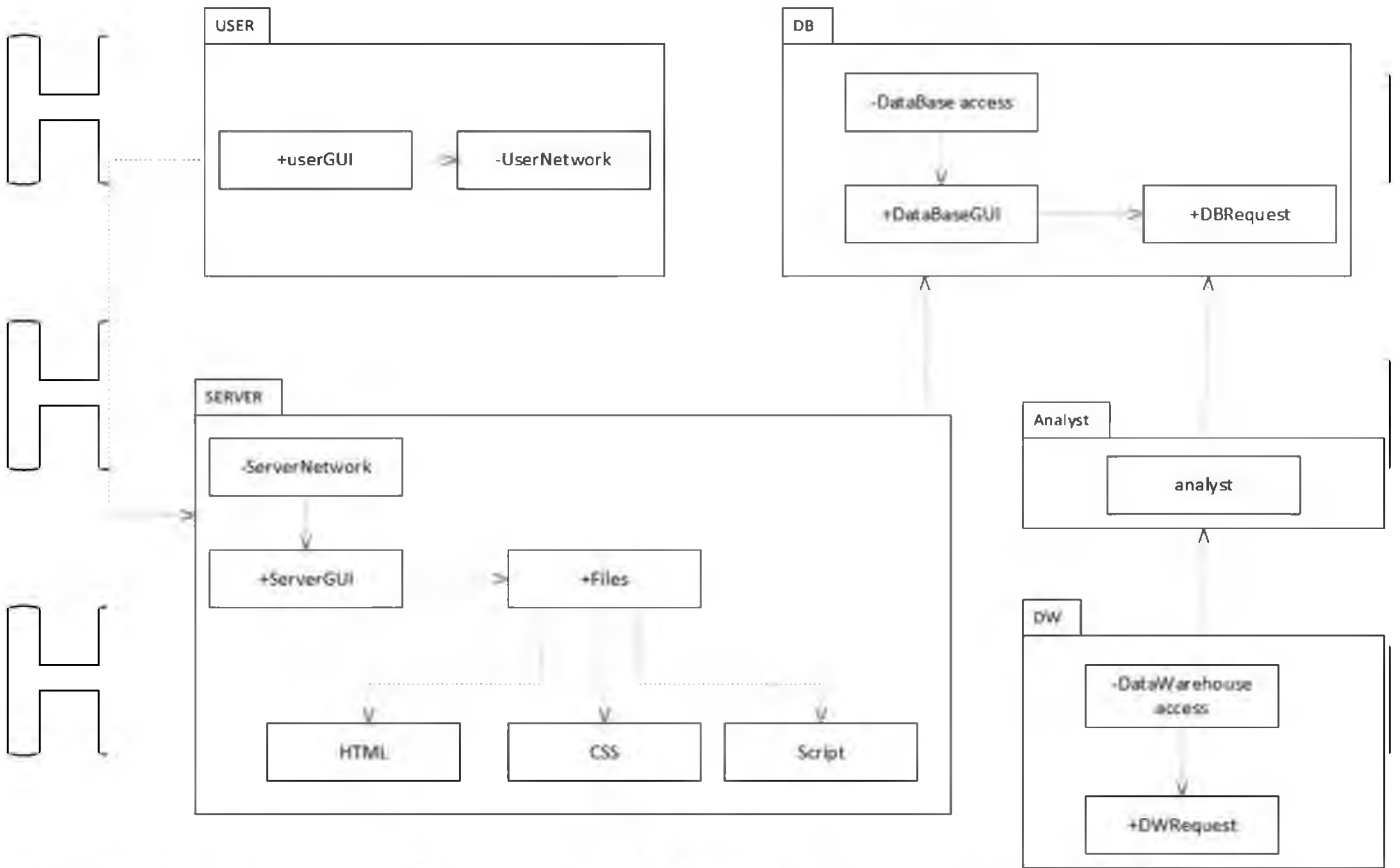


Рис. 2.6. Діаграма пакетів

Діаграма компонентів

Діаграма компонентів – це діаграма яка зазвичай використовується при моделюванні великих систем. Для зручності великі системи розбиваються на керовані системи, в цьому випадку доцільно використати даний тип діаграми. При створенні діаграми описуються компоненти які використовуються для виконання функції а не функціональні можливості системи.

Компонент – це елемент системи. Це частина яка відіграє важливу роль під час виконання та являється логічним елементом. Компонент надає набір інтерфейсів, які реалізує. Декілька компонентів можуть показати всю систему.

Розроблену діаграму компонентів зображено на рис.2.7.

Все починається з компонента «User» який зв'язується з компонентом «Server» завдяки інтерфейсу. В своє чергу компонент сервер складається з двох основних компонентів «Front-end» та «Back-end».

«Front-end» містить в собі:

- `index.php` – вигляд сторінки;
- `style.css` – стилі до коректного відображення дизайну;
- `script.js` – скрипт для роботи.

«Back-end» містить в собі:

- `connect.php` – підключення до бази даних та сховища даних;
- `analyst` – функції аналізу;
- `function.php` – функції для роботи з БД;
- `requestDB` – запит на додавання до БД;
- `requestDW` – запит на додавання до СД;
- `mySQLDB` – база даних;
- `mySQLDW` – сховище даних.

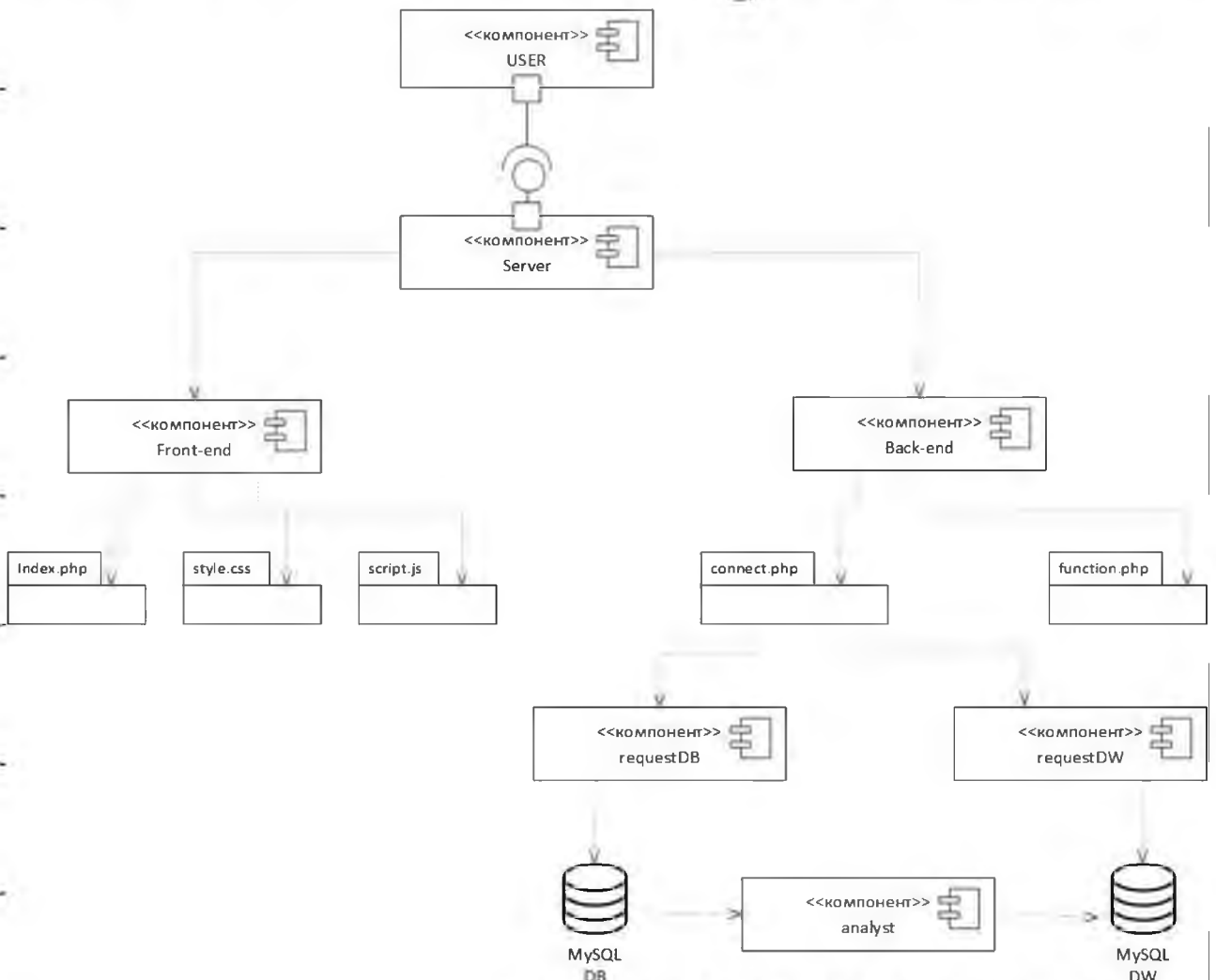


Рис 2.7. Діаграма компонентів

Діаграма класів

При моделюванні ПЗ класи грають незамінну роль. Кожен з створених класів позначає поведінку, структуру об'єкта, зображуються у вигляді прямокутника та складаються з трьох частин. Ці три частини поділяються на розділи: назва класу, атрибути, методи.

Перший розділ містить в собі тільки ім'я класу. Другий розділ містить в собі всі атрибути які має клас. Третій розділ містить всі операції які виконує клас.

Створена діаграма класів зображена на рис.2.8. На даній діаграмі були створені класи:

- agency_employer;
- web_site;
- consultation_order_form;;
- consultation;
- consultation_searcher;
- consultation_employer;
- employer;
- searcher;

Також були встановлені зв'язки між класами:

- Між класами «Consultation» та «Consultation_order_form» встановлено зв'язок композиції;
- Між класами «Web-site» та «Consultation_order_form» встановлено зв'язок агрегації;
- Між класами «Consultation_work», «Consultation_tourism», «Consultation_insurance» та «Consultation» встановлено зв'язок наслідування;

Зв'язок залежності встановлено між класами «agency_employers» та «Consultation», «searcher» та «Consultation order form», «employer» та «Consultation order form».

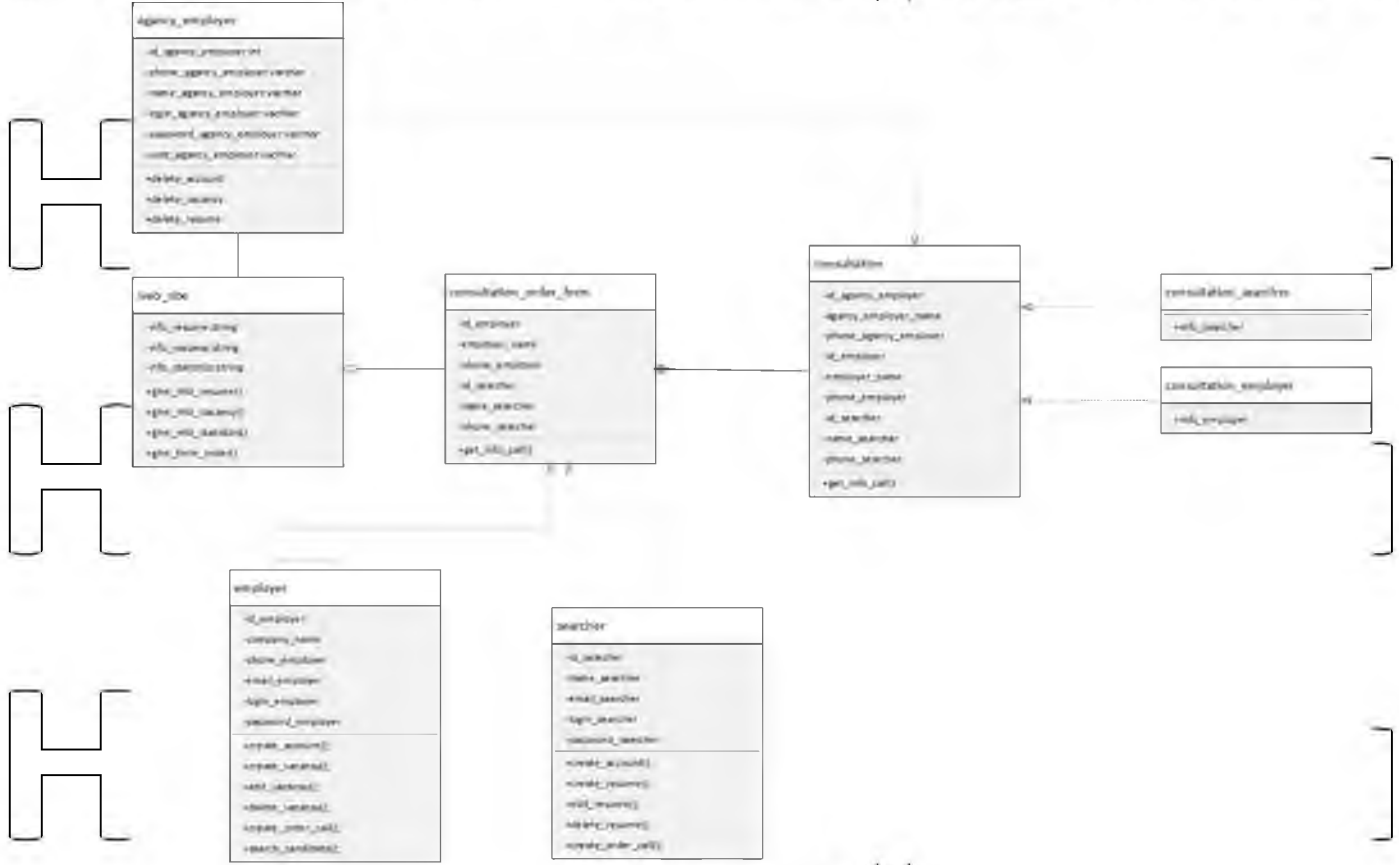


Рис.2.8 Діаграма класів

НУБІП України
НУБІП України
НУБІП України

3 РОЗРОБКА СИСТЕМИ

1 Архітектура системи

Наступним етапом було розроблено архітектуру системи (рис.3.1), вона представлена за допомогою діаграми розгортання. На діаграмі зображено як розподілені компоненти системи по вузлах та маршрути передачі інформації між вузлами системи. Вузлом може виступати як середовище виконання, так і апаратні пристрої.

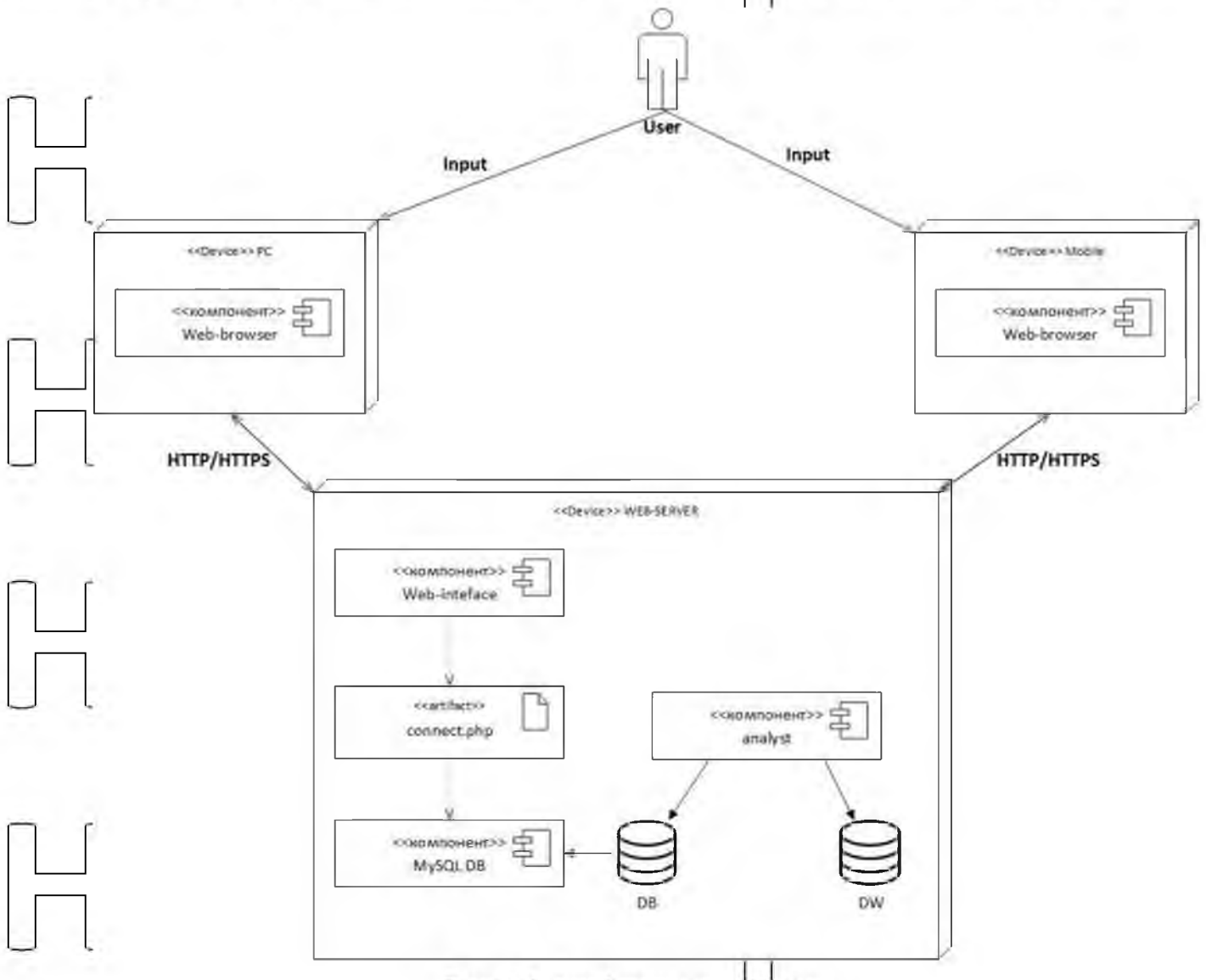


Рис.3.1 Архітектура системи

Система містить три вузла:

- «device» pc;
- «device» mobile;

- «device» web-server;

Кожен з вузлів містить в собі компоненти. Вузол «device» pc та «device» mobile містять в собі компонент «Web browser»;

Вузол «device» складається з двох компонентів, об'єкту та бази даних:

- web-interface - компонент;
- mySQL-DB - компонент;
- «artifact» connect.php – об'єкт;
- Data base – база даних;
- Data Warehouse - сховище даних;
- Analyst – компонент.

