НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ О.Р. Чертов

«___» _____ 2015 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності 6.040301 «Прикладна математика»

на тему: Інформаційна система пошуку вакансій агентства з працевлаштування

Виконала: студентка 4 курсу, групи КМ-11

Зубик Олена Олександрівна _____

Керівник: доцент кафедри ПМА, к.т.н доцент Олефір О.С _____

Консультант з нормоконтролю: ст. викл. Мальчиков В.В. _____

Рецензент: доцент каф. обчислювальної техніки Чебаненко Т.М_____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2015

Національ Національний технічний університет України

«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____О. Р. Чертов

«___»_____2015р..

ЗАВДАННЯ

на дипломну роботу студенту

Зубик Олені Олександрівні

1. Тема роботи «Інформаційна система пошуку вакансій агентства з працевлаштування»

керівник роботи – доцент кафедри ПМА Олефір Олександр Степанович

затверджені наказом по університету від ”19” травня 2015 р. № 1039-С.

2. Термін подання студентом завершеної роботи “12” червня 2015р.

3. Вихідні дані до роботи:

- Уміння користувача;
- Оцінки умінь користувача;

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити):

- Вивчити літературні джерела за тематикою моделей економічного зростання;
- Провести аналіз існуючих програмних рішень у даній галузі;

- Проаналізувати математичне забезпечення, що використовується при пошуку вакансій;
- Спроектувати графічний інтерфейс;
- Розробити програмний продукт, що буде підбирати вакансії для користувача на основі його умінь;
- Оформити документацію до дипломної роботи;

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

- Архітектура програмного забезпечення;
- Структурна схема програми;
- Взаємодія роботи підсистем;
- Екранні форми програми.

6. Консультанти розділів проекту (роботи):

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	ст.викладач Мальчиков В.В.		

7. Дата видачі завдання «28» жовтня 2014

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою роботи	13.11.14	
2.	Проведення порівняльного аналізу математичних методів формування тестових наборів	2.12.14	
3.	Вибір, обґрунтування та опановування методів розв‘язку задачі	8.01.15	
4.	Проектування архітектури розроблюваних програмних засобів	15.01.15	
5.	Визначення складу та форматів вихідних даних та результатів для кожної програми	4.02.15	
6.	Розробка алгоритмів	18.02.15	
7.	Розробка мови управління програмами та проектування користувацького інтерфейсу	18.03.15	
8.	Програмна реалізація	9.04.15	
9.	Розробка алгоритмів та підготовка контрольних задач для їх перевірки	23.04.15	
10.	Розв‘язування контрольних задач на ПЕОМ	13.05.15	
11.	Оформлення документації дипломної роботи		

Студент _____ Зубик О.О.

Керівник роботи _____ Олефір О.С.

АНОТАЦІЯ

Дана дипломна робота виконана на здобуття освітньо-кваліфікаційного рівня “Бакалавр” та присвячено створенню автоматизованої системи пошуку вигідних вакансій для користувача на основі наданих ним умінь.

Метою роботи є створення автоматизованої системи, яка буде підбирати вакансії аналізуючи різницю між вимогами та уміннями користувача.

В рамках дипломної роботи проведено аналіз існуючих програмних систем, що створені для того, щоб спростити процес пошуку вакансій. Розроблено програмне забезпечення, що складається з таких компонентів: серверна частина, база даних, клієнтська частина. Особливу увагу приділено розробці бази даних та надання користувачу можливості працювати з даними, які будуть зберігатися у базі даних.

Дана система є актуальною для компаній, які займаються працевлаштуванням.

Робота виконана на 87 аркушах, містить посилання на список використаних літературних джерел з 10 найменувань. У роботі наведено 31 рисунок, 2 таблиці та 3 додатки.

ABSTRACT

This diploma work is carried out to obtain educational - qualification of "Bachelor" and devoted to the creation of an automated system for finding lucrative jobs given user based on his skills.

The aim is to create an automated system that pick vacancies analyzing the difference between user requirements and skills.

As part of the thesis analyzes existing software systems are designed to simplify the process of finding a job. The software, which consists of the following components: backend, database client. Particular attention is paid to the development of databases and enable users to work with data that will be stored in the database.

This system is important for companies involved in employment.

Work carried out on 87 pages, containing a link to the list of literary sources used 10 names. In work are 31 figures, 2 tables and 3 appendices.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ 10
ВСТУП 11
1 ПОСТАНОВКА ЗАДАЧІ 12
1.1 Опис предметної області 13
1.2 Аналіз існуючих програмних рішень у даній галузі 14
1.3 Висновок до розділу 15
2 ОПИС МОДЕЛІ, ЩО ВИКОРИСТОВУЄТЬСЯ 16
2.1 Теоретичне обґрунтування 16
2.2 Аналітичне представлення моделі 21
2.3 Висновки до розділу 23
3 ПРОЕКТУАННЯ ПРОГРАМНИХ ЗАСОБІВ 24
3.1 Формати вхідних та вихідних даних 24
3.2 Архітектура розроблених програмних засобів 25
3.3 Бізнес - правила 25
3.4 ERD-модель 26
3.5 Ролі та права користувачів 31
3.6 Моделювання бізнес-процесів 38
3.7 Проектування інтерфейсу користувача 41
3.8 Керівництво користувача 46
3.9 Висновки до розділу 48
4 ПРОГРАМНА РЕАЛІЗАЦІЯ 49
4.1 Опис розробленого алгоритму 49
4.2 Структура програмного забезпечення 51
4.3 Опис програми 52
4.4 Опис контрольних прикладів 53
4.5 Висновки до розділу 55
ВИСНОВКИ 56

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 57
ДОДАТКИ 58
Додаток А Лістинг програми 58
Додаток Б Скрипти для створення БД 87
Додаток В Ілюстративний матеріал 90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

DFD (Data Flow Diagram) – модель проектування, графічне представлення «потоків» даних в інформаційній системі.

ERD (Entity Ralationship Diagram) - модель проектування, графічне представлення сутностей і зв’язків між ними.

FK (foreign key) – підмножина атрибутів деякої змінної відношення R2, значення яких повинні збігатися зі значеннями деякого потенційного ключа деякої змінної відношення R1.

IDEF3 (Integrated DEFinition) – методологія моделювання і стандарт документування процесів, що відбуваються в системі.

PFK (primary foreign key) - зовнішній ключ, який став основним.

PK (primary key) – атрибут, або набір атрибутів, що однозначно ідентифікує кортеж даного відношення. Первинний ключ обов'язково унікальний, він єдиний і найголовніший із унікальних ключів.

ІС (інформаційна система)— сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

СКБД (система керування базами даних) — комп'ютерна програма чи комплекс програм, що забезпечує користувачам можливість створення, збереження, оновлення, пошук інформації та контролю доступу в базах даних.

ВСТУП

Одним із найважливіших кроків у житті є вибір професії. Кожна людина повинна робити те, до чого має хист і здібності. Таку працю вона виконуватиме залюбки і з необхідною якістю. Це принесе користь людині і всьому суспільству. Обрати правильно професію – значить обрати таку професію, яка потрібна суспільству, тобто затребувана на ринку праці. Крім того, вона повинна бути доступною для вас, відповідати вашим природним здібностям, знанням і вмінням. І найголовніше – обрана професія має приносити радість, достаток, задоволення.

Коли людина визначається з професією, то наступає наступний, не менш важливий крок, а саме – пошук вакансії. Щоб знайти вакансію, яка буде відповідати усім вимогам та бажанням людини, потрібно витратити досить багато часу на вивчення оголошень у різноманітних друкованих виданнях та Інтернет-джерелах.

Пошук вакансії є досить довгим і нецікавим процесом. Автоматизація даного процесу значно зекономила б час багатьом людям. На сьогоднішній день майже всі знають про автоматизовані системи і в тій, чи іншій мірі використовують їх. Все більше поширюється їх вплив і вони стають частиною нашого життя. Ми використовуємо програмне забезпечення майже кожен день у бізнес справах, таких як банківська справа і в споживчих продуктах, таких як автомобілі та пральні машини. Автоматизація широко використовується у повсякденному житті з метою замінити рутинну роботу і спростити життя людині.

За допомогою автоматизації процесу пошуку вакансій можна систематизувати та впорядкувати великі об’єми інформації, в результаті чого підвищиться ефективність використання часових ресурсів, а також суттєво зменшиться ймовірність прийняття неоптимального рішення.

1 ПОСТАНОВКА ЗАДАЧІ

Мета розробки даної дипломної роботи полягає у створені інформаційної системи, що спростить процедури пошуку вакансії. Користувач матиме змогу підібрати, на основі своїх вмінь, вакансію за доволі короткий час і мінімум зусиль.

Завданням дипломної роботи є розробка клієнт – серверної структури, яка забезпечить цілісність інформації, доступність для роботи клієнту у будь-який момент часу, а також створення зрозумілого та простого інтерфейсу.

Системою можуть користуватися лише зареєстровані користувачі, що встановили даний додаток, вони мають доступ лише до власних даних і не можуть переглядати інформацію про інших користувачів системи, що забезпечує нерозголошення конфіденційної інформації.

Користувач, що шукає вакансію, має право редагувати свій список умінь, додаючи їх із списку запропонованих, або видаляючи деякі з них із наявного сформованого списку.

Адміністратор має дещо інші права. Він може додати нову вакансію або видалити існуючу вакансію. Йому доступний перегляд, редагування списків існуючих вакансій. Адміністратор також має права редагувати, переглядати, додавати та видаляти категорії і уміння. Також адміністратор може переглядати списки зареєстрованих користувачів, їх уміння та підібрані для них вакансії.Для реалізації поставлених цілей система повинна розв’язувати наступні задачі:

- Проведення авторизації та надання прав та привілегій відповідним користувачам;
- Збереження інформації щодо видаленна, додавання або редагування вакансії;

- Збереження інформації, що надається користувачам при реєстрації та оцінки умінь, якими він володіє.

1.1 Опис предметної області

Проблема працевлаштування та зайнятості молоді давно вже є предметом дослідження економістів, соціологів, демографів, психологів та перебуває в центрі уваги державних установ та неурядових організацій. Внаслідок демократизації суспільства відмовилися від напівпримусових форм зайнятості, послаблено державний контроль за працевлаштуванням підвищилася незахищеність молоді на ринку праці, загострилася проблема молодіжного безробіття. Разом з тим зростає роль Державної служби зайнятості у працевлаштуванні випускників, зокрема, професійно-технічних навчальних закладів.

Після закінчення або припинення навчання у загальноосвітніх, професійно-технічних і вищих навчальних закладах, а також після звільнення зі строкової військової або альтернативної (невійськової) служби потрібно витрачати багато часу на пошуки вільної вакансії у друкованих виданнях або проводити дні сидячи за комп'ютером та перечитувати десятки сайтів у пошуках підходящих вакансій.

Тому виникла потреба, у створені автоматизованої системи, яка буде містити усі вільні вакансії та буде підбирати користувачеві ті, що йому підходять. Це значно зменшить час на пошуки роботи, а роботодавці, у свою чергу, зможуть не витрачати кошти на друк оголошень у сотнях газет та журналів.

1.2 Аналіз існуючих програмних рішень у даній галузі

Очевидно, що вже існують програмні рішення, які орієнтовані на подолання даної проблеми, причому їх достатньо багато, але при цьому у них є ряд недоліків. Інфомаційні сиситеми, що намагаються вирішити цю проблему, найчастіше представлені користувачеві у вигляді сайтів. Популярні сайти для пошуку вільних вакансій в Укрвїні:

- rabota.ua;
- trud.gov.ua;
- work.ua;
- job.ukr.net;
- uajobs.com.ua;
- vakansii.online.ua.

Це далеко не повний перелік всіх на сьогоднішній день інформаційних систем для пошуку вільних вакансій. Та усі вони працюють по аналогічній схемі та не аналазують знань користувача. Користувач може лише за допомогою фільтрації відкинути ті, які категорично йому не підходять і після цього повинен перечитати тисячі оголошень у пошуку необхідного. Фільтрацію можна провести по сфері діяльності, тереторіальних характеристиках або розмістити резюме в надії, що роботодавець колись запримітить його. Такий метод пошуку не є ефективним.

Перевага розроблюваної автоматизованої інформаціної системи полягає у тому, що аналізуючи уміння користувача йому надаються вакансії ті, які можуть безпосередньо його зацікавити.

1.3 Висновок до розділу

Метою роботи є створення автоматизованої системи, яка буде підбирати вакансії для користувача на основі його умінь. Дана система буде актуальною для компаній, які займаються працевлаштуванням безробітних, а також звичайних людей які матимуть змогу встановити додаток.

На сьогоднішній день вже існують схожі інформаційні системи, але вони мають ряд недоліків, найбільш суттєві з яких: відсутність аналізу умінь користувача.

2 ОПИС МОДЕЛІ, ЩО ВИКОРИСТОВУЄТЬСЯ

У даному розділі наведено теоретичне обгрунтування вибору моделі та її аналітичне представлення

2.1 Теоретичне обґрунтування

Пошук вигідних вакансій потребає постійний аналіз існуючих вакансій, при чому до уваги мають братись лише актуальні вакансії. Тому дані про вільні вакансії повинні постійно оновлюватись і будь-які зміни повинні бути моментально взяті до уваги. Задля забезпечення таких даних а також можливості паралельного доступу до даних кільком клієнтам краще всього використовувати базу даних, при чому, так як додаток може бути встановлений на персональний комп'ютер, а дані мають постійно оновлюватись, то підключення до БД повинно бути віддаленим. Централізоване накопичення інформації позбавить від надлишковості інформації. Також БД забезпечує цілісність інформації та зручний доступ до даних, що збережена на сервері.

Вакансію не завжди можна однозначно класифікувати до підходящих чи непідходящих для користувача, тому потрібні параметри при аналізі яких можна буде оприділити на скільки вона підхидить. Такими параметрами можуть бути уміння користувача та вимоги до вакансії надані роботодавцем.

Для аналізу вказаних параметріа було реалізовано механізм який визначає на скільки підходить вакансія для користувача.

Способи рішення поставленої задачі:

- За допомогою нечіткої логіки – це дозволить виділити вакансії що найкраще підходять для користувача;

- За допомогою задач лінійного програмування – задача оптимізації;
- За допомогою генетичного алгоритму.

Зада́ча лині́йного програмува́ння — задача оптимізації з лінійною цільовою функцією та допустимою множиною обмеженою лінійними рівностями або нерівностями.

Лінійне програмування (та дослідження задачі лінійного програмування) є однією із найрозвинутиших галузей математичного програмування та теорії оптимізації. Загальна постановка задачі лінійного програмування, та один із підходів до її розв'язання (ідея розрішаючих множників або двоїстих оцінок) вперше наведено в роботі радянського вченого Канторовича Л. В. в 1939. В цій же роботі намічено один із методів розв'язання задачі — метод послідовного зменшення нев'язок.

Методи розв’язання задач лінійного програмування:

•Метод потенціалів — розроблений в 1940 радянськими вченими Канторовичем та Гавуріним Л. В. в застосуванні до транспортної задачі;

•Симплекс-метод — цей метод є узагальненням методу потенціалів для випадку загальної задачі лінійного програмування. Розроблений американським вченим Данциґом Дж.-Б. в 1949 році;

•Двоїстий симплекс-метод розроблений згодом після прямого симплекс-методу, і який є, за сутністю, симплекс-методом розв'язання двоїстої задачі лінійного програмування, але сформульованої в термінах вихідної задачі.

Усі ці методи скінченні. Крім того, існують, також, ітеративні методи розв'язання, які дають можливість обчислювати розв'язки задачі із наперед заданою точністю.

Для розв'язання задач лінійного програмування з великою кількістю змінних та обмежень використовують методи декомпозиції, які дають змогу замість вихідної задачі розв'язувати послідовність задач меншого обсягу.

Методів лінійного програмування недостатньо при накладанні додаткових обмежень на цілочисельність значень змінних. Вивченням таких задач займається цілочисельне програмування.

Методи розв'язання задач лінійного програмування в результаті даюсь оптимальне рішення. Це є повним рішенням поставленої задачі, адже нам потрібні усі вакансії, що підходять для користувача, а не лише оптимальне її рішення [1].

Генети́чний алгори́тм — це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Особливістю генетичного алгоритму є акцент на використання оператора "схрещення", який виконує операцію рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі.

Задача кодується таким чином, щоб її вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи допущені до схрещення (селекція). До осіб застосовується "генетичні оператори" (в більшості випадків це оператор схрещення (crossover) і оператор мутації (mutation), створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (поколінь), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

- знаходження глобального, або надоптимального вирішення;

- вичерпання числа поколінь, що відпущені на еволюцію;
- вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми можуть використатися для пошуку рішень в дуже великих і тяжких просторах пошуку, та у даному випадку складність може створити мутація, в результаті чого можемо отримати неіснуюче рішення.

Генетичні алгоритми мають і низку недоліків. Критерій добору хромосом і використовуваних процедур є евристичним і зовсім не гарантує відшукання «найкращого» рішення. Як і в реальному житті, еволюцію може «заклинити» на якій-небудь непродуктивній галузці. І, навпаки, може статися, що безперспективні батьки, яких вилучить з еволюції генетичний алгоритм, будуть здатні породити високоефективного нащадка. Це особливо стає помітним під час розв'язування багатовимірних задач зі складними внутрішніми зв'язками [2].

Нечітка логіка й теорія нечітких множин - це розділ математики, що є узагальненням класичної логіки і теорії множин. Поняття нечіткої логіки було вперше введене професором Л.Заде в 1965 р. Ним поняття множини було розширено допущенням, що функція приналежності елемента до множини може приймати будь-які значення в інтервалі [0,1], а не тільки 0 або 1. Такі множини були названі нечіткими. Також були запропоновані різні логічні операції над нечіткими множинами й запропоноване поняття лінгвістичної змінної, у якості значень якої виступають нечіткі множини.

Предметом нечіткої логіки є побудова моделей наближених міркувань людини і їх використання в комп'ютерних системах[3].

Використання нечіткої логіки найбільш математично адекватне для рішення проблеми оцінки ризиків в процесі управління ними. Використовуючи нечітку логіку для обробки не детермінованих даних, можна оперувати лінгвістичними змінними, які є більш природними для людського розуміння при описанні елементів економічних систем.

Тобто, за допомогою лінгвістичних змінних можна включити до списку підходящих вакансій, ті які мають незначні відхилення між вимогами роботодавці та уміннями користувача.

Коротко перелічимо відмітні переваги fuzzy-систем у порівнянні з іншими:

- можливість оперувати вхідними даними, заданими нечітко: наприклад, що безупинно змінюються в часі значення (динамічні задачі), значення, що неможливо задати однозначно (результати статистичних опитувань, рекламні компанії і т.д.);
- можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями "більшість", "можливе", переважно" і т.д.;
- можливість проведення якісних оцінок як вхідних даних, так і виведених результатів: ви оперуєте не тільки власне значеннями даних, але їхнім ступенем вірогідності (не плутати з імовірністю!) і її розподілом;
- можливість проведення швидкого моделювання складних динамічних систем і їхній порівняльний аналіз із заданим ступенем точності: оперуючи принципами поведінки системи, описаними fuzzy-методами, ви по-перше, не витрачаєте багато часу на з'ясування точних значень змінних і складання рівнянь, що їх описують, по-друге, можете оцінити різні варіанти вихідних значень.