Логічна модель даних

Для розробки логічної моделі даних застосовується Erwin.

Erwin – це інструмент який використовують для моделювання бізнес-процесів. Він забезпечує інтеграцію між моделями високого рівня та моделями низького рівня. Моделі високого рівня розробляються на етапі початкового проектування системи, а моделі низького рівня розробляються на етапі програмування та тестування [5].

Erwin може оптимізувати модель відповідно з фізичними характеристиками розроблюваної БД. На відміну від своїх аналогів підтримує узгодження між логічною та фізичною схемою та може здійснювати перетворення логічних конструкції в їх реалізацію на фізичному рівні [5].

Можна виділити дві категорії моделі даних які створюються за допомогою ERwin:

- логічна модель,
- фізична модель.

Логічна модель даних – це представлення моделі бази даних яка ще не пов'язана з якоюсь конкретною СУБД. В логічній моделі БД зображають основні сутності та визначають які зв'язки встановлені між зображеними сутностями [5].

Логічна модель системи зображена на рис.3.2.

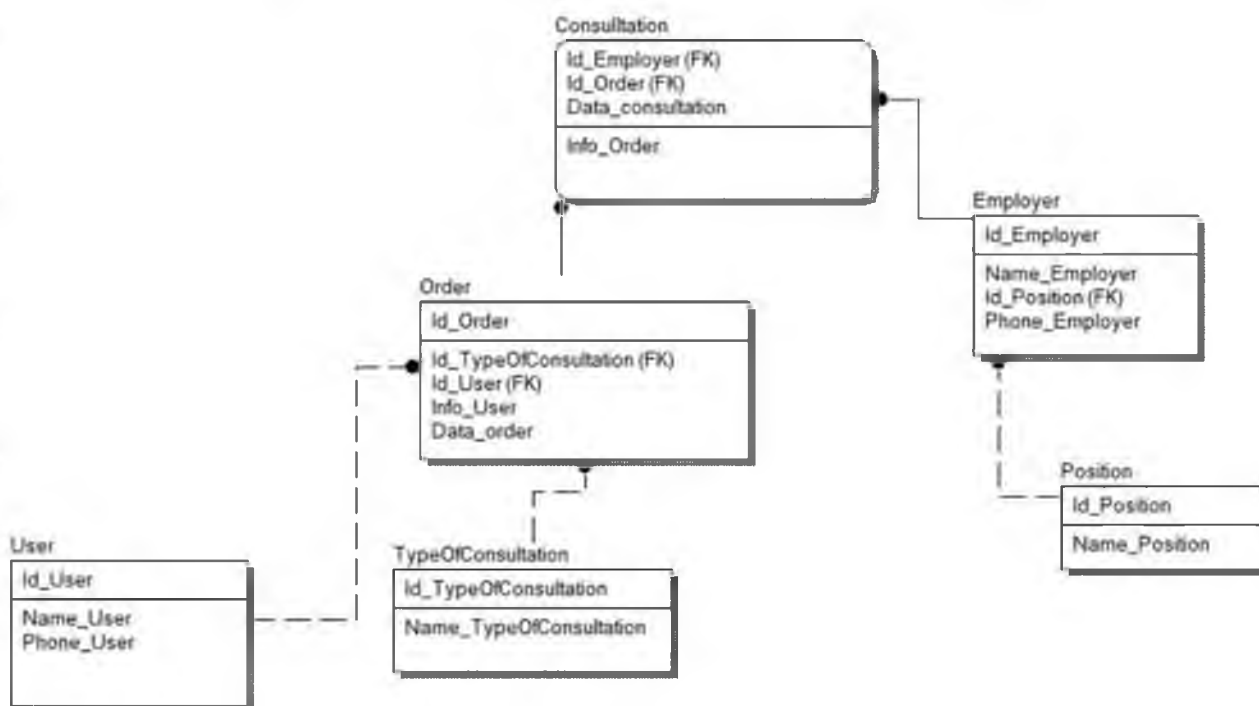


Рис.3.2 Логічна модель БД

Спроектвана логічна модель складатиметься з таких сутностей:

- «User» містить інформацію про клієнта;
- «Order» містить інформацію про замовлену консультацію;
- «TypeOfConsultation» містить інформацію про тип консультації;
- «Consultation» містить інформацію про консультацію та дату проведення консультації
- «Employer» містить інформацію про працівника;
- «Position» містить інформацію про посаду яку займає працівник.

На діаграмі можна побачити два типи зв'язків: ідентифікуючі та неідентифікуючі.

Неідентифікуючі зв'язки встановлені між такими сутностями:

- «User» та «Order»;
- «TypeOfConsultation» та «Order»;

- «Position» та «Employer».

Ідентифікуючі зв'язки встановлені між сутностями:

- «Order» та «Consultation».
- «Employer» та «Consultation».

Фізична модель даних

Джерелом даних системи підтримки прийняття рішень є оперативна база даних (рис. 3.3). Для створення фізичної моделі БД було вибрано MySQL. На даний час MySQL є однією з найпопулярніших БД для веб-додатків. Майже всі веб-ресурси можуть працювати з MySQL. База даних MySQL є простою та дуже гнучкою у використанні.

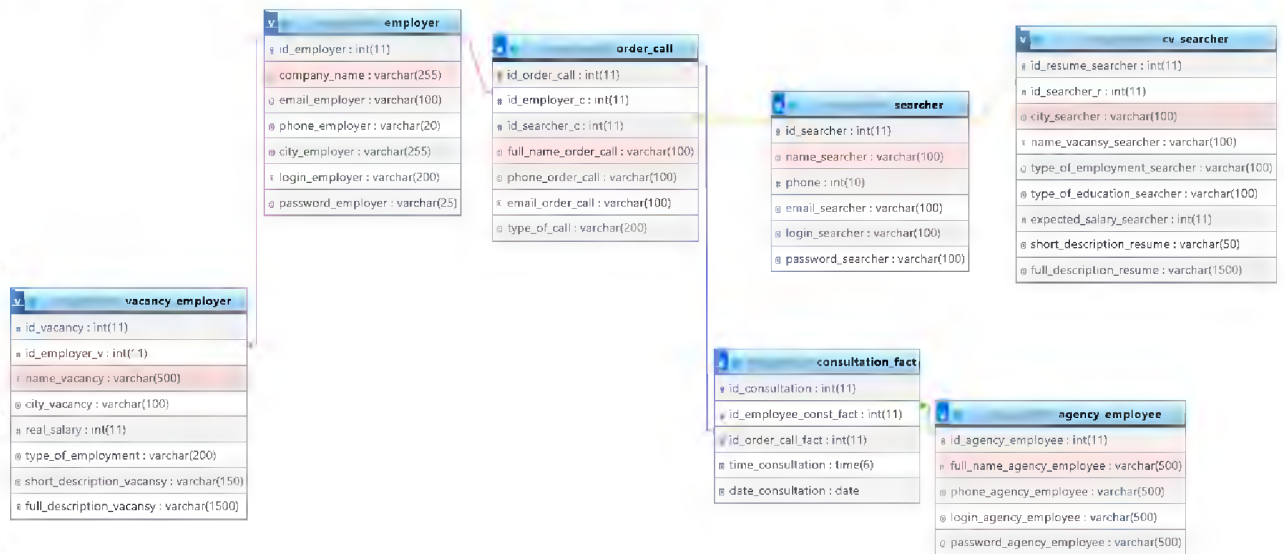


Рис.3.3 Фізична модель БД

Фізична модель даних зображена на рис.3.3, яка складається з таких таблиць:

- **employer**: містить інформацію про шукачу кандидату;
- **vacancy_employee**: містить інформацію про вакансію яку розміщує

шукач;

- **order_call** містить інформацію про замовлення консультації

- searcher: містить інформацію про кандидату;
- cv_searcher: містить інформацію про резюме кандидату;
- agency_employee: містить інформацію про працівника агенції;
- consultation_fact: містить інформацію про проведену консультацію.

Алгоритмізація та програмування програмних модулів

На рис.3.4 представлена блок-схема виконання алгоритму входу в систему.

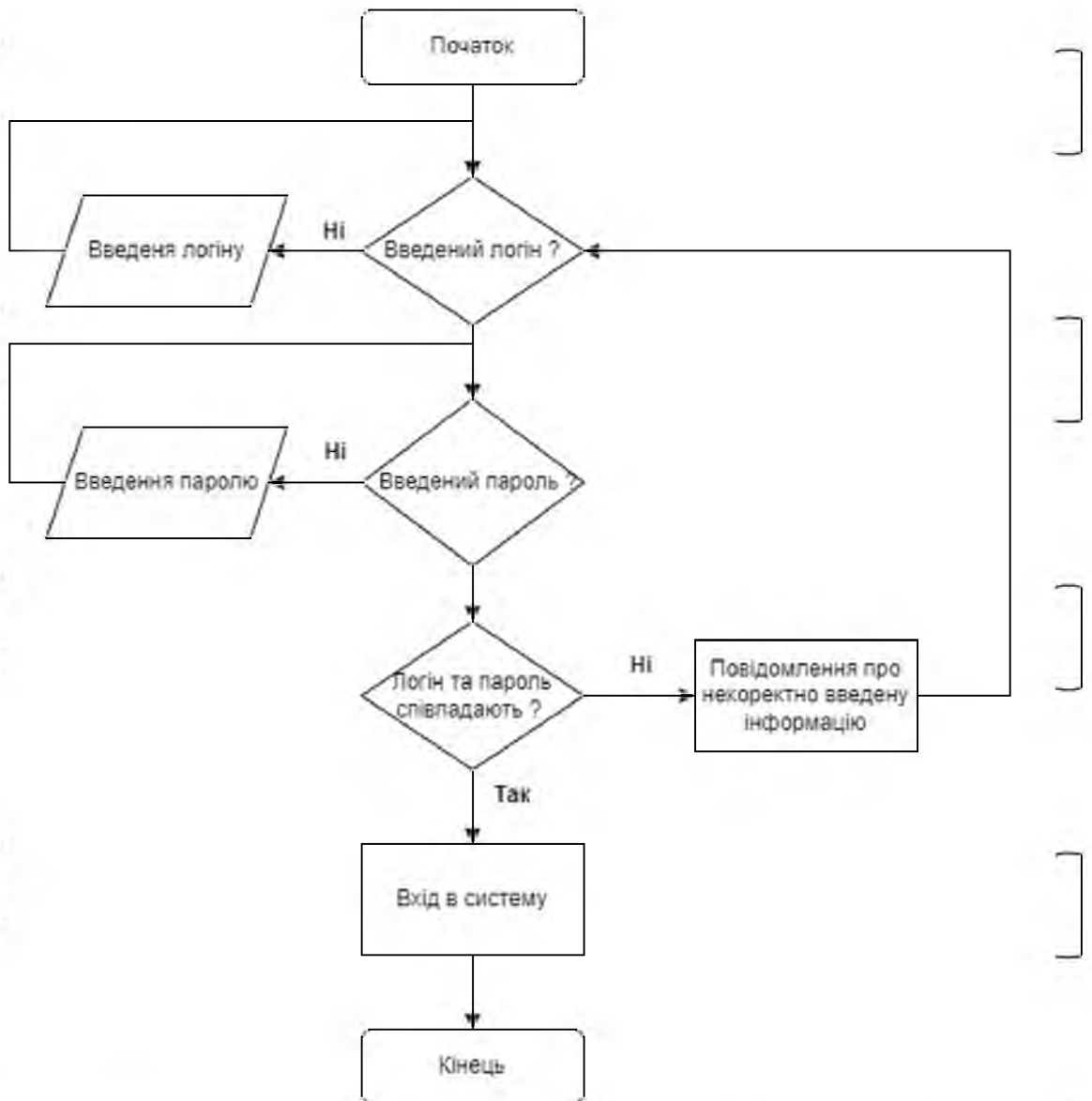


Рис 3.4 Блок-схема входу в систему

Код реалізації алгоритму входу в систему представлений на рис.3.5.

Суть алгоритму полягає в тому, що, після введення інформації в поля та натискання на кнопку 'Вхід в систему', відбувається з'єднання з базою даних та пошук в таблиці введеного логіну.

Після знаходження логіну відбувається перевірка, якщо логін та пароль співпадають, створюється сесія та відбувається переадресація на сторінку користувача. Якщо логін та пароль не співпадають, виводиться повідомлення про не вірний логін або пароль.

```

?php
session_start();
require_once 'connect.php';
$login_employer = $_POST['login_employer'];
$password_employer = $_POST['password_employer'];
$check_user = mysqli_query($connect, "SELECT * FROM `employer` WHERE `login_employer` = '$login_employer' AND `password_employer` = '$password_employer'");
if (mysqli_num_rows($check_user) > 0) {

    $user = mysqli_fetch_assoc($check_user);

    $_SESSION['employer'] = [
        "id_employer" => $user['id_employer'],
        "company_name" => $user['company_name'],
        "email_employer" => $user['email_employer'],
        "phone_employer" => $user['phone_employer'],
        "city_employer" => $user['city_employer'],
        "email_employer" => $user['email_employer'],
        "login_employer" => $user['login_employer'],
        "password_employer" => $user['password_employer']
    ];

    header('Location: ../profile_employer.php');

} else {
    $_SESSION['message'] = 'Не вірний логін або пароль';
    header('Location: ../index.php');
}

```

Рис.3.5 PHP-код входу в систему

На рис.3.6 представлена блок-схема виконання алгоритму внесення заявки на лізнок.

Суть алгоритму полягає в тому, що після введення інформації в поля та натискання на кнопку 'Залишити заявку' відбувається з'єднання з БД та внесення даних то даблиці. Якщо користувач заповнить не всі поля які присутні в формі йому буде показано повідомлення про необхідність заповнити відповідне поле інформацією.

В залежності від вибраного типу консультації данні записуються в відповідну таблицю. Після того як данні успішно додано до таблиці в БД, користувачу буде показано вікно з повідомлення про успішне додавання запису

за допомогою скрипта та переадресує на сторінку залишення заявки на консультацію

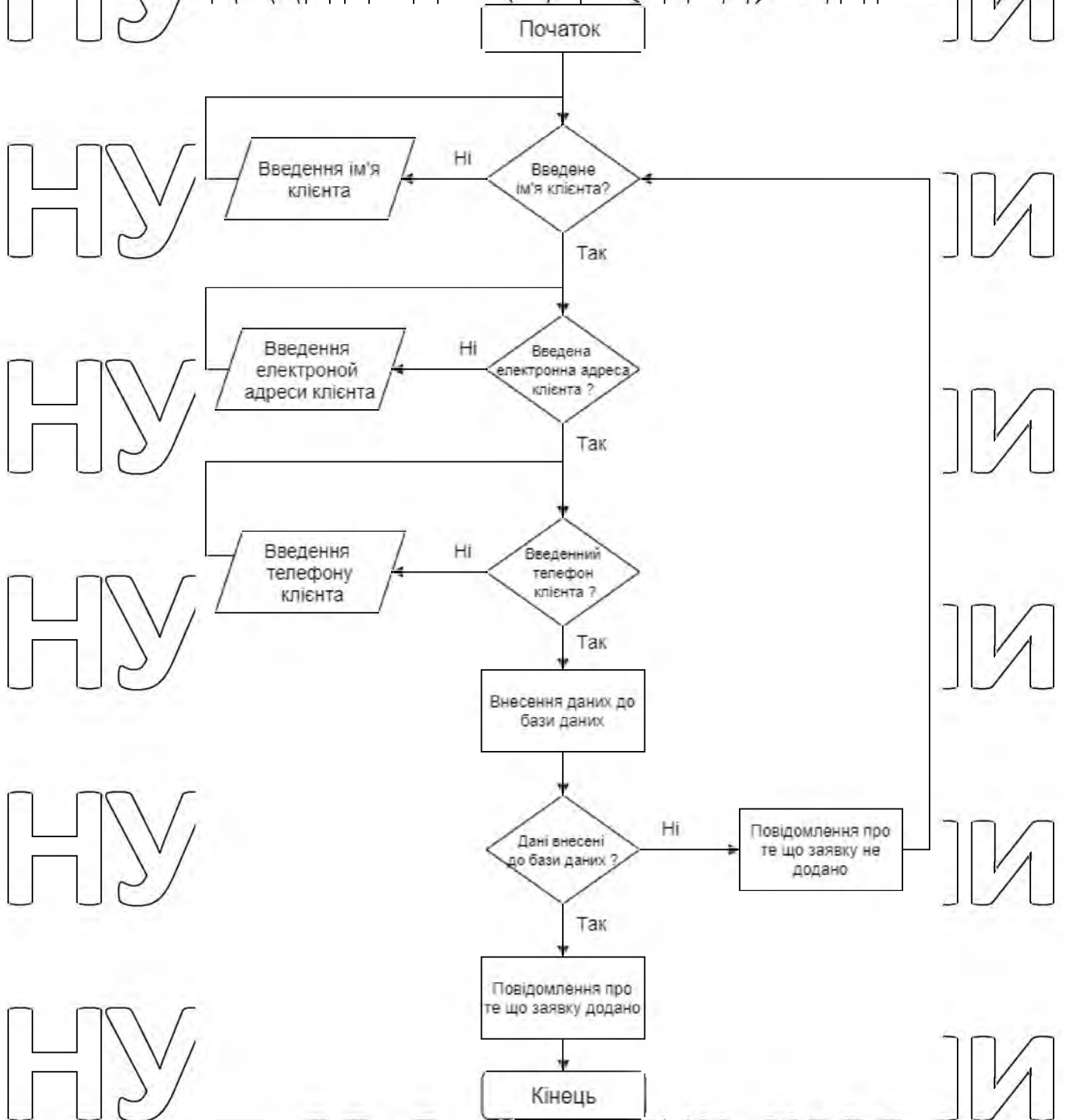


Рис.3.6 Блок схема залишення заявки на дзвінок

Код реалізації алгоритму внесення заявки представлений на рис.3.7.

```

session_start();
require_once 'connect.php';
$full_name = $_POST['full_name'];
$email = $_POST['email'];
$phone = $_POST['phone'];
$sel = $_POST['sel'];
if($sel == 'work')

    mysqli_query($connect, "INSERT INTO `consultation_work` (`id`, `full_name`, `email`, `phone`) VALUES (NULL, '$full_name', '$email', '$phone')");
    header('Location: ../call.php');

if($sel == 'employer')

    mysqli_query($connect, "INSERT INTO `consultation_employer` (`id`, `full_name`, `email`, `phone`) VALUES (NULL, '$full_name', '$email', '$phone')");
    header('Location: ../call.php');

```

Рис.3.7 PHP-код залишення заявки на дзвінок

Код виводу повідомлення про успішне внесення зображено на рис.3.9.

```

$(document).ready(function() {

    $("#form").submit(function() {
        $.ajax({
            type: "POST",
            url: "phpwithdb/leaverequest.php",
            data: $(this).serialize()
        }).done(function() {
            $(this).find("input").val("");
            alert("Дякуємо за заявку! Очікуйте дзвінка :)");
            $("#form").trigger("reset");
        });
        return false;
    });
});

```

Рис.3.9 JS-код виведення повідомлення про успішне додавання заявки

На рис.3.10 представлена блок-схема виконання алгоритму реєстрація користувача.