Сьогодні елементи нечіткої логіки можна знайти в десятках промислових виробів - від систем керування електропоїздами і бойовими вертольотами до пилососів і пральних машин. Рекламні кампанії багатьох фірм (переважно японських) підносять успіхи у використанні нечіткої логіки як особливу конкурентну перевагу. Без застосування нечіткої логіки немислимі сучасні ситуаційні центри керівників західних країн, у яких приймаються ключові політичні рішення і моделюються різні кризові ситуації[4].

Очевидно, що для реалізації поставленої задачі найкраще підходить саме нечітка логіка. За допомогою алгоритму нечіткої класифікації можна реалізувати програмний продукт для пошуку вакансій за допомогою аналізу умінь користувача та вимог до вакансій.

2.2 Аналітичне представлення моделі

Будемо вважаати, щовакансія підходить користувачеві, якщо співпадіння умінь та вимог не менше 90%, майже підходить – 80% і не підходить якщо менше.

Для реалізації поставленої задачі вмористаємо алгоритм нечіткої класифікації, що визначає чи підходить вакансія для користувача за допомогою системи гаусових функцій приналежності:

$$\begin{cases} \mu_{\text{підх}}(m) = \frac{1}{1+\left(\frac{m-0}{0.2}\right)^2}, \text{якщо } m \le 0.1 \\ \mu_{\text{майже підх}}(m) = \frac{1}{1+\left(\frac{m-0.15}{0.1}\right)^2}, \text{якщо } 0.2 \le m < 0.1 \\ \mu_{\text{не підх}}(m) = \frac{1}{1+\left(\frac{m-1}{2}\right)^2}, \text{якщо } 0.2 < m \end{cases} \quad (1)$$

Позначення:

$\mu_{\text{підх}}(m)$ – гаусова функція приналежності для вакансій, що повнісью підходять користувачеві.

$\mu_{\text{майже підх}}(m)$ - гаусова функція приналежності для вакансій, що підходить користувачеві хоча й має деякі відхилення.

$\mu_{\text{не підх}}(m)$ - гаусова функція приналежності для вакансій, що не підходить користувачеві.

m – коефіцієнт приналежності, який обчислюється за допомогою формути 2.

$$\mathrm{m} = \frac{d(x_i, c_\mathrm{i})}{D_{max} \max\limits_{n} d(x_i, c_\mathrm{i})} \quad (2)$$

Позначення:

x_i– уміння вакансії.

c_i- уміння користувача.

i – кількість умінь користувача

D_{max}- можлива найбільша відстань у даному випадку. $D_{max} = 9$ – найбільша можлива різниця між оцінками умінь.

$d(x_i, c_\mathrm{i})$ – відстань між оцінкою уміння користувача та вимогами до вакансій, яка обчислюється за допомогою формули 3.

$$d(x_i, c_\mathrm{i}) = \sqrt{\sum_i (x_i - c_\mathrm{i})^2} \quad (3)$$

2.3 Висновки до розділу

Було описані можливі методи реалізації поставленої задачі та проаналізовано їх недостатки в порівнянні із обраним методом.

В якості математичного апарату було реалізовано алгоритм нечіткої класифікації. Було реалізовано поділ на вакансії, що підходять для користувача, вакансії, що не зважаючи на деякі відхилення підходять для користувача та ті, що не підходять. За допомогою такого поділу з'явилась змога включити у список ті вакансії, які раніше були б не доступні.

3 ПРОЕКТУАННЯ ПРОГРАМНИХ ЗАСОБІВ

У даному розділі наведено результати роботи над проектуванням програмних засобів розроблюваної інформаційної системи. А саме описано формати вхідних та вихідних даних, архітектуру розроблюваної системи, бізнес-правила, проектування інтерфейсу користувача, керівництво та ролі користувачів, а також наведено схеми зв'язків сутностей системи.

3.1 Формати вхідних та вихідних даних

Оскільки програма передбачає можливість роботи з великим обсягом інформації, тодля зручності дані будуть зберігатися та завантажуватися з БД.

З програмної точки зору вхідні та вихідні дані будуть знову ж таки зберігатися у БД.

Вхідні дані системи:

- Особисті дані користувача, а саме ім'я, вік, стать, логін, пароль, уміння та оцінки володіння ними;
- Дані про вакансії: категорія, назва, вимоги (уміння та рівень необхідного володіння ними), адреса електронної пошти, додаткова інформація, розмір мінімальної заробутньої плати.

Вихідними даними системи є:

- Вакансії, які підходять користувачеві.

3.2 Архітектура розроблених програмних засобів

Архітектура розроблюваної системи формально можна поділити на два модулі: база даних та програма користувача.

Дані модулі працюють окремо, тому один із них можна розширювати без шкоди для іншого. Розширення можна проводити як одного модуля так і двох одночасно, це не принесе ніякої шкоди системі.

Опишемо детальніше кожен з модулів:

- Програма користувача – це безпосередньо програма, що запущена на ПК і працює із даними наданими користувачем та взятих із бази даних;
- База даних – зберігає поточну інформацію про вакансії та користувачів. Дана БД може розміщуватися як на тому ж ПК, що і сама система, так і на віддаленому сервері, залежно від потреб користувача;

3.3 Бізнес - правила

Бізнес-правила – правила, які чітко визначені розробником користувачеві, під час користування його програмою[6].

Автоматизована інформаційна система відслідковує життєвий цикл наступних об’єктів:

Користувач – створюється при реєстрації в системі. Відповідно до професійних навичок працівників, система формує список підходящих вакансій.

В разі видалення користувача із системи, каскадно видаляються записи із таблиць претендентів на вакансію та умінь користувача.

Вакансія – створює, редагує і видаляє адміністратор. Відповідно до професійних вимог, система формує списки підходящих користувачів.

В разі видалення із системи, каскадно видаляються записи з таблиць претендентів на вакансію та вимог до навичок.

Категорії умінь – створює, редагує і видаляє виключно адміністратор бази даних. Необхідно для групування умінь по сферах застосування.

Забороняється видаляти непорожні категорії.

Уміння – створює, редагує і видаляє виключно адміністратор бази даних. Використовується для опису професійних навичок користувачів та вимог до кандидатів на вакансію.

Забороняється видаляти вміння, які використовуються.

3.4 ERD-модель

Модель "сутність-зв'язок" (ER-модель) - модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. Вона зручна при проектуванні інформаційних систем, баз даних, комп'ютерних архітектур та інших систем (моделей).

Існує ряд моделей для представлення знань. Одним з найзручніших інструментів уніфікованого представлення даних, незалежного від реалізовуючого його програмного забезпечення, є модель "сутність-зв'язок" (entity - relationship model, ER - model).

Модель "сутність-зв'язок" ґрунтується на якійсь важливій семантичній інформації про реальний світ і призначена для логічного представлення даних. Вона визначає значення даних в контексті їх взаємозв'язку з іншими даними. Будь-який фрагмент наочної області може бути представлений як безліч сутностей, між якими існує деяка безліч зв'язків [7].

У розроблюваній інформаційній системі наявні наступні сутності:

Users – сутність потрібна для зберігання інформації про користувачів.

Free vacancy – сутність потрібна для зберігання інформації про вільні вакансії.

Skills – сутність потрібна для зберігання інформації про уміння користувачів.

Skills and vacancy – сутність потрібна для опису вимоги до кандидатури на вакансію, уміння якими він має володіти.

Users and skills – сутність використовується для опису умінь користувача та рівня володіння ними.

Suitable vacancy for users – сутність необхідна для зберігання вакансій, що були підібрані для користувача.

Атирибути сутностей наведені у таблиці 3.1. Зв'язки між сутностями наведено у вигляді діаграми на рис. 3.1.

Таблиця 3.1 – Атрибути сутностей системи

Назва сутності: назва атрибуту	Код	Тип	Опис	PK
Users: login	user_login	Varaible characters(20)	Логін користувача для входу в систему	+

Продовження таблиці 3.1

Назва сутності: назва атрибуту	Код	Тип	Опис	PK
Users: password	user_password	Varaible characters(20)	Пароль користувача для входу в систему	-
User: name	user_name	Varaible characters(15)	Ім'я користувача	-
User: age	user_age	Namber(2)	Вік користувача	-
User: sex	user_sex	Varaible characters(1)	Стать користувача	-
Free vacancy: name	vacancy_name	Varaible characters(30)	Назва вакансії	+
Free vacancy: tell	vacancy_tell	Varaible characters(15)	Номер телефону	-
Free vacancy: email	vacancy_ email	Varaible characters(20)	Email	+
Free vacancy: info	vacancy_ info	Varaible characters (254)	Додаткова інформація	-
Free vacancy: salary	vacancy_ salary	Namber	Початкова заробітня плата	-

Продовження таблиці 3.1

Назва сутності: назва атрибуту	Код	Тип	Опис	PK
Free vacancy: category	vacancy_ category	Varaible characters(20)	Категорія, до якої належить вакансія	-
Skills: name	skills_name	Varaible characters(30)	Назва уміння	+
Skills: category	skills_ category	Varaible characters(20)	Категорія, до якої належить уміння	-
Skills and vacancy: vacancy name	vacancy_name	Varaible characters(30)	Назва вакансії	+
Skills and vacancy: vacancy email	vacancy_ email	Varaible characters(20)	Email	+
Skills and vacancy: skill name	skills_name	Varaible characters(30)	Назва уміння	+
Skills and vacancy: vacancy skill level	vacancy_skill_l evel	Namber	Оцінка уміння, що відповідає вимозі до знань майбутньо го працівника	-

Продовження таблиці 3.1

Назва сутності: назва атрибуту	Код	Тип	Опис	PK
Users and skills: user login	user_login	Varaible characters(20)	Логін користувача, що шукає роботу	+
Users and skills: skills name	skills_name	Varaible characters(30)	Назва уміння	+
Users and skills: user skill level	user_skill_level	Number	Рівень вородіння умінням (від 1 до 10)	-
Suitable vacancy for users: vacancy name	vacancy_name	Varaible characters(30)	Назва вакансії	+
Suitable vacancy for users: vacancy email	vacancy_ email	Varaible characters(20)	Email	+
Suitable vacancy for users: user login	user_login	Varaible characters(20)	Логін користувача, що шукає роботу	+

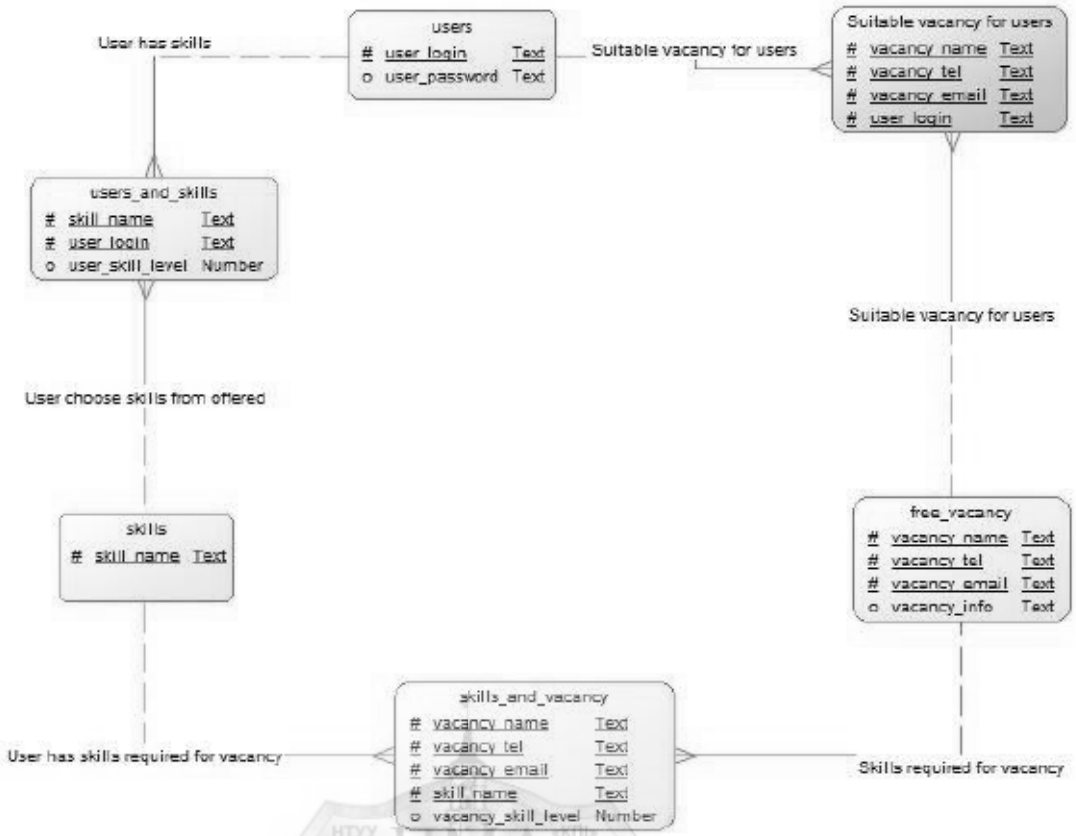


Рисунок 3.1 – ERD - модель системи

3.5 Ролі та права користувачів

Дана система передбачена для користувачів різного рівня володіння ПК, тому вона повинна бути інтуїтивно зрозумілою та має забезпечувати цілісність інформації. Тобто, дії користувача повинні обмежуватись не лише за допомогою надання певної ролі, але й програмно, тобто, за допомогою різноманітних обмежження на ввод даних. У системі передбачено два види користувачів: адміністратор і звичайний користувач.

а) Користувач може виконати наступні дії:

- Реєстрація:
  - логін < 21 символів, пароль < 21 символа;

- поля «Логін» та «Пароль» не можуть бути пустими;
- поле «Ім'я» може містити до 15 латинських літер;
- у полі «Стать» користувач обирає один із запропонованих варіантів;
- поле «Вік» заповнюється лише цифрами (2 цифри).

– Авторизація:

- логін < 21 символів, пароль < 21 символа;
- поля «Логін» та «Пароль» не можуть бути пустими;
- поля «Логін» та «Пароль» можуть містити лише цифри та латинські літери;
- в систему можуть ввійти тільки зареєстровані користувачі.

– Формування списку умінь:

- користувач обирає категорію і отримує перелік умінь, що відповідають даній категорії, після чого користувач повинен оцінити свої знання кожного з умінь категорії по шкалі від 1 до 10;

– Фільтрація списку вакансій:

- користувач може виконати фільтрацію по заробітній платі.

b) Адміністратор може здійснювати наступні дії:

– Додавання нової вакансії;

– Редагувння існуючої вакансії (не підлягають редагуванню поля «Категорія вакансії», «Назва вакансії», «Email»);

– Видалення існуючої вакансії.

Список подій, що будуть доступні кожній з груп вище наведених користувачів наведено у вигляді матриці елементарних подій (ELM, табл. 3.2). Список подій (ELM) - матриця, яка описує дії зовнішніх сутностей і реакцію системи на них [8].

Таблиця 3.2 – Матриця подій

№	Опис події	Тип події	Реакція на подію
1	Користувач бажає здійснити реєстрацію і натискає на кнопку «Реєстрація»	N	Надати реєстраційну форму
2	Користувач заповняє реєстраційну форму	N	Надати можливість заповнення полів реєстраційної форми
3	Користувач натискає на кнопку «Здійснити реєстрацію»	N	Перевірити коректність заповнення полів, якщо допущено помилки або у разі наявності вказаного логіну у системі, повідомити про це користувача та попросити внести зміни. Перевірити на наявність заповнених обов'язкових полів, якщо хоча б одне із них залишилось пустим – повідомити про це користувача та залишитись на формі реєстрації. Якщо претензій немає, здійснити реєстрацію та перехід на форму безпосереднього пошуку вакансій

Продовження таблиці 3.2

№	Опис події	Тип події	Реакція на подію
4	Користувач ввівши логін і пароль натискає кнопку «Увійти»	N	Перевірити наявність вказаного логіну та паролю БД. Якщо співпадінь немає то відмовити у вході в систему та повідомити про це користувача. Якщо вказані дані містяться у БД, дозволити вхід в систему
5	Користувач відкриває категорію умінь	N	Відкрити список умінь що відповідає обраній користувачем категорії
6	Користувач оцінює уміння по шкалі від 1 до 10 відносно рівня володіння даним умінням	N	Внести уміння та оцінку їх володіння до бази даних
7	Обрана користувачем категорія не містить списку умінь	N N	Повідомити адміністратора
8	Користувач бажає підібрати вакансію відносно введених ним умінь	N	Надати перелік вакансій, що відповідає умінням користувача
9	Користувач хоче повернутись і змінити вказані ним уміння	N	Надати можливість повернення до попередньої форми, дозволити користувачеві редагувати дані таблиці, перевірити коректність введених змін.

Продовження таблиці 3.2

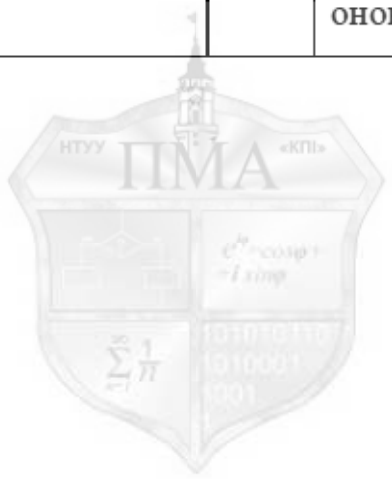
№	Опис події	Тип події	Реакція на подію
10	Відсутність зв'язку з БД	N N	Спробувати підключитись до БД. Перезапустити скрипти БД.
11	Користувач бажає фільтрувати список	N	Надати можливість користувачеві на заповнення полів фільтрації, перевірити коректність заповнених полів. У разі некоректного введення видати повідомлення про некоректне заповнення. У разі правильного заповнення полів фільтру врахувати надані користувачем умови та видати відфільтрований список вакансій
12	Користувач хоче вийти із системи	N	Надати користувачеві можливість вийти із свого профілю
13	Користувач хоче вийти з програми	N	Надання користувачеві можливості закрити програму
14	Адміністратор входить в систему під зарезервованим логіном та паролем	N	Надати форму для роботи з програмою адміністратора

Продовження таблиці 3.2

№	Опис події	Тип події	Реакція на подію
15	Адміністратор виконує фільтрацію обравши одну із запропонованих карегорій та натискає кнопку «Фільтрувати»	N	Відфільтрувати список вакансі з урахуванням заданого адміністратором параметра фільтрації
16	Адміністратор обирає одну із вакансій і натискає кнопку видалити	N	Видалити дані про обрану вакансію із представленого списку та БД
17	Адміністратор обирає одну із вакансій та натискає кнопку редагування	N	Відкрити вікно для редагування із даними, що були надані про вакансію. Користувач не може змінювати категорію до якої належить вакансія, назву вакансії та email. Всі інші поля при редагуванні перевіряються на коректність вводу. Якщо усі дані були вірно введені, то внесені зміни заносяться до БД. Адміністратору надається оновлений список вакансій

Продовження таблиці 3.2

№	Опис події	Тип події	Реакція на подію
18	Адміністратор натискає кнопку «Додати вакансію»	N	Надати форму для додавання вакансії. Після натиску на кнопку «Додати вакансію» введенні дані перевіряються на коректність. Якщо всі поля заповнені вірно – дані про вікансію заносяться до БД, а вдміністратору надається оновлений список вакансій



3.6 Моделювання бізнес-процесів

Моделювання бізнес-процесів — це процесове відображення (як правило графічне) діяльності підприємства з тим, щоб в подальшому дані процеси можна було аналізувати і вдосконалювати. Взаємозв'язана система бізнес-процесів зображає весь комплекс завдань і функцій структурних підрозділів, виконання яких необхідно забезпечити в процесі діяльності інформаційної системи. Графічний опис бізнес-процесів та їх імітація це методи аналізу бізнес-процесів, ефективність яких доведена багаторічною практикою використання та численними дослідженнями.