Суть алгоритму полягає в тому, що після введення інформації в поля та натискання на кнопку 'Реєстрація' відбувається з'єднання з БД та внесення даних то даблиці. Якщо користувач заповнить не всі поля які присутні в формі йому буде показано повідомлення про необхідність заповнити відповідне поле інформацією.

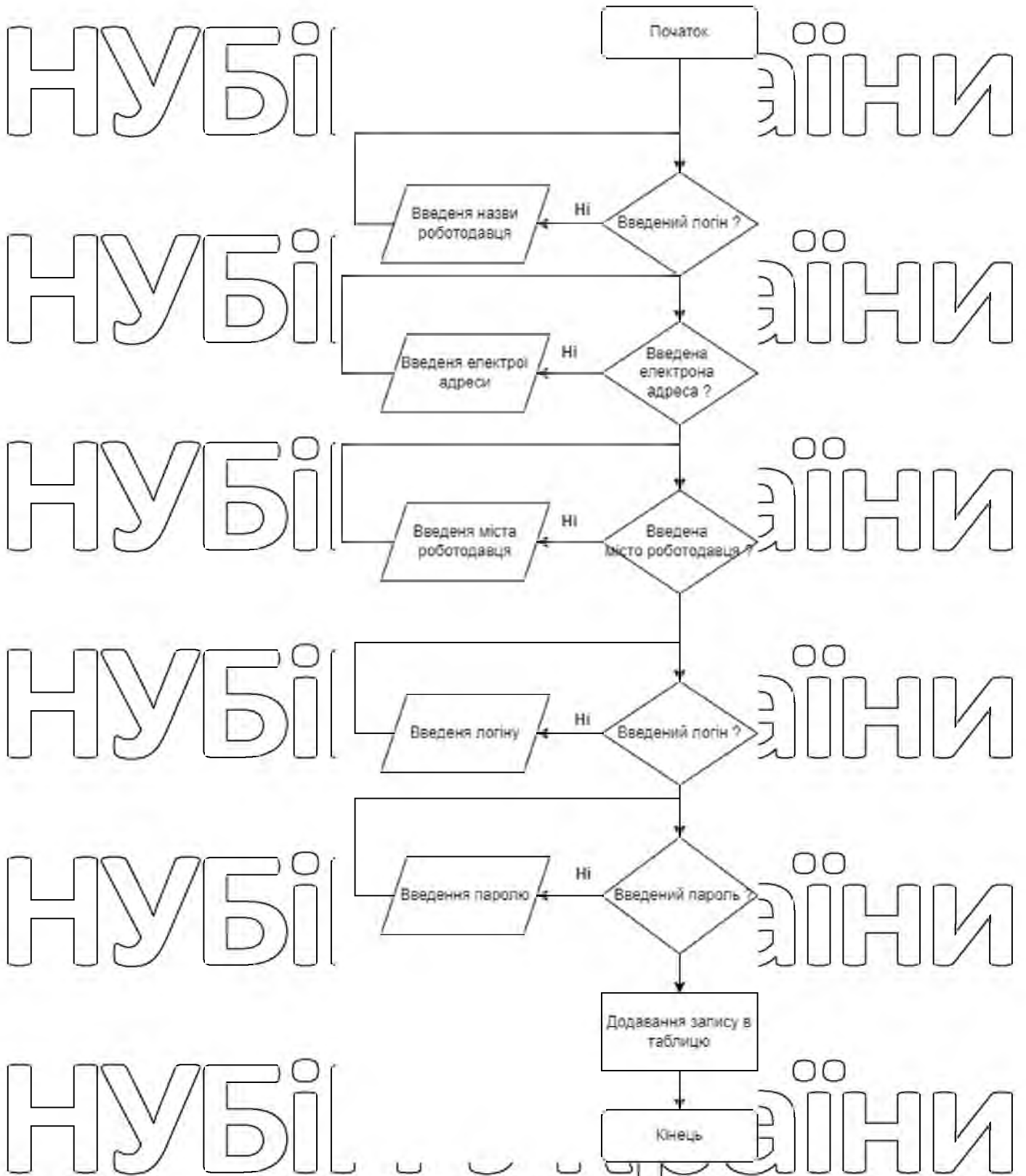


Рис.3.10 Блок схема реєстрації користувача

Код реалізації реєстрації користувача представлений на рис.3.11.

```

<?php
session_start();
require_once 'connect.php';
$name_employer = $_POST['name_employer'];
$email_employer = $_POST['email_employer'];
$phone_employer = $_POST['phone_employer'];
$city_employer = $_POST['city_employer'];
$login_employer = $_POST['login_employer'];
$password_employer = $_POST['password_employer'];
mysql_query($connect, "INSERT INTO `employer` (`id_employer`, `name_employer`, `email_employer`, `phone_employer`, `city_employer`, `login_employer`, `password_employer`)
VALUES (NULL, '$name_employer', '$email_employer', '$phone_employer', '$city_employer', '$login_employer', '$password_employer')");
mysql_error($connect);

```

Рис.3.11 PHP-код реєстрації користувача

Код форми представлений на рис.3.12

```

<form action="request_register_employer.php" method="POST">
<p class="p0_txt_form">Реєстрація</p>
<input type="email" placeholder="Назва компанії" name="name_employer" class="inpt" required>
<input type="email" placeholder="Електронна адреса" name="email_employer" class="inpt" required>
<input type="phone" placeholder="Телефон" name="phone_employer" class="inpt" required>
<input type="phone" placeholder="Місто" name="city_employer" class="inpt" required>
<input type="login" placeholder="Логін" name="login_employer" class="inpt" required>
<input type="password" placeholder="Пароль" name="password_employer" class="inpt" required>
<button class="btn"> Реєстрація </button>
</form>

```

Рис.3.12 HTML,CSS форма реєстрація користувач

3.5 Проектування сховища даних

3.5.1 Логічна модель сховища даних

Логічна модель сховища даних – демонстрація моделі СД яка не пов'язана з якоюсь конкретною СУБД. Для дослідження була реалізована створена оперативна БД, логічна схема якої зображена на рис. 3.13. В логічній моделі СД зображають основні сутності та визначають, як зв'язки будуть встановлені між зображеними сутностями..

Відсутність логічної моделі даних може привести до погано проектування бази даних та може бути фактором через який програма буде працювати не належним чином.

В логічній моделі були виділені такі сутності:

- «Candidate» містить інформацію про кандидата;
- «Order_search_work» містить інформацію про замовлений пошук роботи в фірмі;

- «Order_search_worker» містить інформацію про замовлення фірми на пошук працівника.
- «Consultation_candidate» містить інформацію про консультацію кандидата та дату проведення консультації
- «Consultation_firm» містить інформацію про консультацію фірми та дату проведення консультації
- «Employee» містить інформацію про працівника.
- «Position» містить інформацію про посаду яку займає працівник.

На діаграмі можна побачити неідентифікуючі зв'язки.

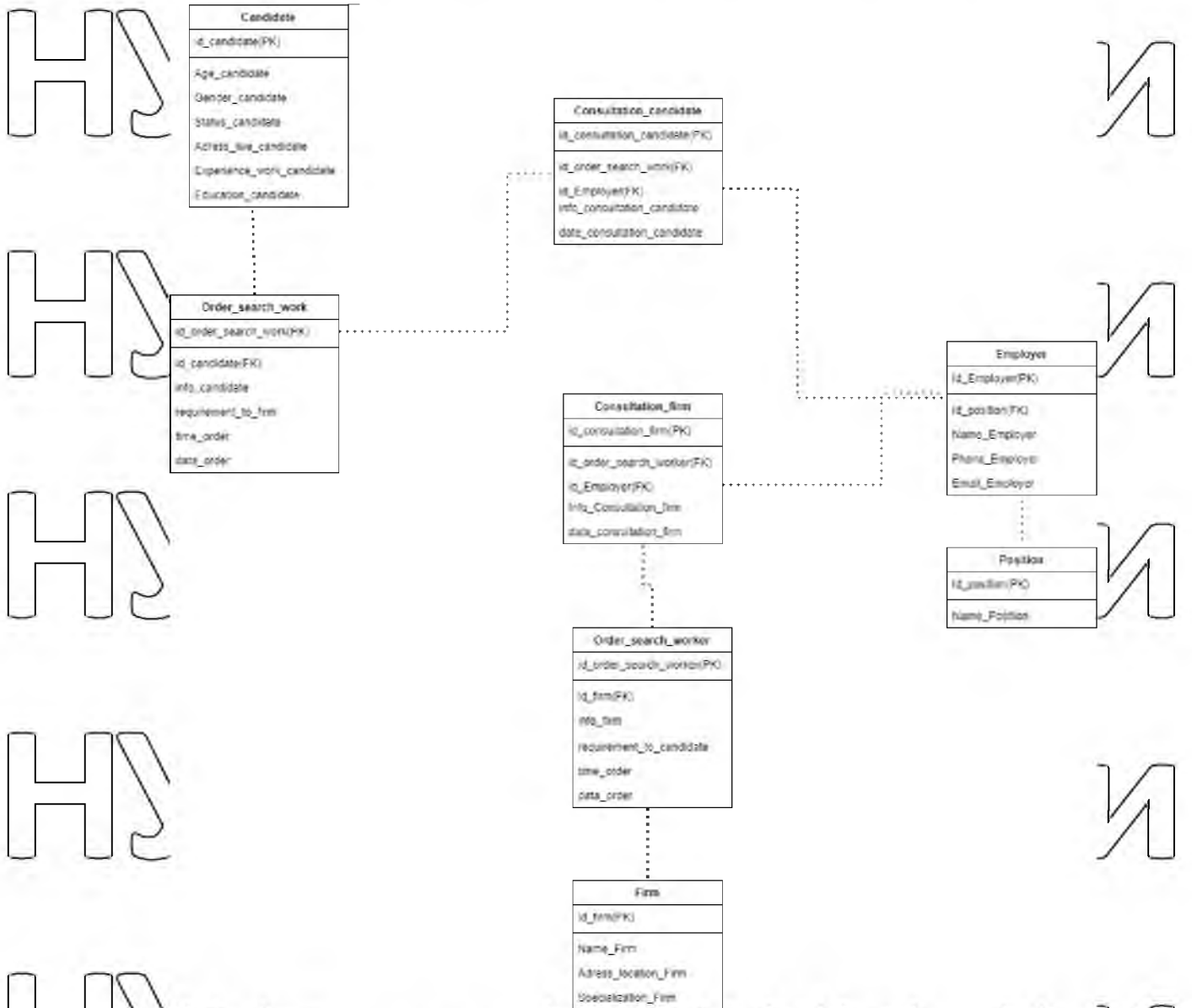


Рис. 3.13 Логічна модель сховища даних

3.5.2 Фізична модель сховища даних

Для збереження даних по роботі агентства для подальшого аналізу було спроектовано СД, фізична модель, якої представлена на рис. 3.14.

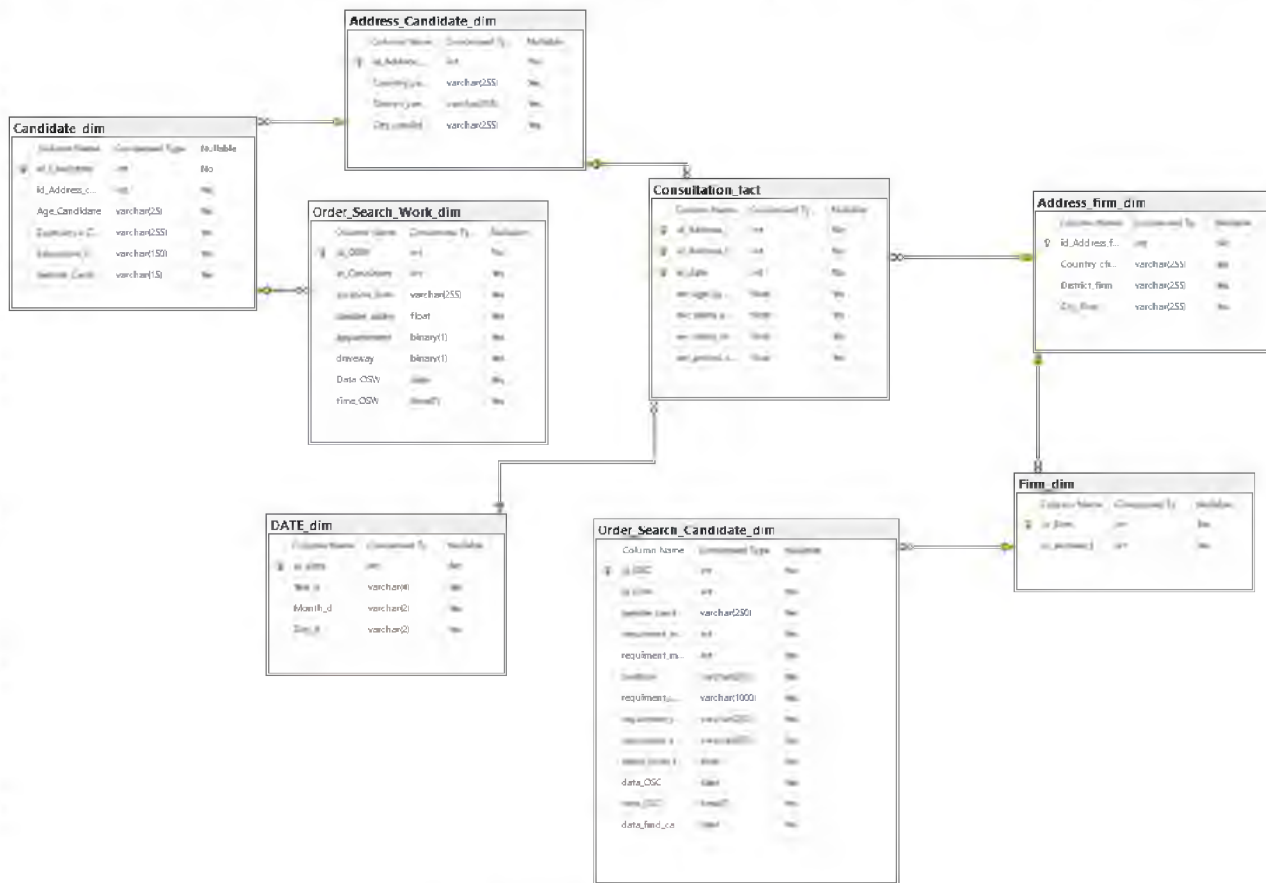


Рис. 3.14 Сховище даних

Вимір – це об’єкти одного або декількох типів, організованих у вигляді ієрархічної структури та які забезпечують інформаційний контекст числового показника (факту).

Факт – це величина, яка є предметом аналізу. Для збереження необхідних даних були розроблені таблиці вимірів та таблицю фактів:

- ✓ Candidate_dim – містить інформацію про кандидата;
- ✓ Address_Candidate_dim – містить інформацію про місце проживання кандидата;
- ✓ Order_Search_Work_dim – містить інформацію про замовлення кандидата;

✓ DATE_dim – містить інформацію про дату аналізу замовлень;
 ✓ Firm_dim – містить інформацію про фірму;
 ✓ Address_firm – містить інформацію про адресу розміщення фірми;
 ✓ Order_Search_Candidate_dim – містить інформацію про замовлення фірми на пошук кандидата;

✓ Consultation_fact – містить інформацію про середній вік кандидатів на вакансії, середній період пошуку кандидату на вакансію, середню очікувану заробітну плату та середню реальну заробітну плату від фірми на вакансію.