Для кращого розуміння технологічних процесів пошуку вакансій для користувача (того, що відбувається на кожному етапі технологічного процесу) та переходів станів об'єктів, із урахуванням існуючих проміжних станів у об'єктів в системі було побудовано IDEF3 (рис.3.2) [9].

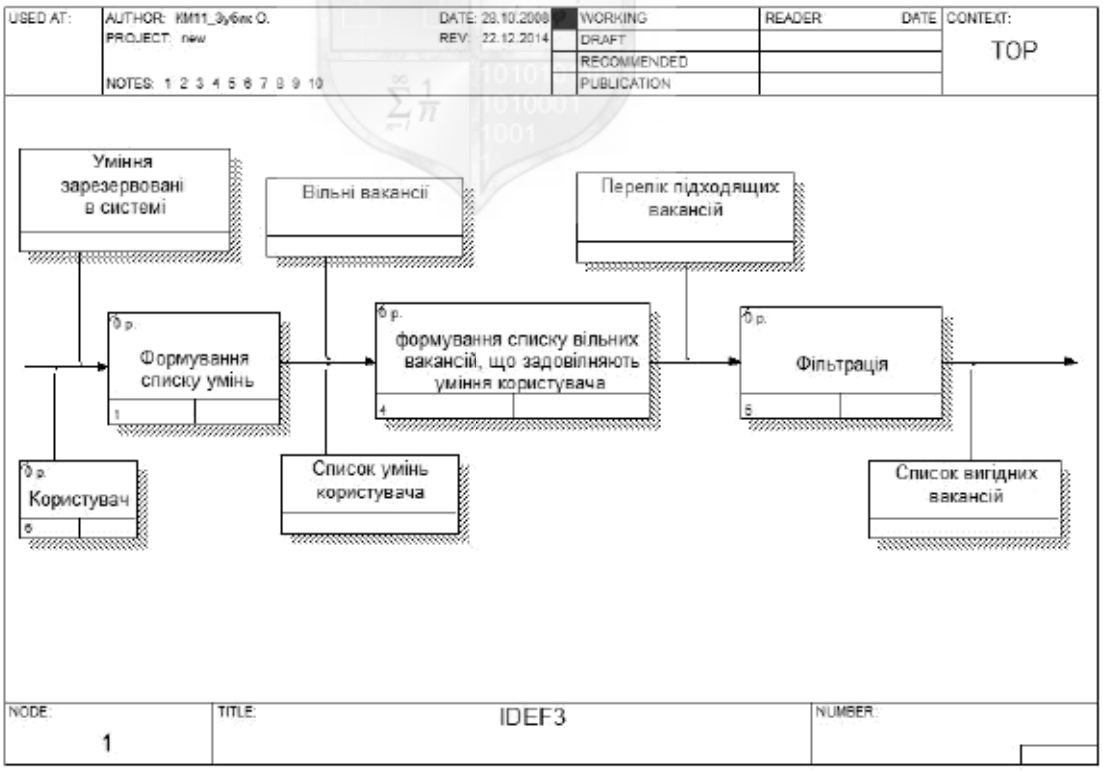


Рисунок 3.2 - IDEF3 процесу пошкуку вакансій для користувача

Для демонстрації того, як кожен процес перетворює вхідні дані у вихідні нижче наведено опис потоків даних DFD [10].

3.6.1 DFD

Основний процес системи – пошук вигідних вакансій для користувача опитаючись на його вміння. Для виконання завдання необхідна наступна інформація від зовнішніх сутностей:

a) про уміння користувача

b) про вільні вакансії та їх вимоги

Узагальнена схема потоку даних представлена DFD0 і зображена на рис.3.3

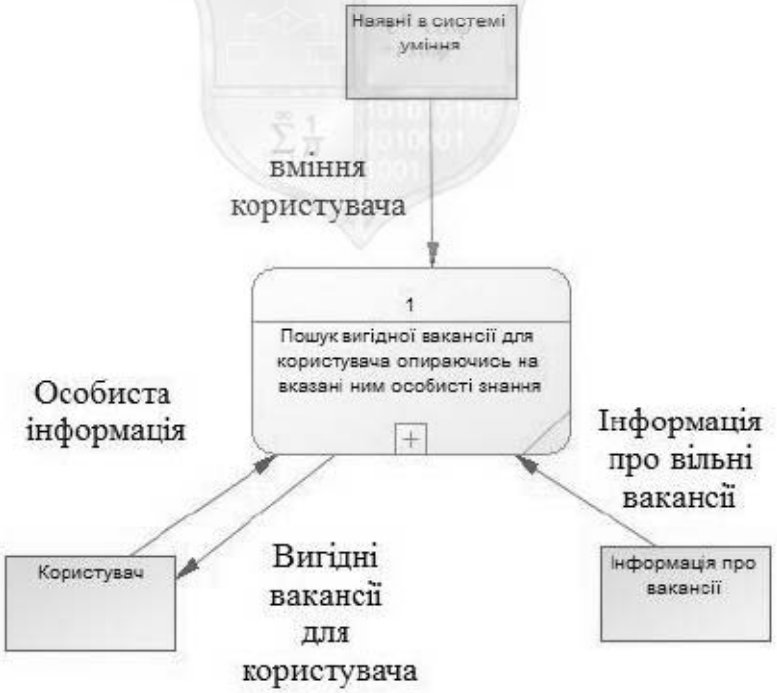


Рисунок 3.3 - DFD0 процесу пошуку вільних вакансій для користувача

Більш детальна схема потоку даних представлена DFD1 і зображена на рис.3.4 Маємо наступні підпроцеси:

a) формування списку умінь користувача. Потребує від користувача обрати із наявних у системі умінь ті, якими він володіє;

b) формування списку вільних вакансій на основі сформованого списку вмінь. Потребує відомості про наявні в системі вакансії та їх вимоги;

c) фільтрування списку потенційних вакансій за обраними користувачам критеріями.

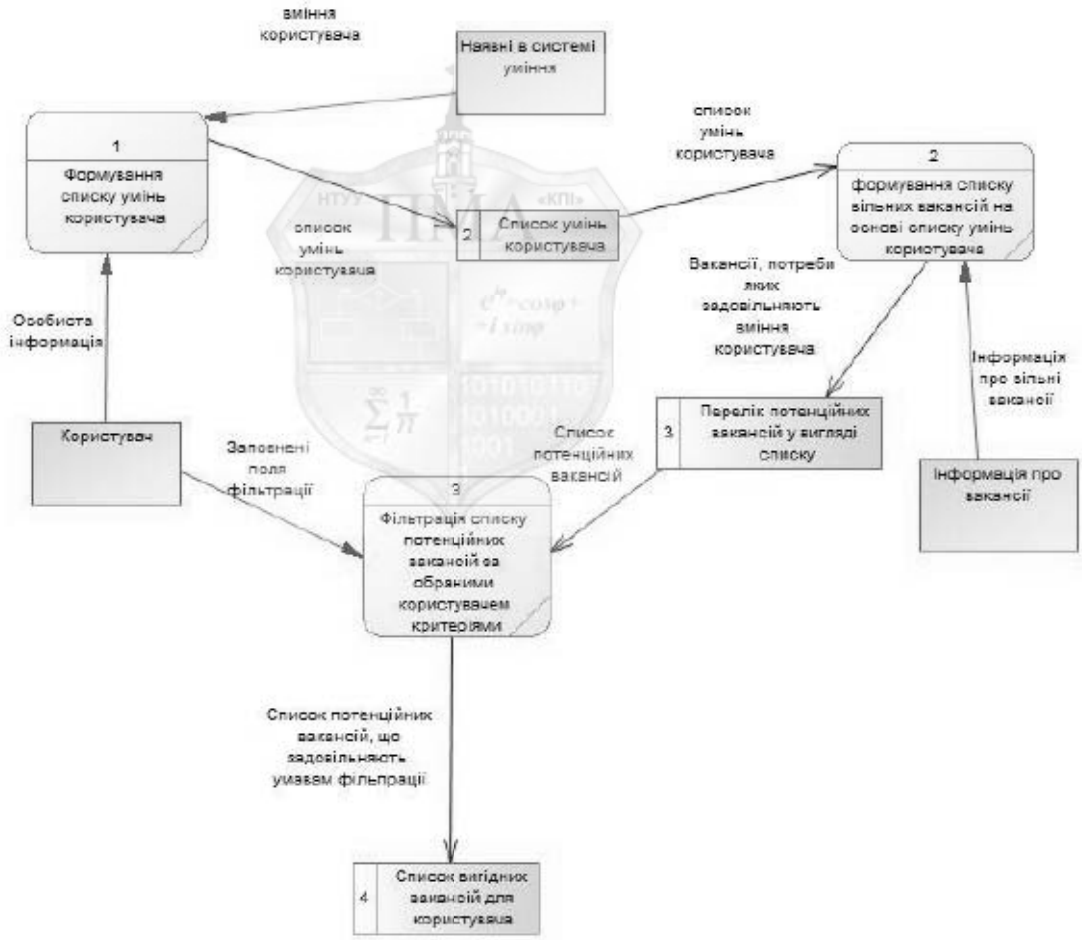


Рисунок 3.4 – DFD1 процесу пошуку вільних вакансій для користувача

3.7 Проектування інтерфейсу користувача

Програма розроблена на мові програмування C# у середовищі Visual Studio 2013.

При запуску програми для користувача надається список усіх вакансій, які він може фільтрувати по розміру заробітньої плати. Дана форма наведена на рис. 3.5.

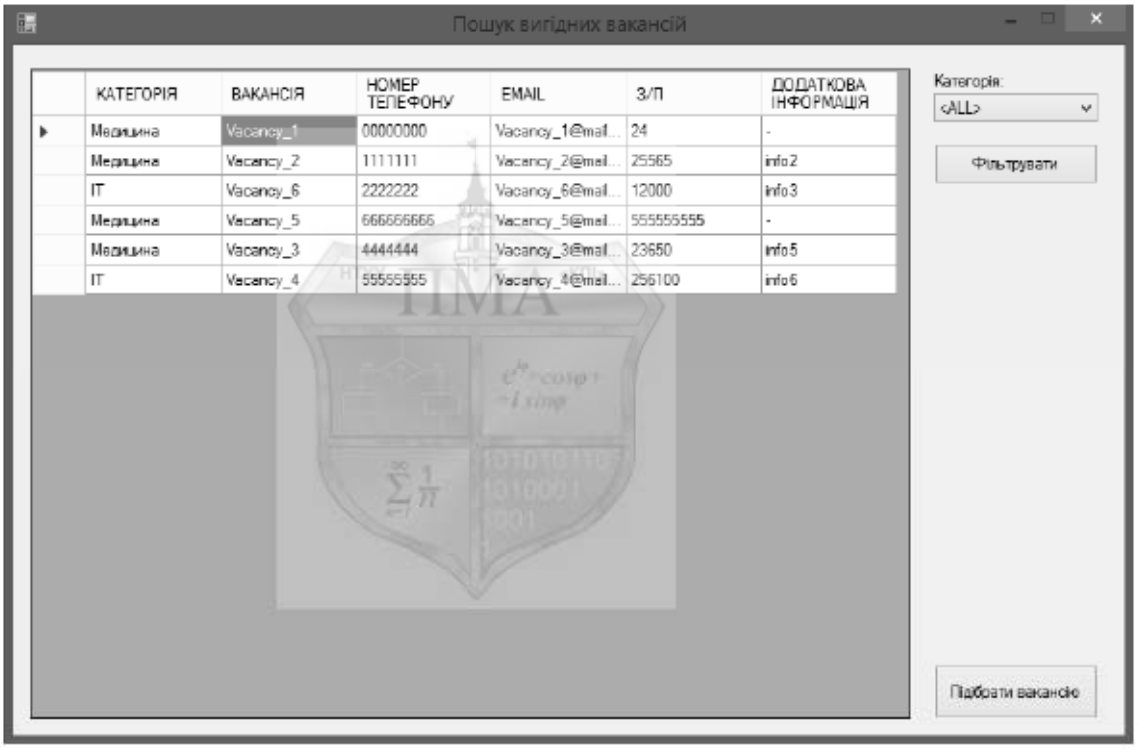


Рисунок 3.5 – Форма представлення існуючих вільних вакансій

При натиснені на кнопку «Підібрати вакансію» з'являється вікно авторизації користувача приведена на рисунку 3.6

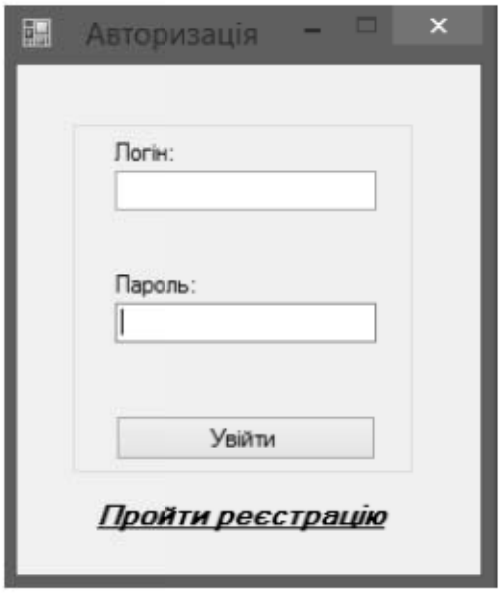


Рисунок 3.6 – Форма авторизації користувача

Якщо користувач вперше у системі, то він може пройти реєстрацію натиснувши на ссилку «Пройти реєсрацію» і заповнивши форму зображену на рис. 3.7.

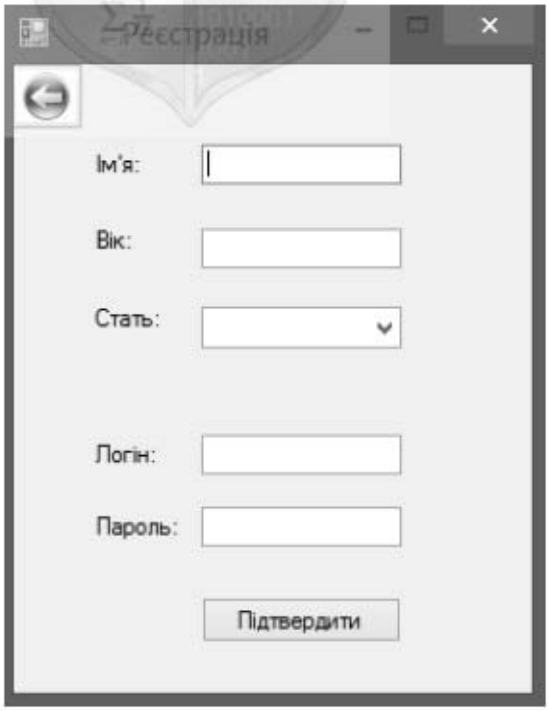


Рисунок 3.7 – Форма реєстрації користувача

Авторизованому користувачеві надається перехід на основну робочу форму, що наведена на рисунку 3.8, де користувач повинен оцінити свої уміння у обраній категорії.

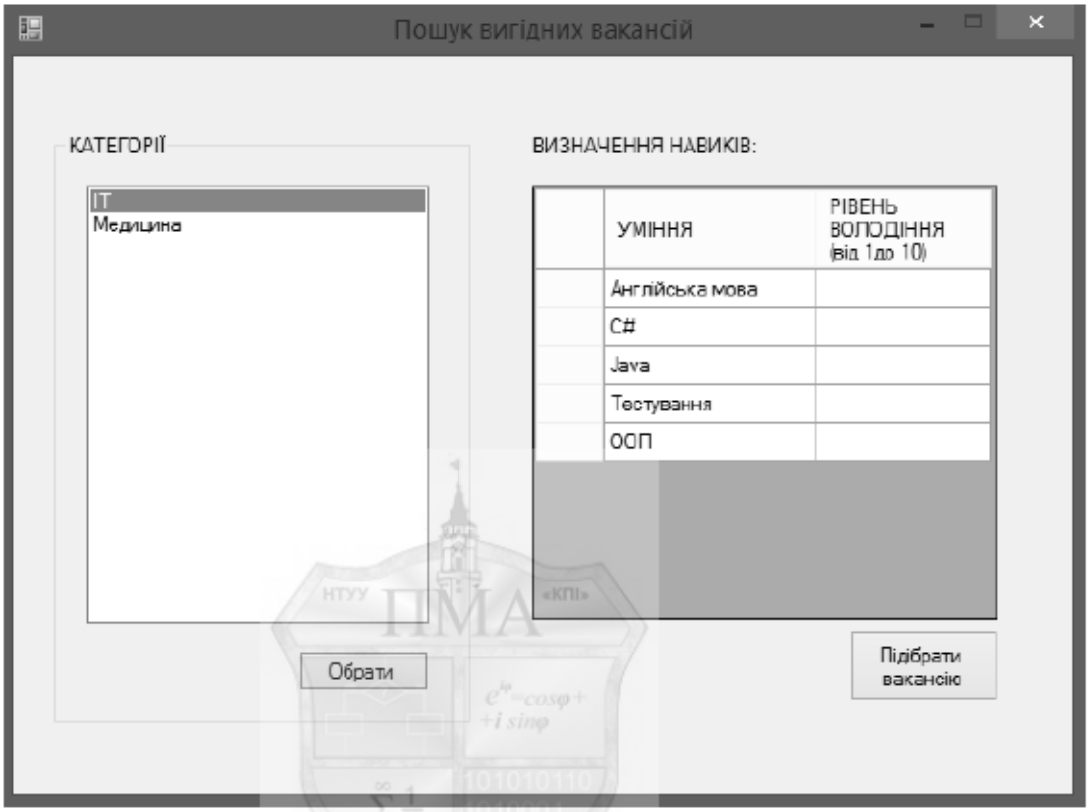


Рисунок 3.8 – Головне вікно системи

Після заповнення усіх полів виконується підбір вакансій та виводяться результати пошуку у відповідному вікні, що зображено на рис.3.9.