Таблиця вимір «Candidate_dim» містить такі атрибути:

- id_Candidate (PK);
- id_Address_candidate (FK);
- Age_candidate (Вік кандидатів);
- Experience_Candidate (Досвід роботи);
- Education_Candidate (Освіта кандидату);
- Gender_Candidate (Стать кандидата).

Таблиця вимір «Address_Candidate_dim» містить такі атрибути:

- Id_Address_candidate (PK);
- Country_candidate (Країна проживання кандидату);
- District_candidate (Область проживання кандидату);
- City_candidate (Місто проживання кандидату).

Таблиця вимір «Address_firm_dim» містить такі атрибути:

- Id_Address_firm (PK);
- Country_firm (Країна розміщення фірми);
- District_firm (Область розміщення фірми);
- City_firm (Місто розміщення фірми).

Таблиця вимір «Date_dim» містить такі атрибути:

- Id_date (PK);
- Year_d (Рік звітності);

- Month_d (Місяць звітності);
- Day_D (День звітності).

Таблиця вимір «Firm_dim» містить такі атрибути:

- Id_Firm (PK);
- Id_Address_firm (FK);

Таблиця вимір «Order_Search_Candidate_dim» містить такі атрибути:

- Id_OSC (PK);
- Id_Firm (FK);
- Gender_candidate (Стать кандидату);

- Requirement_min_age_to_candidate (Мінімальний вік кандидату);
- Requirement_max_age_to_candidate (Максимальний вік кандидату);
- Position (Посада для працевлаштування);
- Requirement_to_education (Вимоги до освіти кандидату);

- Requirement_to_location_candidate (Вимоги до місця проживання кандидату);
- Requirement_to_experience_candidate (Вимоги до досвіду роботи кандидату);

- Salary_from_firm (Заробітна плата);
- Date_OSC (Дата оформлення замовлення);
- Time_OSC (Час оформлення замовлення);
- Data_find_candidate (Дата знаходження кандидату);

Таблиця вимір «Order_Search_Work_dim» містить такі атрибути:

- Id_OS_W (PK);
- Id_Candidate (FK);
- Location_firm (Вимоги до розміщення фірми);
- Desider_salary (Очікувана зарплата);

- Apartment (Наявність житла);
- Driveway (Наявність оплачуваного перевезення до роботи);

- Date_OSW (Дата оформлення замовлення);

- Time_OSW (Час оформлення замовлення);

Таблиця фактів «Consultation_fact» містить такі атрибути

- Id_Address_candidate (PK);

- Id_Address_firm (PK);

- Id_date (PK);

- Avr_age_by_position (середній вік кандидатів на вакансію);

- Avr_salary_expected_position (середня очікувана зарплата на вакансію);

- Avr_salary_real_position (середня реальна заробітна плата за вакансію);

- Avr_period_search_candidate (середній період пошуку кандидату).

Запити по заповненню таблиць-вимірів та таблиць-фактів написано на SQL та представлено на рис 3.15.

```
USE TEST4
insert into
Candidate_dim(id_Candidate,id_Address_candidate, Age_Candidate
, Experience_Candidate, Education_Candidate, Gender_Candidate)
VALUES ('1', '2', '26', 'Більше 1 року', 'Вища', 'Чоловічий');
insert into
Candidate_dim(id_Candidate,id_Address_candidate, Age_Candidate
, Experience_Candidate, Education_Candidate, Gender_Candidate)
VALUES ('2', '2', '39', 'Більше 1 року', 'Вища', 'Чоловічий');
insert into
Candidate_dim(id_Candidate, id_Address_candidate, Age_Candidate
, Experience_Candidate, Education_Candidate, Gender_Candidate)
VALUES ('3', '2', '23', 'Відсутній', 'Вища', 'Чоловічий');
```

Рис. 3.15 SQL-запит для наповнення таблиць-вимірів

На рис.3.16 продемонстровано інформацію, якою заповнено таблицю замовлення кандидатів на пошук роботи.

	id_OSW	id_Candidate	location_firm	desider_salary	appartement	driveway	Data_OSW	time_OSW
1	1	1	Київ	30000	0x01	0x00	2022-04-17	11:00:00.00000000
2	2	2	Київ	25000	0x01	0x00	2022-04-17	09:00:00.00000000
3	3	3	Одеса	12000	0x00	0x01	2022-03-17	12:00:00.00000000
4	4	4	Одеса	15000	0x00	0x01	2022-03-17	08:00:00.00000000
5	5	5	Одеса	60000	0x00	0x01	2022-02-16	14:00:00.00000000
6	6	6	Одеса	5000	0x00	0x01	2022-06-16	16:00:00.00000000
7	7	7	Одеса	10000	0x01	0x00	2022-04-16	15:00:00.00000000
8	8	8	Одеса	27000	0x01	0x00	2022-04-16	18:00:00.00000000
9	9	9	Одеса	21000	0x00	0x01	2022-03-17	18:30:00.00000000
10	10	10	Одеса	15000	0x00	0x01	2022-04-16	18:30:00.00000000
11	11	11	Київ	20000	0x01	0x00	2022-03-17	18:35:00.00000000
12	12	12	Київ	15000	0x00	0x01	2022-04-15	21:00:00.00000000
13	13	13	Київ	15000	0x00	0x01	2022-04-15	21:30:00.00000000
14	14	14	Київ	10000	0x00	0x01	2022-03-17	21:40:00.00000000
15	15	15	Київ	25000	0x00	0x01	2022-04-15	21:55:00.00000000
16	16	16	Київ	20000	0x00	0x01	2022-04-15	22:30:00.00000000
17	17	17	Київ	7000	0x00	0x01	2022-04-15	22:40:00.00000000
18	18	18	Київ	7000	0x00	0x01	2022-04-15	22:50:00.00000000
19	19	19	Київ	20000	0x00	0x01	2022-03-17	13:00:00.00000000
20	20	20	Одеса	45000	0x00	0x01	2022-04-15	13:00:00.00000000

Рис.3.16 Дані таблиці-вимір Заовлення кандидатів

На рис.3.17 зображено дані, якими наповнено таблицю виміру Дата.

	id_date	Year_d	Month_d	Day_d
1	1	2022	01	15
2	2	2022	02	15
3	3	2022	03	15
4	4	2022	04	15
5	5	2022	05	15
6	6	2022	06	15
7	7	2022	07	15
8	8	2022	08	15
9	9	2022	09	15
10	10	2022	10	15
11	11	2022	11	15
12	12	2022	12	15

Рис.3.17 Дані таблиці-вимір Дата

Рис.3.18 демонструє заповнену даними таблицю замовлення від фірм на пошук кандидата.

id_Candidate	id_Address_candidate	Age_Candidate	Experience_Candidate	Education_Candidate	Gender_Candidate
1	1	26	Більше 1 року	Вища	Чоловічий
2	2	39	Більше 1 року	Вища	Чоловічий
3	3	23	Відсутній	Вища	Чоловічий
4	4	31	Більше 1 року	Вища	Чоловічий
5	5	39	Більше 1 року	Вища	Чоловічий
6	1	39	Більше 1 року	Вища	Чоловічий
7	2	32	Більше 1 року	Вища	Чоловічий
8	3	30	Більше 1 року	Вища	Чоловічий
9	4	21	Відсутній	Незакінчена вища	Чоловічий
10	1	20	Відсутній	Незакінчена вища	Чоловічий
11	4	22	Відсутній	Вища	Чоловічий
12	1	31	Більше 1 року	Вища	Чоловічий
13	4	26	Більше 1 року	Вища	Чоловічий
14	1	21	Більше 1 року	Вища	Чоловічий
15	2	31	Більше 1 року	Вища	Чоловічий
16	2	41	Більше 1 року	Вища	Чоловічий
17	4	20	Більше 1 року	Вища	Чоловічий
18	1	18	Відсутній	Вища	Чоловічий
19	1	31	Відсутній	Вища	Чоловічий
20	4	51	Більше 1 року	Вища	Чоловічий

Рис.3.18 Пошук кандидатів

На рис.3.19 представлено наповнення таблиці виміру Candidate.

id_Candidate	id_Address_candidate	Age_Candidate	Experience_Candidate	Education_Candidate	Gender_Candidate
1	1	26	Більше 1 року	Вища	Чоловічий
2	2	39	Більше 1 року	Вища	Чоловічий
3	3	23	Відсутній	Вища	Чоловічий
4	4	31	Більше 1 року	Вища	Чоловічий
5	5	39	Більше 1 року	Вища	Чоловічий
6	1	39	Більше 1 року	Вища	Чоловічий
7	2	32	Більше 1 року	Вища	Чоловічий
8	3	30	Більше 1 року	Вища	Чоловічий
9	4	21	Відсутній	Незакінчена вища	Чоловічий
10	1	20	Відсутній	Незакінчена вища	Чоловічий
11	4	22	Відсутній	Вища	Чоловічий
12	1	31	Більше 1 року	Вища	Чоловічий
13	4	26	Більше 1 року	Вища	Чоловічий
14	1	21	Більше 1 року	Вища	Чоловічий
15	2	31	Більше 1 року	Вища	Чоловічий
16	2	41	Більше 1 року	Вища	Чоловічий
17	4	20	Більше 1 року	Вища	Чоловічий
18	1	18	Відсутній	Вища	Чоловічий
19	1	31	Відсутній	Вища	Чоловічий
20	4	51	Більше 1 року	Вища	Чоловічий

рис.3.19 Таблиця-вимір Candidate

Рис.3.20 зображає інформацію про адреси розміщення фірм.

id_Address_firm	Country_cfim	District_firm	City_firm
1	Україна	Київська область	Київ
2	Україна	Одеська область	Одеса
3	Україна	Черкаська область	Черкаси
4	Україна	Харківська область	Харків
5	Україна	Дніпропетровська область	Дніпро

Рис.3.20 Дані таблиці-вимір адреса фірм.

3.6 Створення OLAP кубу

Онлайн-аналітична обробка – технологія, за допомогою якої можна проводити складний аналіз та організовувати бізнес-бази даних. Ця технологія може бути використана для виконання аналітичних запитів.[7]

OLAP-системи були розроблені для того, щоб допомогти витягти бізнес-аналітичну інформацію з даних у високопродуктивний спосіб. Це пов'язано з тим, що бази даних OLAP оптимізовані для великих навантажень для читання і малих навантажень на запис.[7]

За допомогою кубу-OLAP можна швидко аналізувати дані. Створюємо Куб-OLAP його за допомогою служби SQL Server Analysis Services.

SQL Server Analysis Services (SSAS) – служба, яка дозволяє працювати з кубами OLAP та забезпечує створення розширень та налаштування сховища даних.

Для роботи з SSAS було використано середовище розробки Visual Studio. Спочатку визначаємо джерелом даних яким є сховище даних. Куб-OLAP підключається до джерела даних для обробки даних та опрацювання пов'язаних з ним вимірів

На рис.3.21 представлено створення джерела даних за допомогою диспетчера з'єднання та підключення джерела даних на рис.3.22 за допомогою модуля Data Source Wizard.

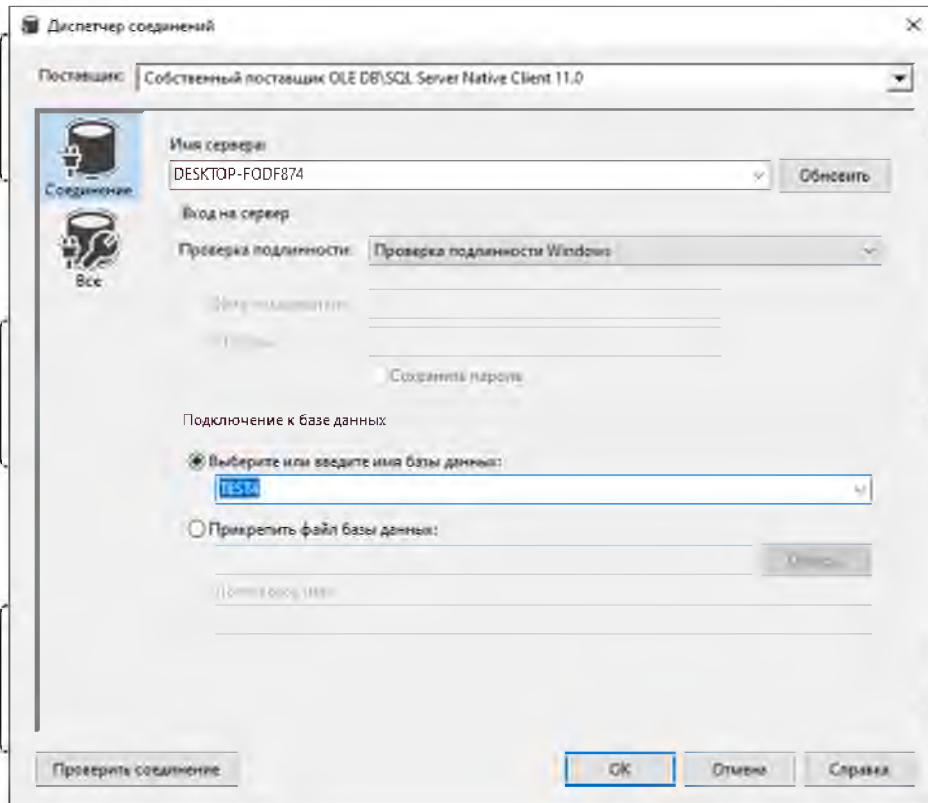


Рис. 3.21 Створення джерела даних

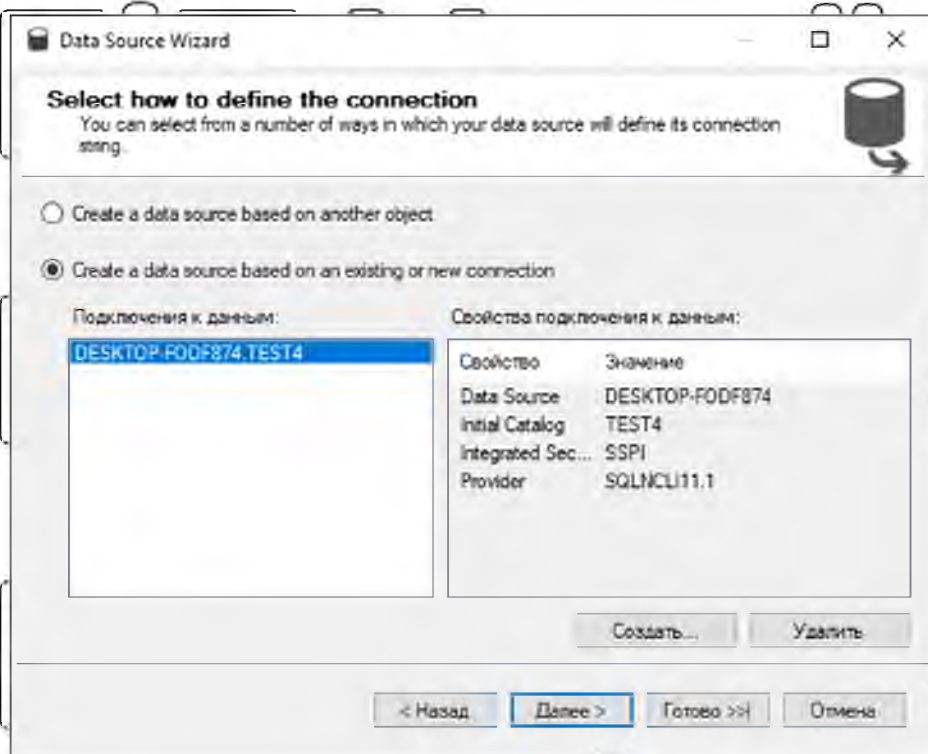


Рис. 3.22 Підключення СД

Наступним кроком після підключення сховища даних є створення уявлення джерела даних. Уявлення відображає відношення між таблицями. На

рис. 3.23 та 3.24 представлено створення подання джерела даних за допомогою модуля Data Source View Wizard.

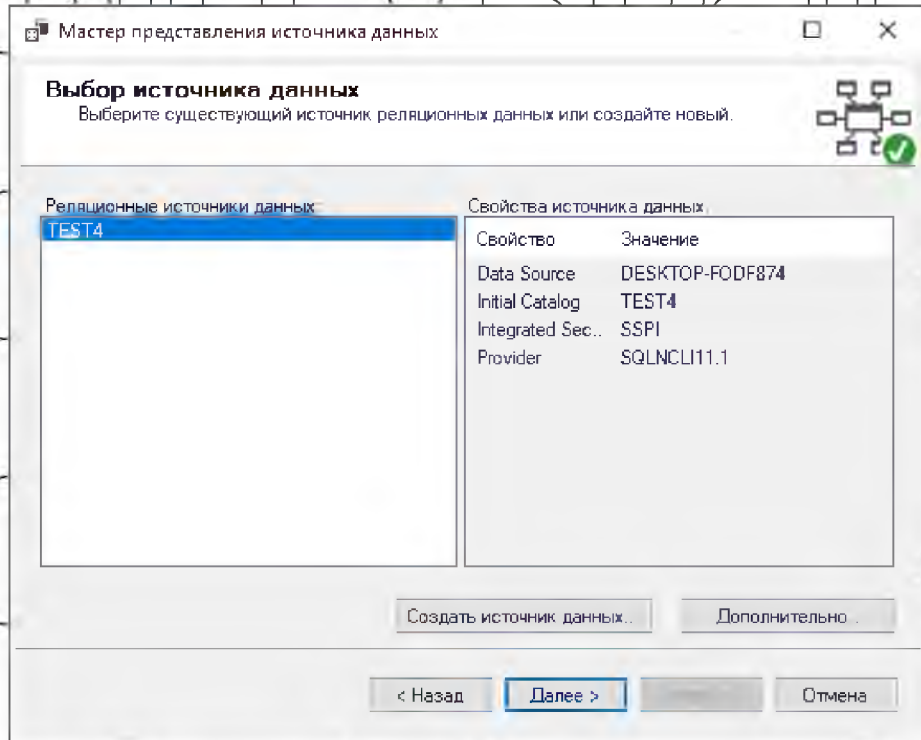


Рис. 3.23 Уявлення джерела даних на основі СД

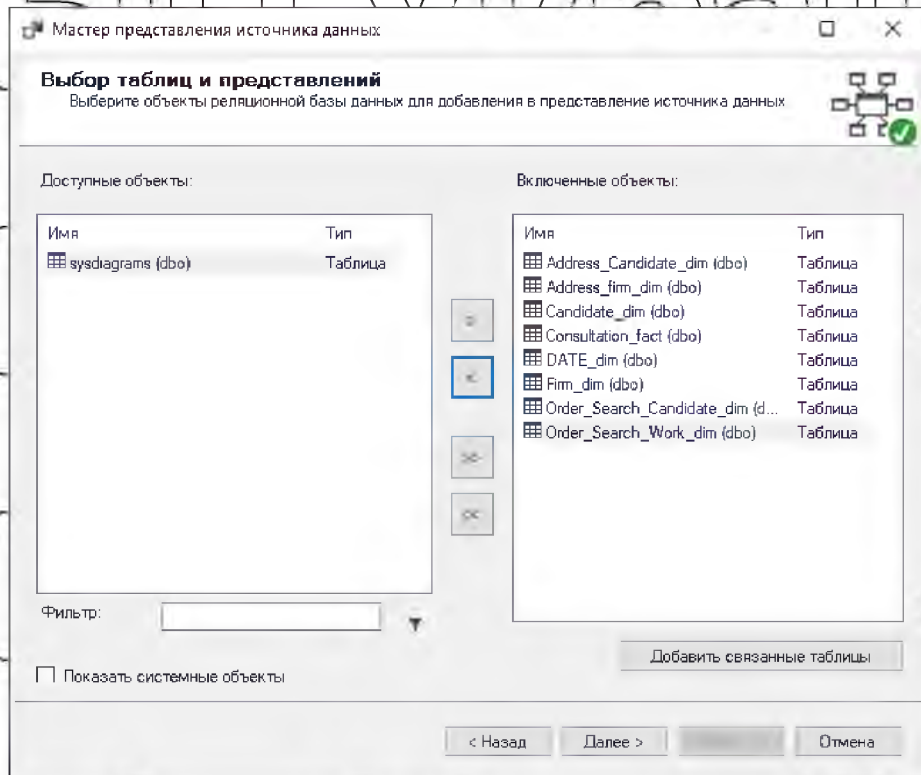


Рис. 3.24 Таблиці що формують уявлення

Після налаштування та створення джерела даних, створимо виміри для куба OLAP. Куб матиме три основних виміри, на рис.3.25 представлені створені виміри.

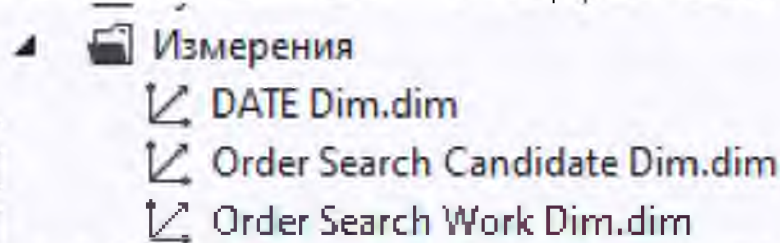


Рис.3.25 Створені виміри

Після того як були налаштовані та створені виміри, розгортаємо OLAP куб за допомогою модуля Cube Wizard (рис.3.26).

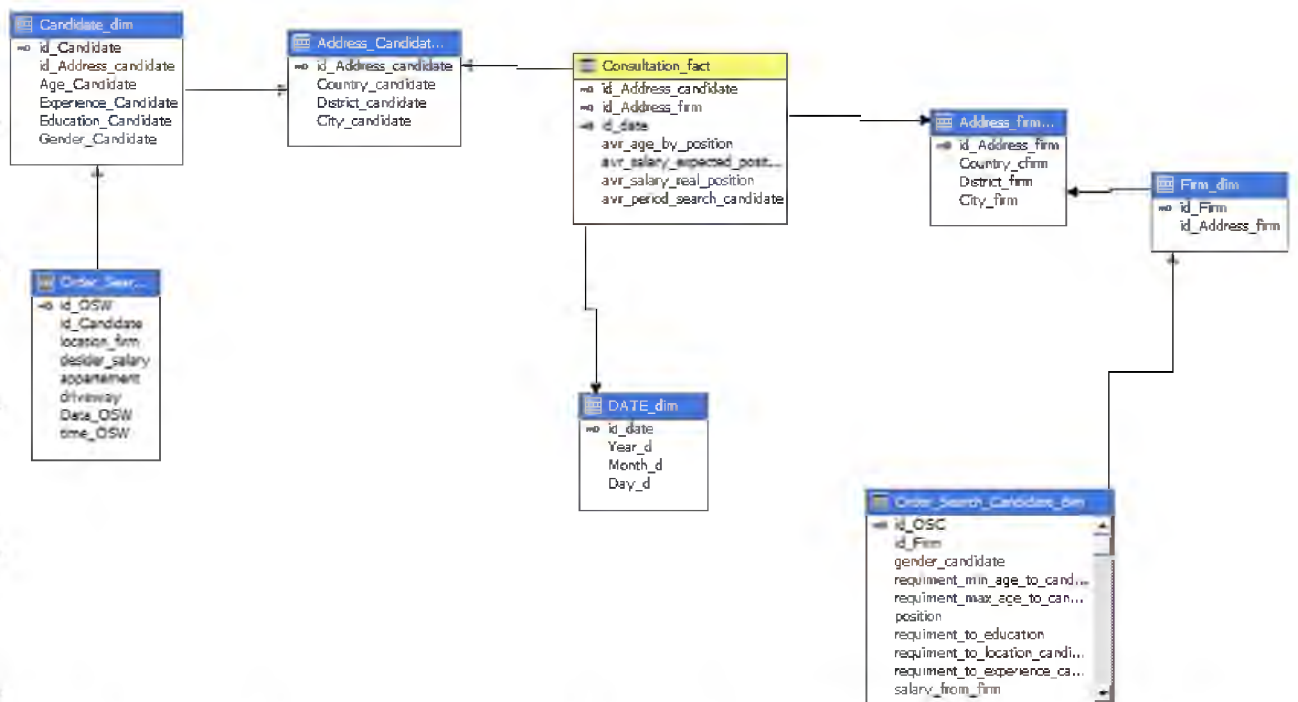


Рис.3.26 Розгорнутий куб

3.7 Реалізація отримання даних за допомогою Data Flow

Data Flow – це служба яка використовується для наповнення куба даними. Наповнення таблиці-фактів було поділено на 5 етапів що представлені на рис.3.27.



Рис.3.27 Потоки даних для наповнення таблиці-фактів.

На першому етапі відбувається заповнення всіх ключових полів таблиці-фактів. Він складається з двох блоків, що подані рис.3.28.



Рис.3.28 Перший етап наповнення

На рис.3.29 зображено блок «Формуємо вибірку» який з'єднується з таблицями-виміри та формує потрібну вибірку. Блок «Зберігаємо в таблицю фактів» зберігає отримані дані в таблиці-фактів.

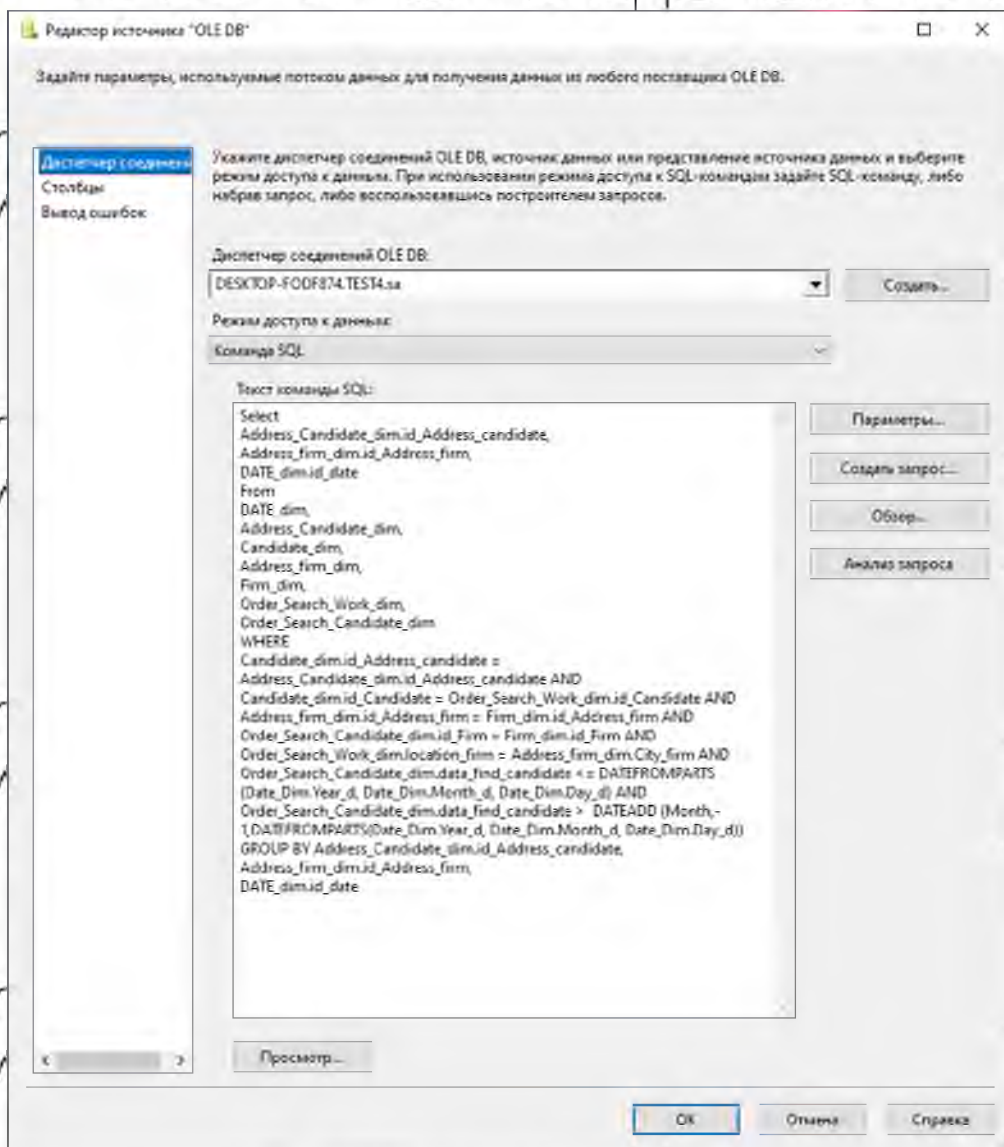


Рис.3.29 Формування вибірки

Наступні блоки мають однакові структури, що продемонстрована на рис.3.30.

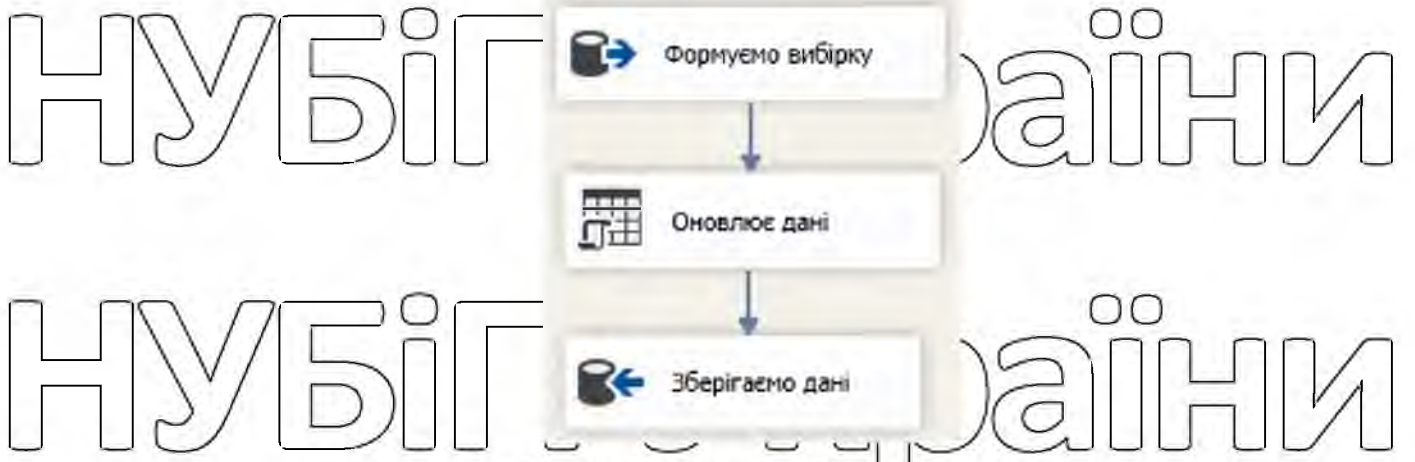


Рис.3.30 Структура 2-5 етапів.

На рис.3.31 представлено заповнену таблицю-фактів.

	id_Address_candidate	id_Address_firm	id_date	avr_age_by_position	avr_salary_expected_position	avr_salary_real_position	avr_period_search_candidate
1	1	1	4	30	21051.2820512821	15000	0
2	1	1	5	33	21051.2820512821	14000	30
3	1	1	6	40	21051.2820512821	8500	61
4	1	2	5	22	21051.2820512821	15000	30
5	1	2	8	35	21051.2820512821	14000	122
6	1	2	11	29	21051.2820512821	18000	214
7	1	3	5	32	21051.2820512821	10000	30
8	1	3	7	25	21051.2820512821	15000	91
9	1	3	12	32	21051.2820512821	14000	244
10	1	5	5	33	21051.2820512821	16500	30
11	2	1	4	30	33125	15000	0
12	2	1	5	33	33125	14000	30
13	2	1	6	40	33125	8500	61
14	2	2	5	22	33125	15000	30
15	2	2	8	35	33125	14000	122
16	2	2	11	29	33125	18000	214
17	3	1	4	30	17100	15000	0
18	3	1	5	33	17100	14000	30
19	3	1	6	40	17100	8500	61
20	3	2	5	22	17100	15000	30
21	3	2	8	35	17100	14000	122
22	3	2	11	29	17100	18000	214
23	4	1	4	30	20428.5714285714	15000	0
24	4	1	5	33	20428.5714285714	14000	30
25	4	1	6	40	20428.5714285714	8500	61
26	4	2	5	22	20428.5714285714	15000	30
27	4	2	8	35	20428.5714285714	14000	122
28	4	2	11	29	20428.5714285714	18000	214
29	5	1	4	30	33076.9230769231	15000	0
30	5	1	5	33	33076.9230769231	14000	30
31	5	1	6	40	33076.9230769231	8500	61
32	5	2	5	22	33076.9230769231	15000	30
33	5	2	8	35	33076.9230769231	14000	122
34	5	2	11	29	33076.9230769231	18000	214
35	5	3	5	32	33076.9230769231	10000	30
36	5	3	7	25	33076.9230769231	15000	91
37	5	3	12	32	33076.9230769231	14000	244

Рис.3.31 Результат виконання потоків даних

НУБІП України

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Побудова звітності в середовищі BI

SQL Server Reporting Services – служба, яка використовується для

створення звітів на основі даних з сховища даних. Для побудови звітів

спочатку підключимо джерело даних, що представлено на рисунку 4.1.

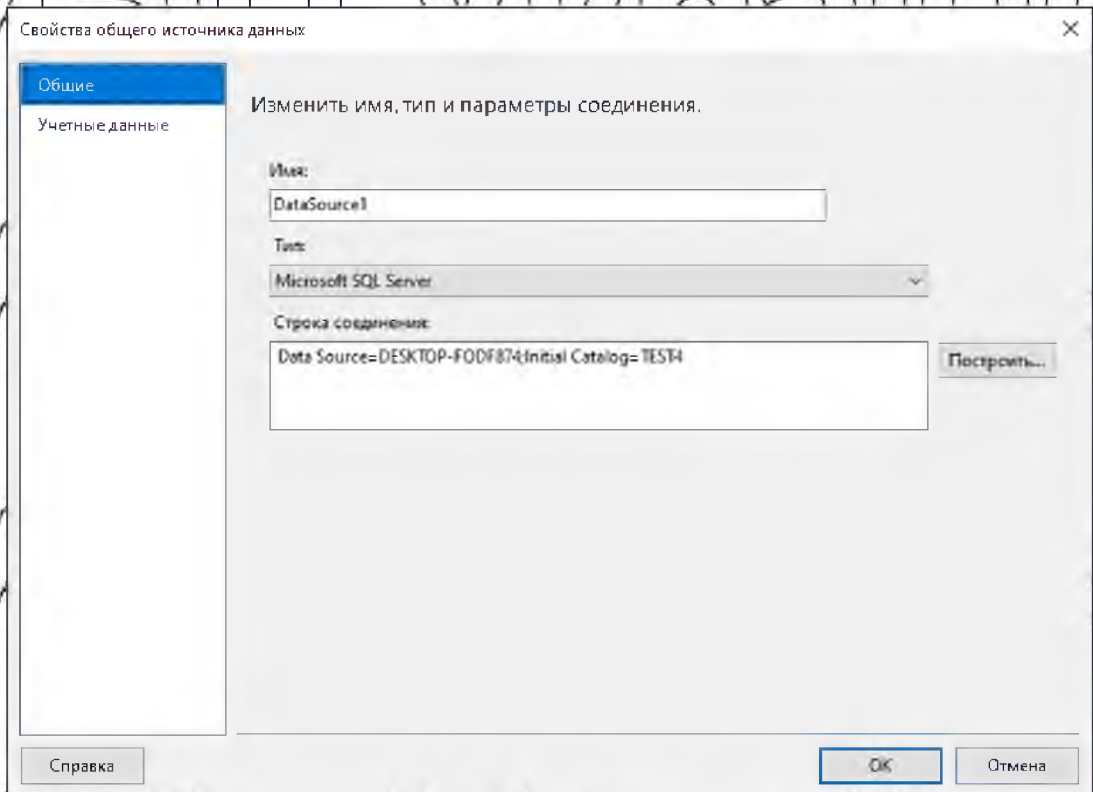


Рис. 4.1 Підключення джерела даних

Після підключення до джерела даних переходимо в майстер звітів та створюємо звіт.