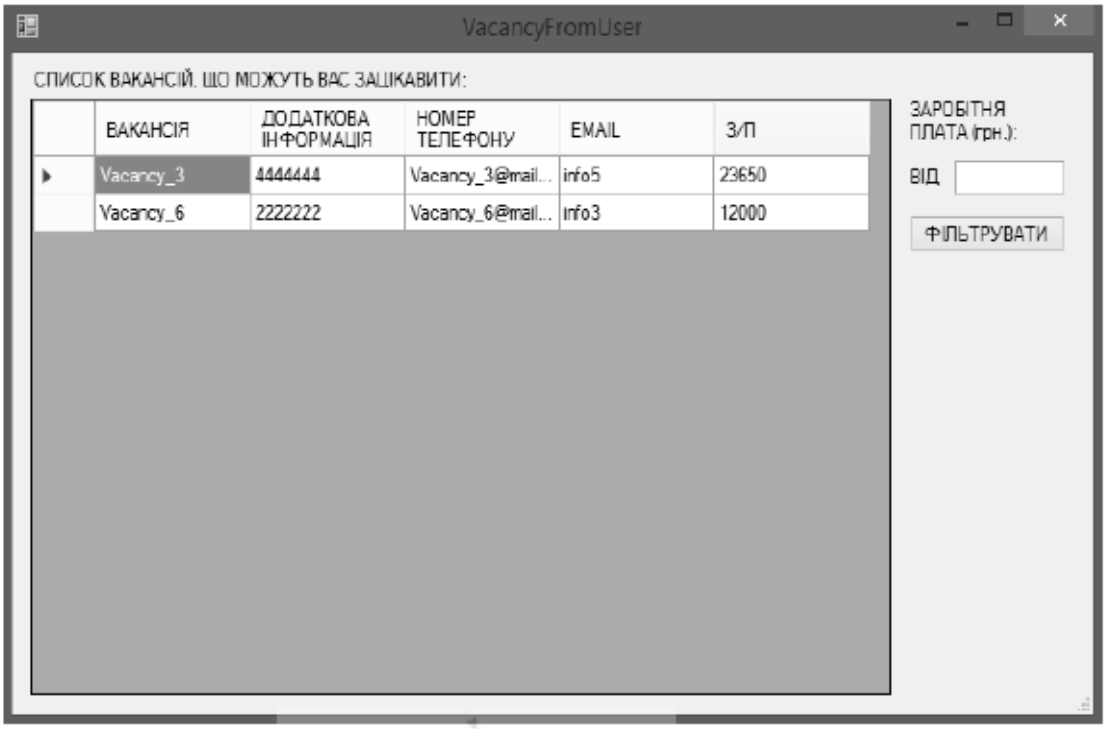


Рисунок 3.9 – Вікно результатів підбору вакансій

При авторизації адміністратора системи відкривається головне вікно для обробки вакансій, що зображено на рис.3.10.

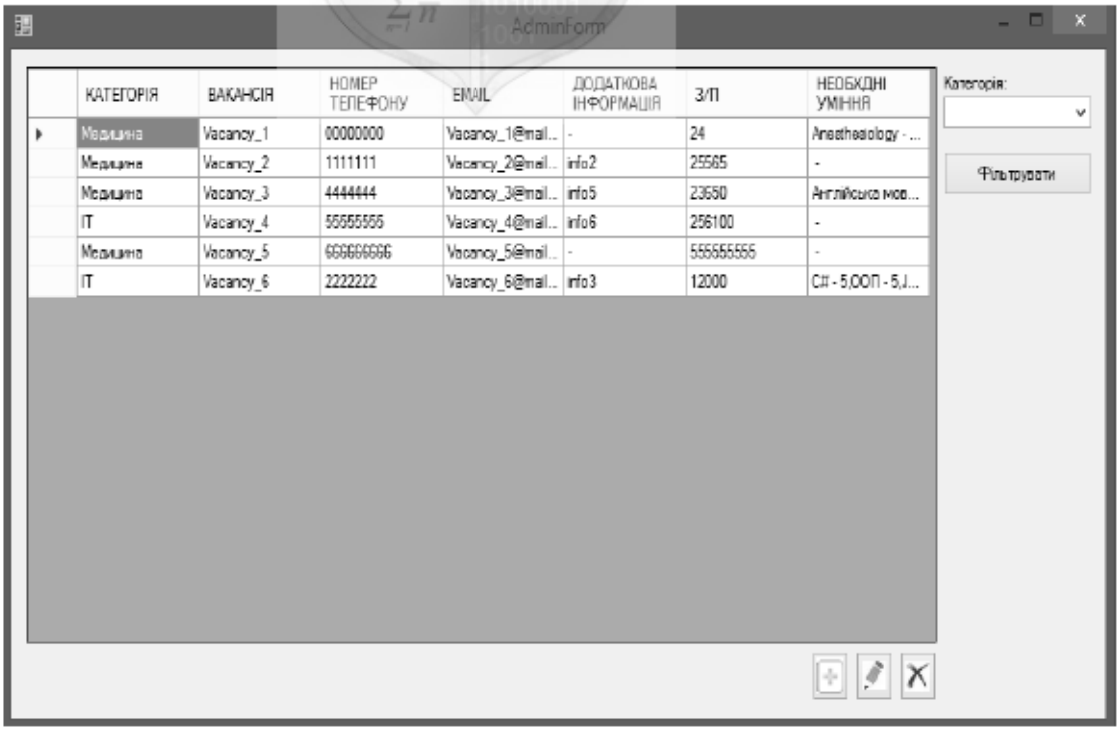


Рисунок 3.10 – Головне вікно для роботи адміністратора

Для швидкого пошуку потрібної вакансії потрібно скористатись полем фільтрації.

Адміністратор може редагувати, додавати а також видаляти вакансію.

Натиснувши кнопку ⊞ - додати вакансію відкривається вікно зображене на рис. 3.11.

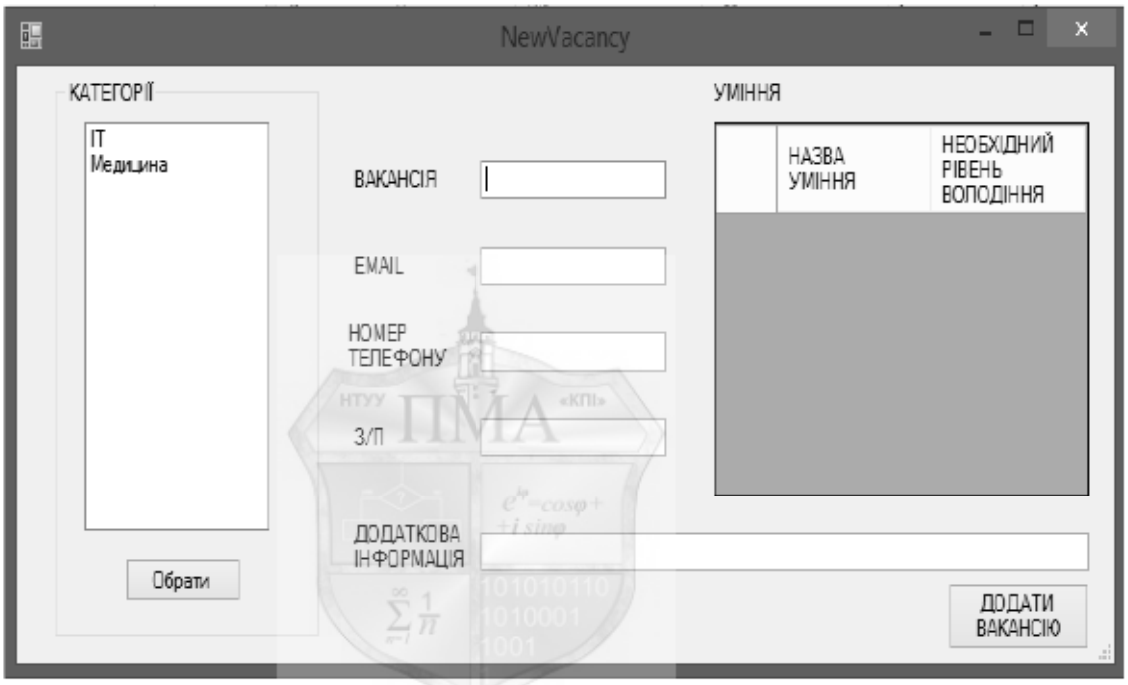


Рисунок 3.11 – Вікно для додавання нової вакансії

Якщо адміністратору необхідно редарувати існуючу вакансію, йому потрібно обрати її із таблиці гололовного вікна для роботи адміністратора та натиснути кнопку ✎ - редагувати обрану вакансію. При натиску на дану кнопку з’являється вікно зображене на рис. 3.12.