За допомогою Майстра звітів створимо звіти для порівняння очікуваної та дійсної заробітної плати 2021, та період пошуку кандидату на позицію.

Етапи створення звітів представлені на рисунках 4.2 та 4.3

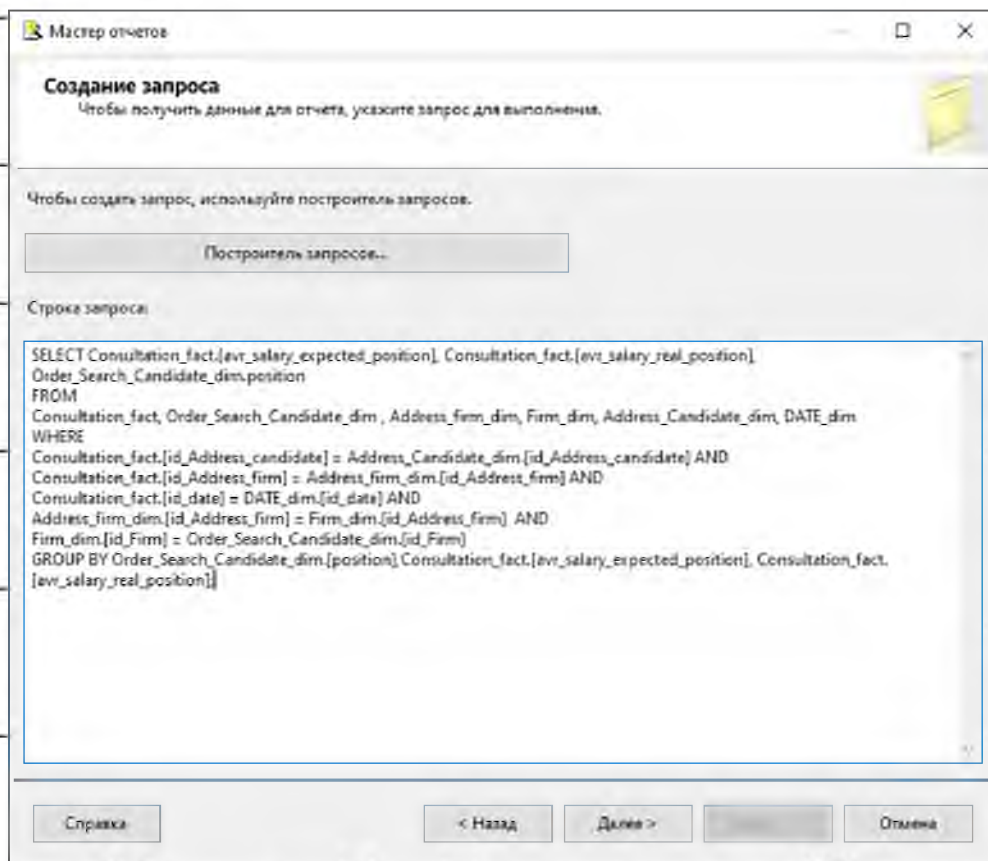


Рис. 4.2 Створення звіту порівняння очікуваної та дійсної заробітної.

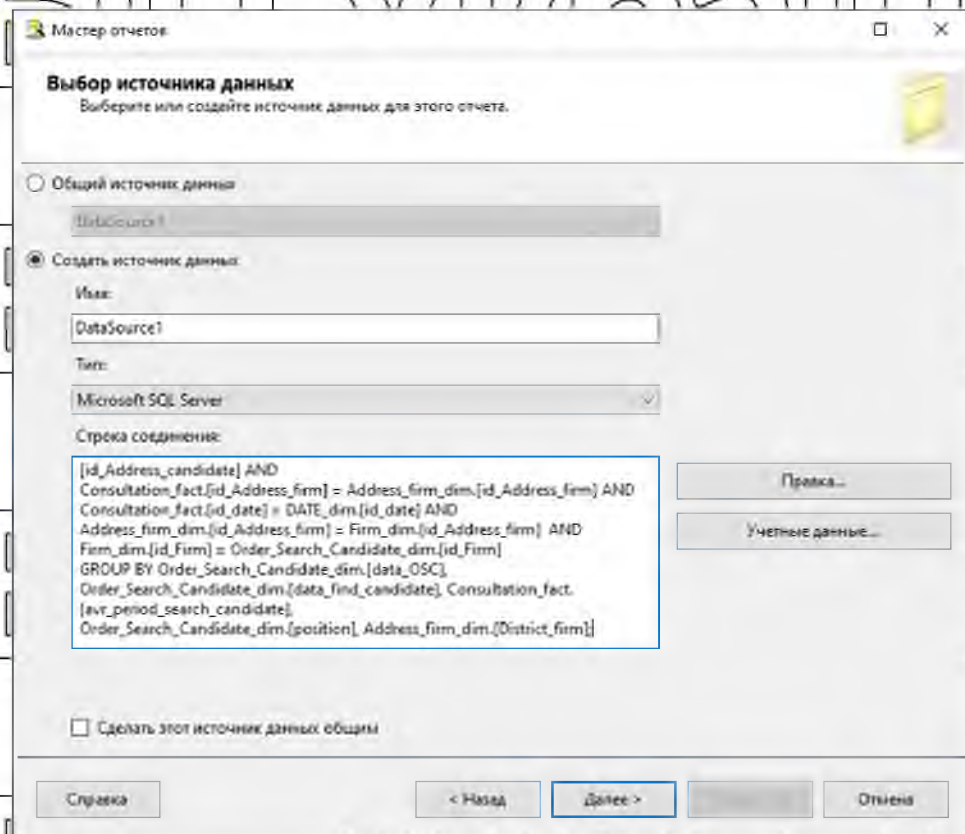


Рис. 4.3 Створення звіту період пошуку кандидату на позицію.

На рисунку 4.4 представлено сформований звіт для порівняння очікуваної та дійсної заробітної плати 2021

Порівняння очікуваної та дійсної заробітної плати 2021

Посада	Очікувана заробітна плата від кандидата	Заробітна плата від фірми
Business Analyst	18000	9000
Junior Java developer	8500	11100
	8500	15000
Junior QA Engineer	8500	11100
	8500	15000
	15000	15000
Контент-менеджер	10000	8000
	15000	8500

Рис. 4.4 Звіт для порівняння очікуваної та дійсної заробітної плати 2021

На рисунку 4.5 представлено сформований звіт період пошуку кандидату на позицію

Період пошуку кандидату на позицію

	Дніпропетровська область		Київська область		Одеська область		Харківська область	
	Дата замовлення	Дата знаходження кандидата	Період пошуку кандидата	Дата замовлення	Дата знаходження кандидата	Період пошуку кандидата	Дата замовлення	Дата знаходження кандидата
Business Analyst	11.01.2022	08.02.2022	26					
Junior Java developer			04.01.2022	04.02.2022	31			
Junior QA Engineer			01.01.2022	25.01.2022	31	01.01.2022	25.01.2022	32
Контент-менеджер								
Програміст 1С								

Рис. 4.5 Створений звіт період пошуку кандидату на позицію

4.2 Розрахунок KPI

KPI (ключовий показник ефективності) – це кількісний індикатор фактичних досягнень.

Для нашої системи ми визначили такий ключовий показник ефективності:

за який період відбувається пошук кандидата – за допомогою цього показника можна оцінити, на скільки швидко відбувається підбір персоналу і чи досягається ціль запити.

Для визначення економічної ефективності швидкості підбору персоналу

були проведені наступні кроки:

1) Задане ім'я ключового показника ефективності.

2) Визначення значення та цілі виразу (рис. 4.6), а саме значення отримуємо з таблиці фактів, а ціль зазначаємо – 30.

3) Визначення ознаки та виразу стану, що представлені на рис. 4.6.

4) Обрахунок KPI (рис. 4.7)

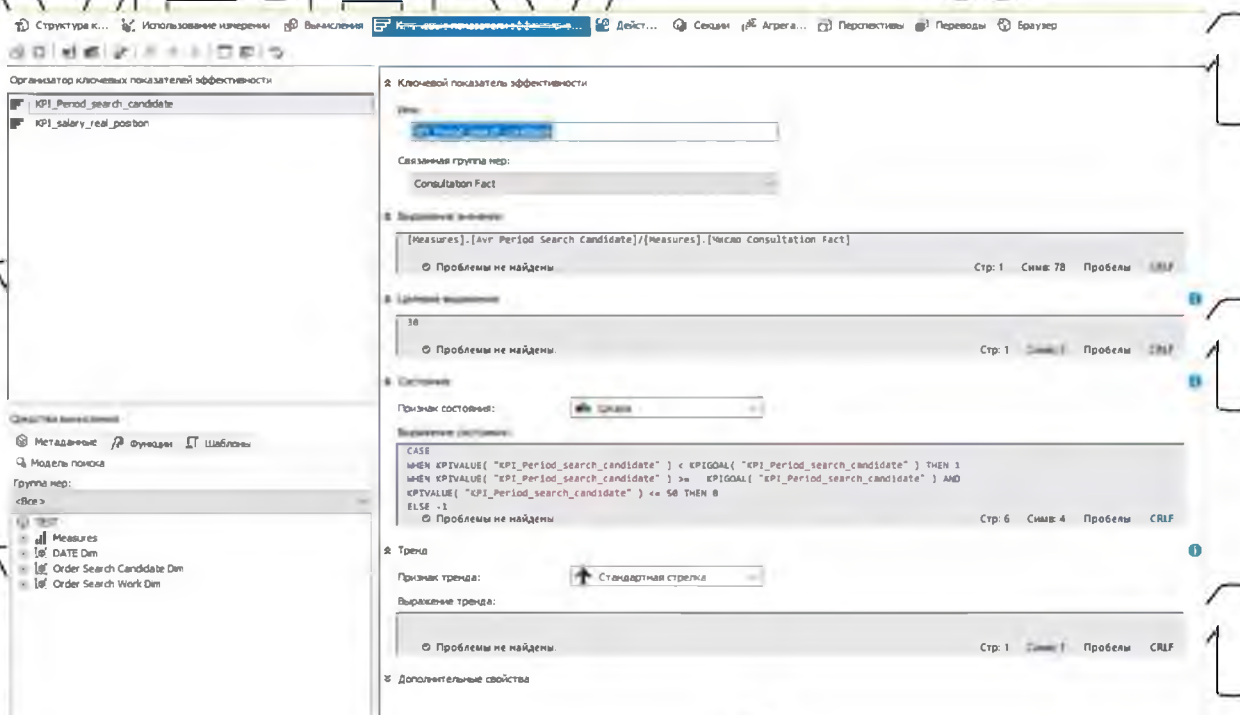


Рис. 4.6 Визначення значення та цілі для показника «Швидкість підбору персоналу»

Образить структуру	Значение	Цель	Состояние
KPI_Period_Search_Candidate	23	30	

Рис. 4.7 Результат обрахунку KPI

В результаті обрання видно, що підбір персоналу має великий попит, про це свідчить зелене забарвлення.

4.3 Алгоритм дерева рішень

Microsoft Decision Trees — це алгоритм класифікації та регресії для використання в прогнозному моделюванні дискретних і безперервних атрибутів. Створюємо нову структуру та обираємо метод, використовуючи створений куб (рис.4.8).

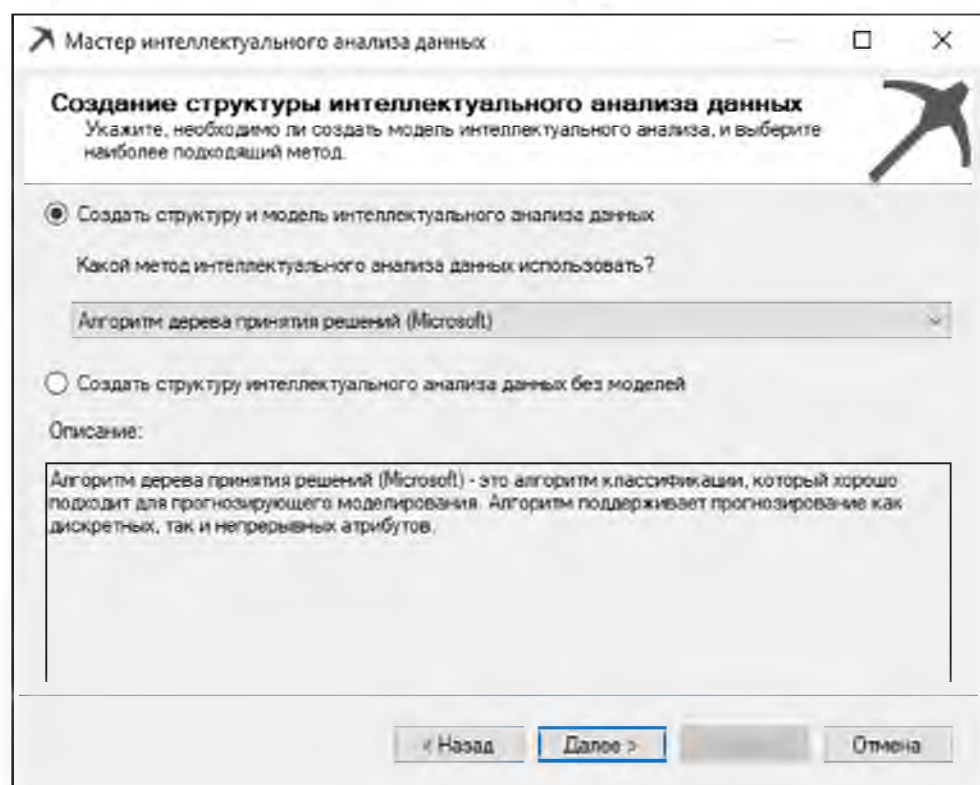


Рис.4.8 Вибір методу, який будемо використовувати

Після того як ми вибрали метод, який будемо використовувати, визначаємо вимір, ключове поле, аналізовані та прогнозовані поля, тип даних структури за допомогою, яких буде відбуватись обробка структури та будуватись графік, хід створення графіку представлено на рис.4.8 - 4.9.

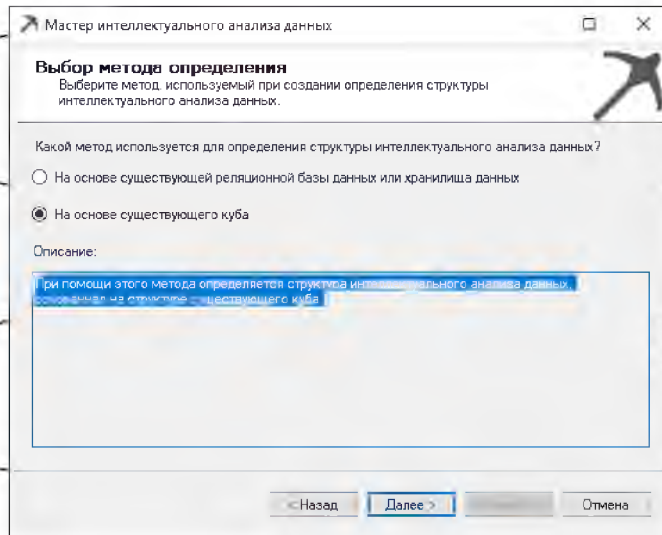


Рис.4.8 Вибраний метод

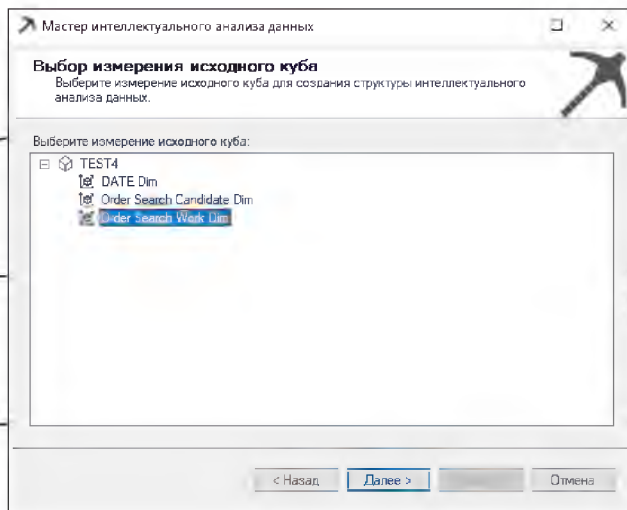


Рис.4.9 Вибраний вимір

Вибираємо ключовий атрибут та визначаємо аналізовані поля (рис.4.10 - 4.11)

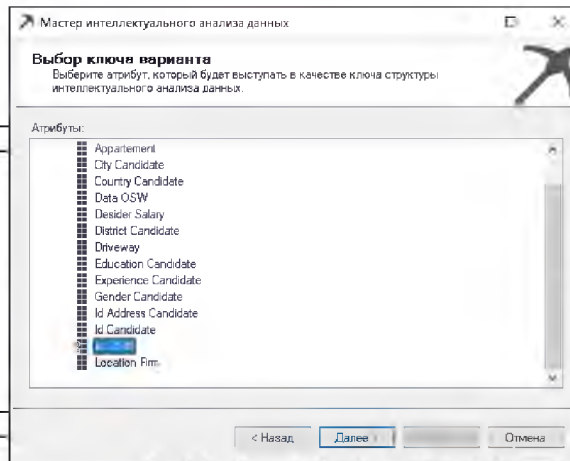


Рис.4.10 Вибираємо ключовий атрибут

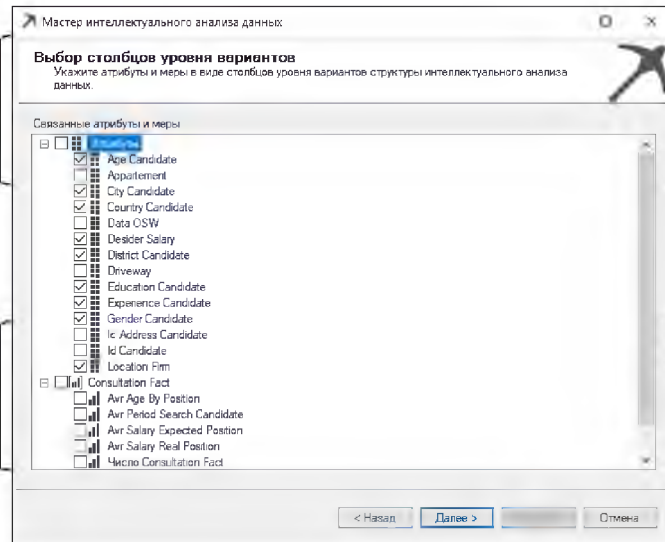


Рис.4.11 Визначаємо аналізовані поля

Визначаємо прогнозовані поля та типи даних (рис.4.12 – 4.13).

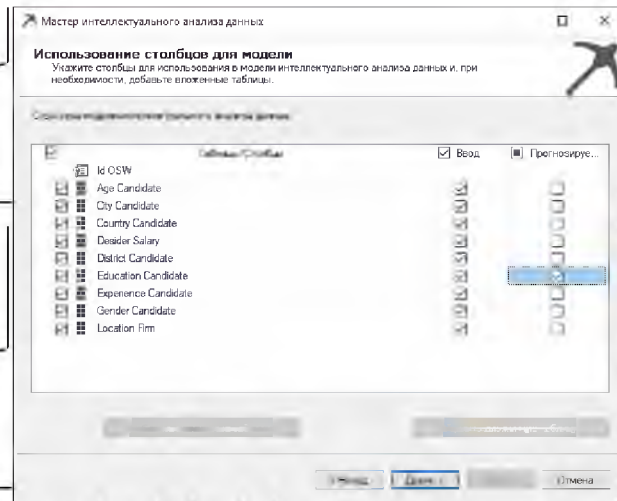


Рис.4.12 Прогнозовані поля

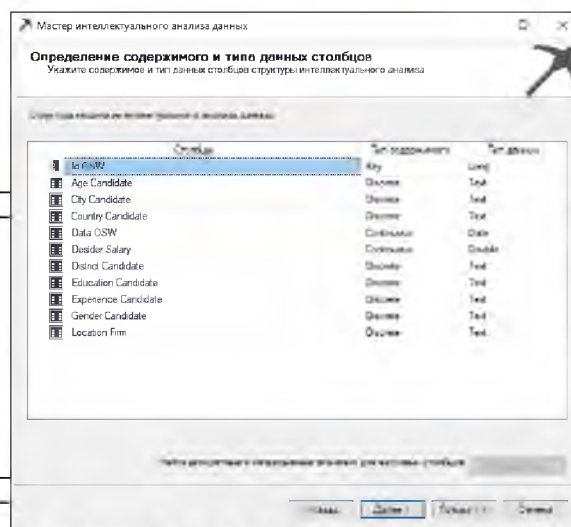


Рис.4.13 Визначаємо типи даних

В результаті отримуємо дерево рішень, яке зображено на рис.4.14.

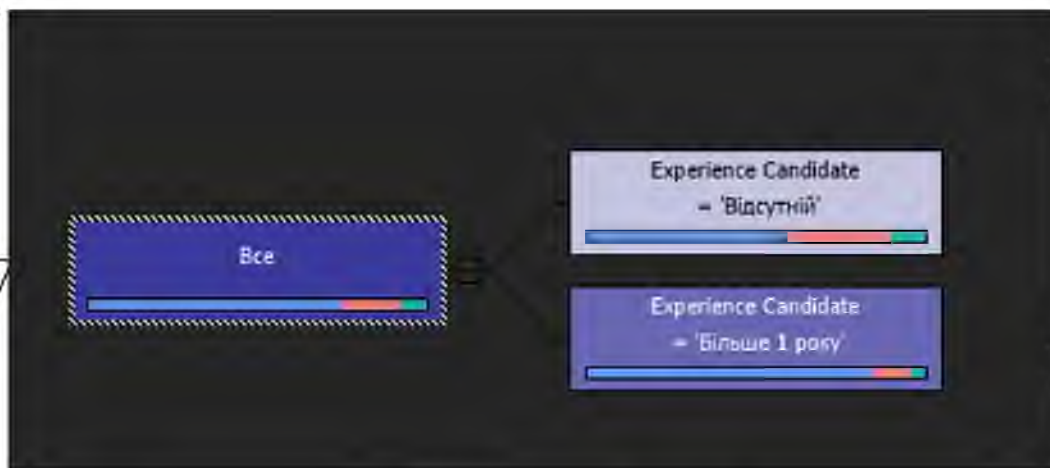


Рис.4.14 Дерево рішення

Порівняння досвіду роботи, коли він відсутній та більше 1 року, представлено на рисунках 4.15 та 4.16

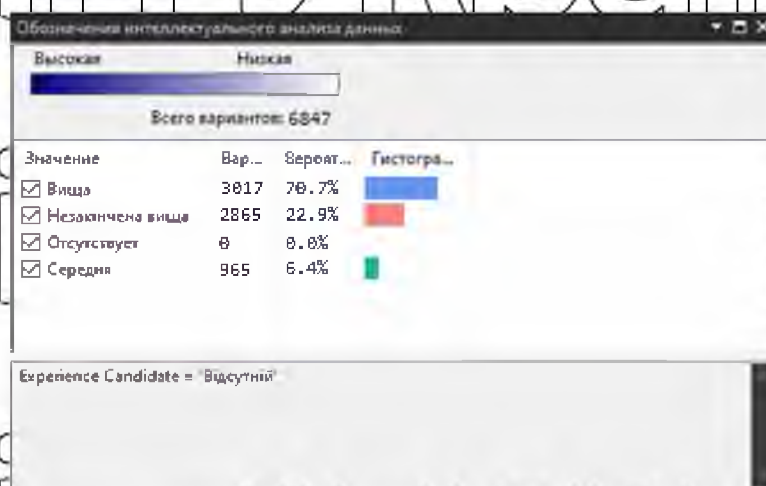


Рис.4.15 Досвід роботи відсутній

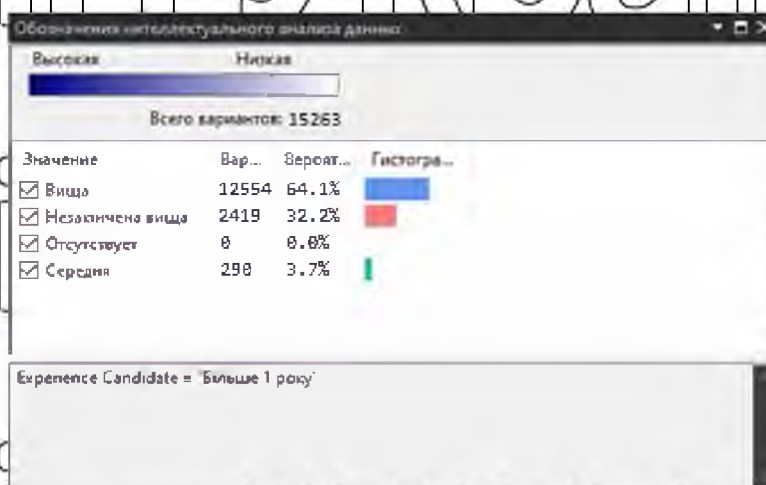


Рис.4.16 Досвід роботи більше року

На основі створеного дерева ми можемо зробити висновок:

• Досвід роботи більше 1 року становить мають 15263 кандидатів, з них:

- 12554 мають вищу освіту;
- 2419 мають незакінчену вищу освіту;
- 290 мають середню освіту.

• 6847 кандидатів не мають досвіду роботи, з них:

- 3017 мають вищу освіту;
- 2865 мають незакінчену вищу освіту;
- 965 мають середню освіту.

• 74% мають досвід роботи більше 1 одного;

• 26% не мають досвіду роботи.

4.4 Спрощений алгоритм Байєса

Microsoft Naive Bayes — алгоритм класифікації, заснований на теоремах Байєса, який може використовуватися як для дослідницького, так і для прогнозного моделювання.

Використовуючи створений нами куб, створюємо нову структуру, рис.4.17

Далі процес створення структури нічим не відрізняється від процесу створення структури в пункті 4.3, процес створення структури можна побачити на рис.4.19-4.13.

НУБІП України

НУБІП України

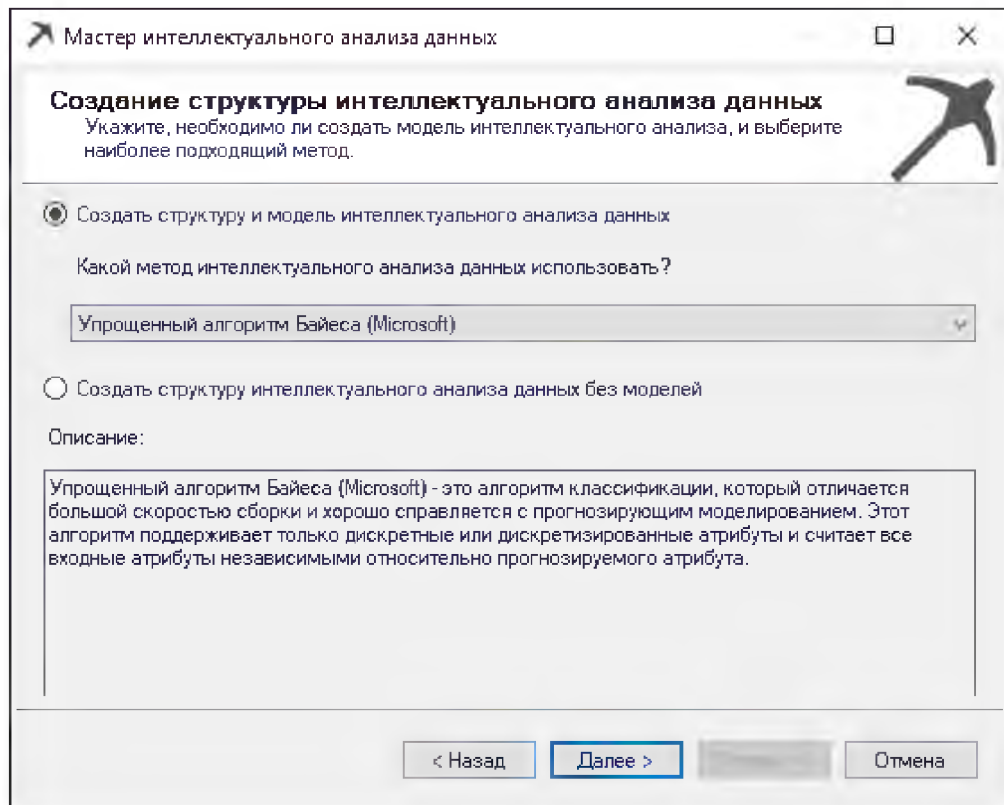


Рис.4.17 Вибір методу

Результатом розгортання структури є отримання мережі залежностей, яка представлена на рис.4.18.

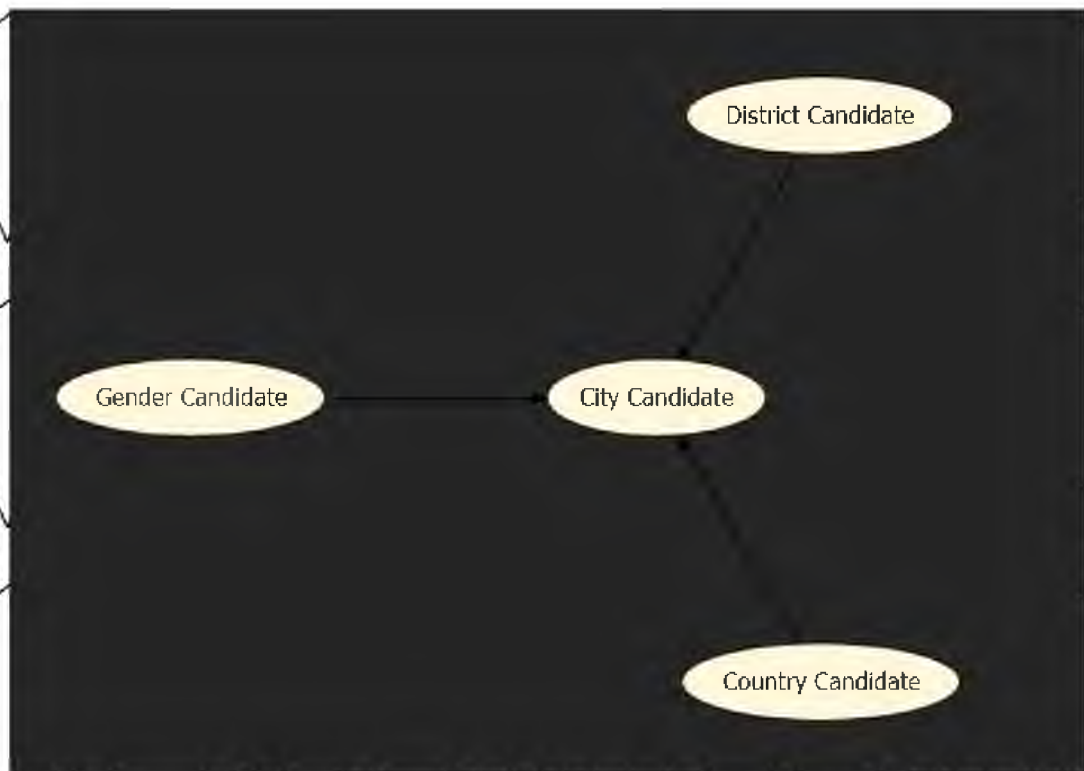


Рис.4.18 Мережа залежностей «Місто»

З профілю атрибуту «Місто» можна побачити, що всі кандидати, які шукають роботу знаходяться в Україні, з них:

- 46.4%(10285) – Київ;
- 9.8%(2157) – Харків;
- 13.4% (2953) Львів;
- 10.8%(2389) – Одеса;
- 9.9%(2188) – Дніпро;
- 3.8%(847) – Запоріжжя;
- 1.1%(249)– Кривий Ріг;

- 1.4%(301) – Миколаїв;
- 1.8%(395) – Тернопіль;
- 1.6%(346)– Луцьк.

Кількість кандидатів, які не мають досвіду роботи становить 31%, з них

складають:

- Чоловіки – 19,7% (4363);
- Жінки – 11,3% (2484).

Кількість кандидатів, які мають досвід роботи більше 1 року становить

69%, з них складають:

- Чоловіки – 40,4% (8925);
- Жінки – 28,6% (6338);

4.5 Алгоритм кластеризації

Clustering Algorithm – це алгоритм кластеризації, який перебирає випадки в наборі даних, щоб групувати їх у кластери, які містять подібні характеристики.

Ці угруповання корисні для вивчення даних, виявлення аномалій у даних і створення прогнозів.

Використовуючи створений нами куб ми створюємо нову структуру,
рис.4.19

Далі процес створення структури нічим не відрізняється від процесу створення структури в пункті 4.3, процес створення структури можна побачити на рис. 4.9-4.13.

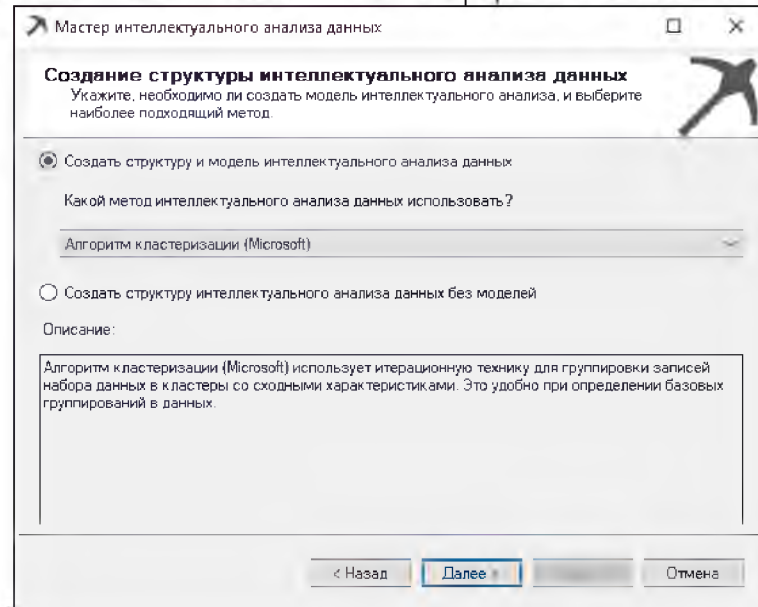


Рис.4.19 Вибір методу

Після обробки структури отримуємо модель даних, що представлено на рис. 4.20.

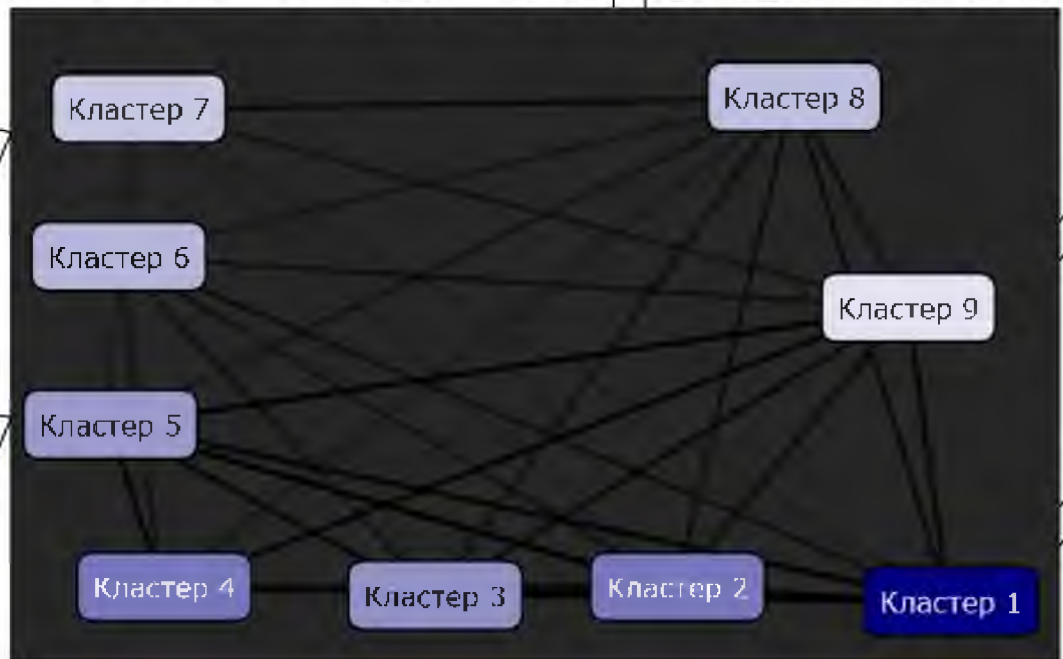


Рис.4.20 Кластеризація всіх обраних даних.

На рисунку 4.21 наведено характеристики усіх кластерів у графічному вигляді.

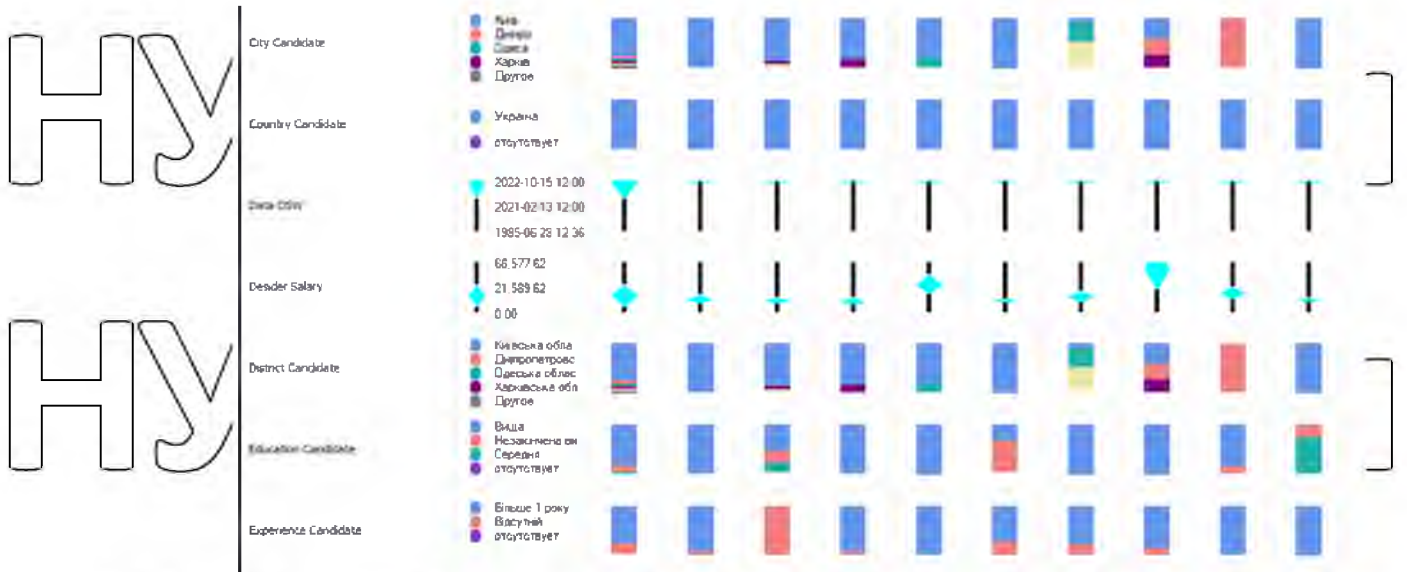


Рис.4.21 Профілі кластерів

З наведеної вище діаграми можна зробити висновок, що найбільше даних зосереджено саме в кластерах 1, 4, 3.

Більш детально розглянемо кластер 1 на рис.4.22.

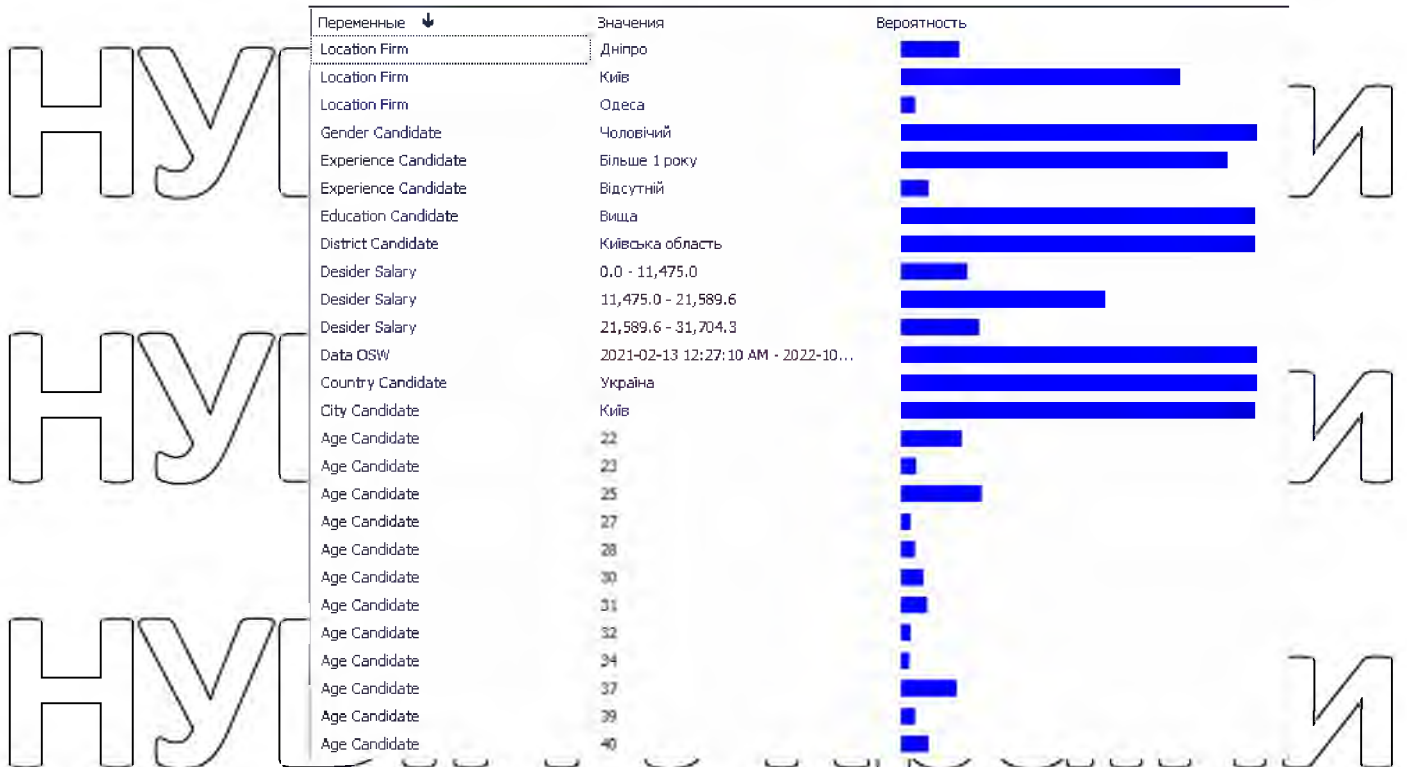


Рис.4.22 Кластер 1

Можна зробити висновок, що кандидат буде взятий на роботу з

ймовірністю

• 79% в місті Київ;

• 17% в місті Дніпро;
 • 4% в місті Одеса.
 З ймовірністю 93% кандидат буде мати досвід роботи більше 1 року та 7% що кандидат не буде мати досвіду роботи. Ймовірність того що кандидат буде мати вищу освіту становить 100%.

Кандидат буде розраховувати на заробітню платню:

- 22000-32000 тис. Гривень з ймовірністю 24%;
- 12000-22000 тис. Гривень з ймовірністю 56%;
- До 12000 тис. Гривень з ймовірністю 20%

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВИСНОВКИ

В ході виконання наукової роботи, було розроблене програмне забезпечення системи підтримки прийняття рішень для агенції з надання послуг в сфері працевлаштування. В першому розділі було проаналізовані існуючі рішення на даний час, актуальність теми та поставлене завдання до розробленої системи.

Наступним кроком було проектування діаграм для предметної області. Для більш детального розуміння для чого потрібна системи, та як відбувається взаємодія з нею було побудовано діаграми прецедентів, послідовності та активності.

Після успішного побудування діаграм для предметної області, було розпочато створення діаграм для моделювання архітектури системи. Було побудовано:

- діаграму компонентів для розуміння які компоненти є в розроблюваної системи;
- діаграму пакетів для розуміння які існують пакети в системі та як вони взаємодіють;
- діаграма розгортання для показу топології системи;
- діаграма класів для показу які існують класи в системі, та які атрибути та методи містить кожен з класів.

Наступною дією було побудування логічної моделі БД, та створення самої БД. БД було реалізовано за допомогою MySQL, на основі якої створено сховище даних.

В ході розробки ПЗ було проведене дослідження за допомогою OLAP-технології та Data Mining. Після моделювання СД було визначено структуру та створено сховище даних. Були побудовані звіти за допомогою BI та пораховано ключові показники ефективності. За допомогою алгоритмів Decision Trees, Naive Bayes, Clustering було проведене дослідження.

За допомогою Decision Trees алгоритму було створено дерево рішень що дозволяє класифікувати дані за досвідом роботи кандидатів.

За допомогою Naive Bayes алгоритму побудовано мережу залежностей, яка демонструє кореляцію між даним.

За допомогою кластерного алгоритму було визначено такі вірогідності: кандидат, який отримає роботу; освітній рівень кандидата, що працевлаштується, середня сума заробітної плати кожної позиції.

В ході аналізу даних за допомогою різних методів було визначено, найпопулярніші міста для працевлаштування, вимоги до кандидатів та середню заробітну плату які допоможуть в прийнятті подальших рішень для покращення роботи агенції.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

СНИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мізерний В. М. Технології пошуку роботи: навчальний посібник / В. М. Мізерний, Л. І. Северин, О. М. Тарасова – Вінниця : ВНТУ, 2014. – 263 с.
2. Літвінчук Л. Світові практики цифровізації надання послуг з працевлаштування / Людмила Літвінчук. // Науковий вісник: Державне управління №4(6). – 2020. – С. 186–197.
3. Хрущ Н. А. Проблеми прийняття управлінських рішень в системі стратегічного управління підприємствами / Н. А. Хрущ, О. С. Корпан, М. В. Желіховська. // Вісник Хмельницького національного університету. – 2010. – № 1, Т. 1 – С. 41–45.
4. Рішення BAS [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bas-soft.eu/soft/bas-mass/bas-integrated-enterprise-management/#popularPosts>
5. Основні компоненти універсальної мови моделювання UML – [Електронний ресурс] – Режим доступу: [https://lubbook.org/book/746/glava/6/Tema/6/Osnovni%D1%96%20komponenti.un.html](https://lubbook.org/book/746/glava/6/Tema/6/Osnovni%20%D1%96%20komponenti.un.html)
6. Призначення діаграма uml. UML - діаграми – [Електронний ресурс] – Режим доступу: <https://shongames.ru/uk/windows-10/naznachenie-diagramm-uml-uml-diagramma-vidy-diagramm-uml-standartnava/>
7. Розробка бази даних — [Електронний ресурс] – Режим доступу: https://otherreferats.allbest.ru/programming/00786860_0.html#text;
8. Visual Studio Code [Електронний ресурс] – Режим доступу: [https://bizzapps.ru/p/vs-code/;](https://bizzapps.ru/p/vs-code/)
9. Что такое HTML? – [Електронний ресурс] – Режим доступу: [https://www.hostinger.ru/rukovodstva/shto-takoe-html/;](https://www.hostinger.ru/rukovodstva/shto-takoe-html/)
10. CSS для починаючих — [Електронний ресурс] – Режим доступу https://www.prostobiz.ua/biznes/biznes_start/statii/css_dlya_nachinayuschih_istoriya_poyavleniya_plyusy_i_minusy;

11. Система управления базами данных MySQL [Электронный ресурс] — Режим доступа: <https://depix.ru/articles/sistema-upravleniya-bazami-dannyh-mysql/>;

12. Online analytical processing [Электронный ресурс] — Режим доступа до ресурсу: <https://learn.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing>;

13. SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных — [Электронный ресурс] — Режим доступа: <https://devacademy.ru/article/sqlite-vs-mysql-vs-postgresql/>;

14. Диаграммы UML: зачем они нужны? — [Электронный ресурс] — Режим доступа: <http://uml3.ru/library/uml-diagrams/uml-diagrams.html>

15. Диаграммы классов — [Электронный ресурс] — Режим доступа: <https://itteach.ru/rational-rose/diagrammi-klassov>;

16. Диаграммы деятельности — [Электронный ресурс] — Режим доступа: https://sites.google.com/site/anisimovkhy/learning/pris/lecture/tema14/tema14_3;

17. Что находится между идеей и кодом? — [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/508710/>;

18. Как использовать PHP для вставки строк в базу данных MySQL — [Электронный ресурс] — Режим доступа: <https://www.internet-technologies.ru/articles/php-dlya-vstavki-strok-v-bazu-dannyh-mysql.html>;

19. Как в PHP загрузить на сайт изображение с изменением веса и размера? — [Электронный ресурс] — Режим доступа: <http://tradebenefit.ru/zagruzka-izobrazheniya-na-sajt-s-izmeneniem-vesa-i-razmera>;

20. Запись в базу данных с помощью php из html формы — [Электронный ресурс] — Режим доступа: <https://fructcode.com/ru/qa/php-zanesenie-dannyh-v-tablicu-cerez-formu-html-y0anlm/>;

21. Как загрузить сайт на хостинг — [Электронный ресурс] — Режим доступа: <https://myrusakov.ru/site-to-hosting.html>;

22. Что такое OLAP — [Электронный ресурс] — Режим доступа:

<https://clickhouse.com/docs/ru/faq/general/olap/>;

23. Ядро OLAP системы — [Электронный ресурс] — Режим доступа:

<https://basegroup.ru/community/articles/olap-core-part1>;

24. OLAP отчеты — [Электронный ресурс] — Режим доступа:

<https://habr.com/ru/post/510750/>;

25. Data Mining — [Электронный ресурс] — Режим доступа:

<https://learn.microsoft.com/en-us/analysis-services/data-mining/data-mining-ssas?view=asallproducts-allversions>;

26. Data Mining Algorithms — [Электронный ресурс] — Режим доступа:

<https://learn.microsoft.com/en-us/analysis-services/data-mining/data-mining-algorithms-analysis-services-data-mining?view=asallproducts-allversions>;

27. Microsoft Data Mining — [Электронный ресурс] — Режим доступа:

<https://www.sciencedirect.com/book/9781555582425/microsoft-data-mining>

28. Data Mining Solution — [Электронный ресурс] — Режим доступа:

<https://learn.microsoft.com/en-us/analysis-services/data-mining/data-mining-solutions?view=asallproducts-allversions>;

29. Data Mining Dimension — [Электронный ресурс] — Режим доступа:

<https://learn.microsoft.com/en-us/analysis-services/data-mining/create-a-data-mining-dimension?view=asallproducts-allversions>;

30. Data Mining Model Viewers — [Электронный ресурс] — Режим

доступу: <https://learn.microsoft.com/en-us/analysis-services/data-mining/data-mining-model-viewers?view=asallproducts-allversions>.