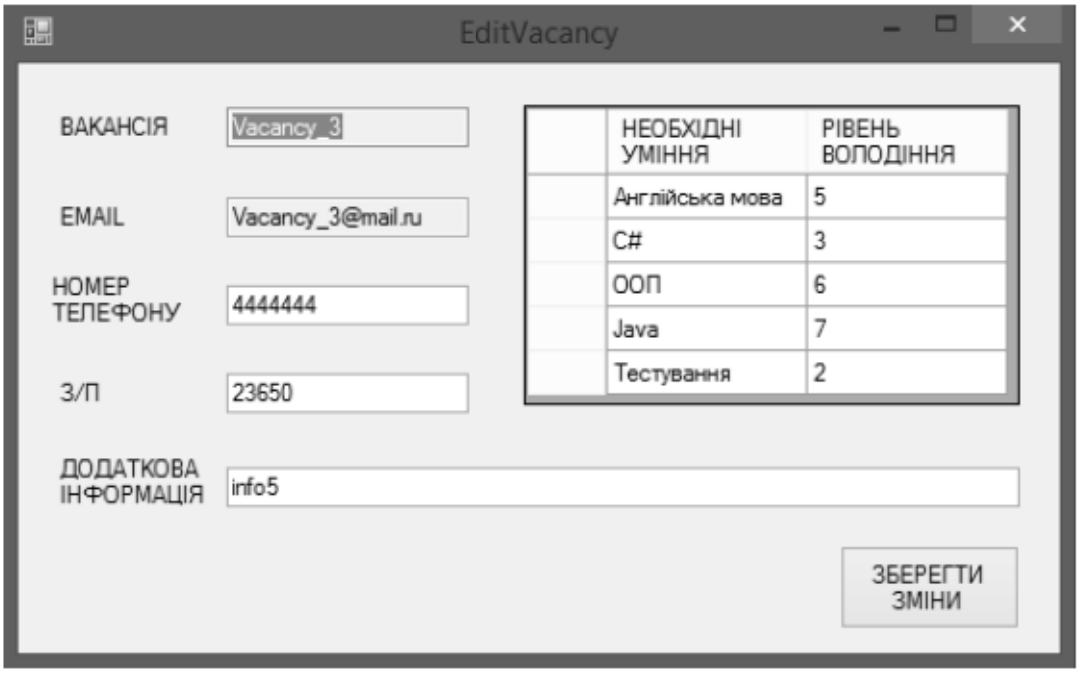


Рисунок 3.12 – Вікно для редагування існуючої вакансії адміністратором

При редагуванні існуючої вакансії забороняється редагувати ігформацію у полях «Вакансія» та «Email», так як ці поля є головними атрибутами сутності. Програмно заблоковано дані поля для забезпечення достовірності інформації.

Якщо вакансію потрібно видалити, то адміністратор повинен обрати її із таблиці гололовного вікна для роботи адміністратора та натиснути кнопку ✕ - видалити обрану вакансію.

3.8 Керівництво користувача

Оскільки, програма потребує роботу з БД, то на комп'ютері користувача повинна бути встановленна СУБД, а саме Oracle. Це необхідний для того, щоб система мала доступ до даних, що необхідні для її роботи.

Далі, потрібно запустити скрпит “Test_baza.sql” вже відповідно у СУБД. Для цього запускаємо СУБД, створюємо схему даних, підключаємося та запускаємо скрипт у командному вікні.

І лише після цього запускаємо вже безпосередньо програму. Для коректно використання програми необхідно папку “ІС пошуку вигідних вакансій” скопіювати в будь-яке місце на жорсткому диску та запустити файл “ SuitableVacancyForUsers.exe”.

З’являється вікно із списком усіх вакансій, які можна фільтрувати обравши одну із запропонованих категоррій. Для того, щоб підібрати вакансію необхідно зарієструватись або авторизуватись, якщо користувач раніше був зарієстрований. Після чого користувач повинен обрати категорію, яка відповідає сфері у якій користувач бажає знайти роботу. Після чого він оцінює запропоновані уміння по шкалі від 1 до 10.

При невірному введенні даних користувач отримає відповідне повідомлення про помилку.

При правильному введенні даних користувачері надається список вакансій, що відповідає його умінням, у вигляді табдлиці.

3.9 Висновки до розділу

Для реалізації автоматизованої системи відслідковування помилок необхідна робота з базою даних. Для правильного створення таблиць у БД, було створенно ERD-модель, яка чітко ілюструє зв'язок між таблиця, залежність між ними.

Також в даному розділі наведений список бізнес правил системи, тобто яка логіка системи, що вона не допукає, при яких умовах не можна вносити дані та інші нюанси.

Система має дві групи користувачів: звичайний користувач та адміністратор. Відповідно, кожній з цих груп були надані певні права та привілегії.

Графічний інтерфейс є невід'ємною частиною будь-якої програми. В якості ознайомлення користувача з інтерфейсом системи, були надані екрані знімки програми для кращого розуміння системи.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

Даний розділ містить результати роботи над програмною реалізацією інформаційної системи, що розроблялась. Тут описано розроблений алгоритм, структуру розроблюваного програмного забезпечення та проведено тестування системи на контрольних прикладах.

4.1 Опис розробленого алгоритму

Основний алгоритм включає в себе етапи:

1) Введення вхідних даних;
2) Перевірка введеної інформації;
3) Виправлення помилок за їх наявності;
4) Збереження даних до БД;
5) Надання вакансій, що відповідають уміннням користувача.

На рисунку 4.1 ілюстрований алгоритм розробленої програми.

Рисунок 4.1 - алгоритм розробленої програми

4.2 Структура програмного забезпечення

На рисунку 4.2 зображена структура програмного забезпечення.

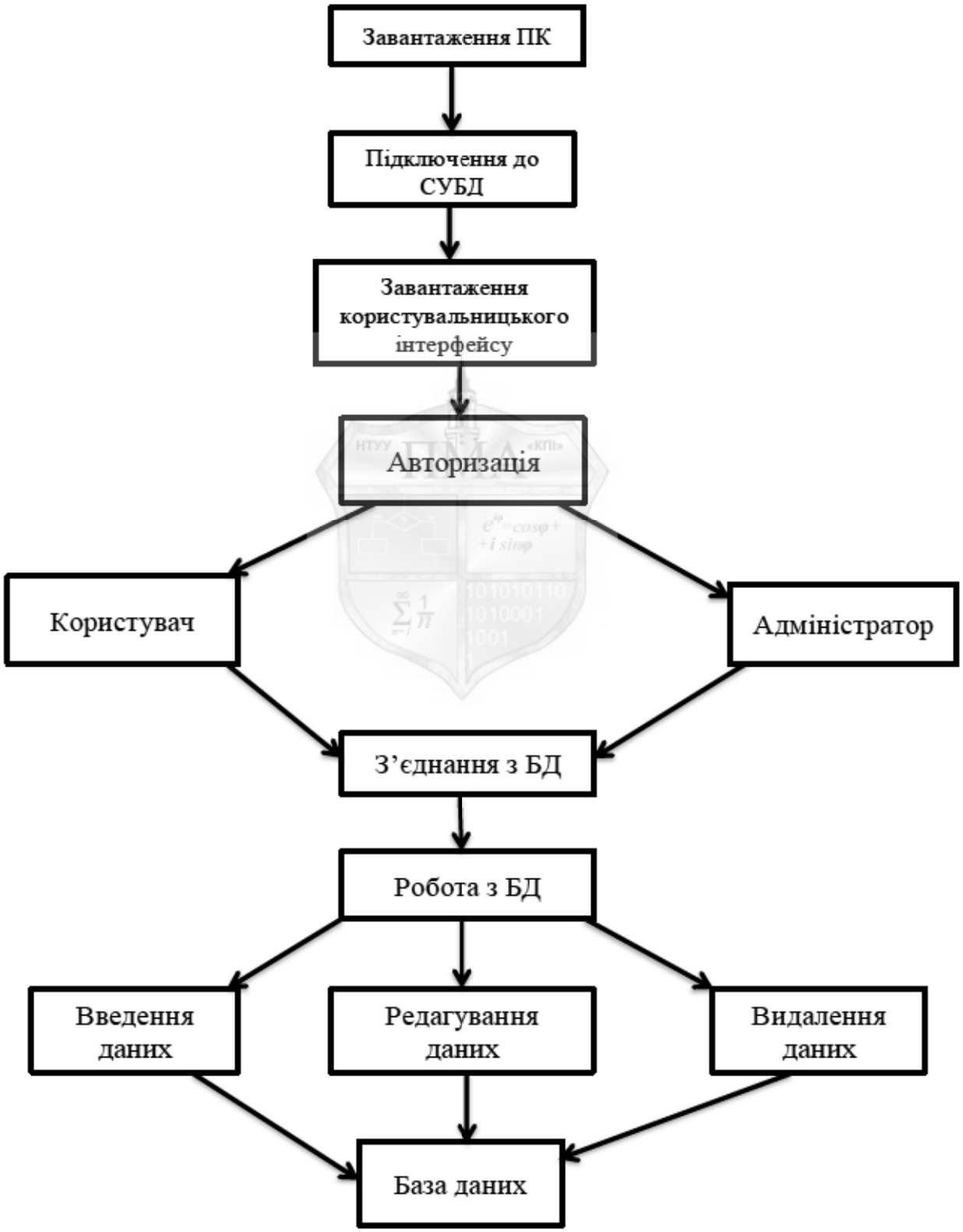


Рисунок 4.2 – Структура програмного забезпечення

4.3 Опис програми

Програмне забезпечення реалізовано на мові програмування C#та на мові структурних запитів Oracle.

Користуватися системою можуть лише авторизовані користувачі, тому для роботи у системі необхідно авторизуватися - ввести логін і пароль.

Програма складається з 8ми форм. З них три для роботи користувача, три – для адміністратора і 2 спільні.

Форми для роботи користувача:

- Форма реєстрації у системі;
- Головне вікно для введення даних;
- Вікно виводу результатів.

Форми для роботи адміністратора:

- Вікно із існуючими вакансіями;
- Вікно для редагування вакансії;
- Вікно для додавання вакансії.

Спільні форми для адміністратора та користувача:

- Вікно для перегляду існуючих вільних вакансій;
- Форма авторизації.

У програмі реалізовані транзакції для додавання, редагування, видалення даних адміністратором системи, що дозволяє уникнути збоїв у системі при одночасній роботі адміністратора та користувачів.

4.4 Опис контрольних прикладів

В якості контрольного прикладу було створені таблиці з даними. Тобто були вакансії з вимогами (рис.4.3) . Для прикладу було надано настувні вакансії з вимогами:

- Vacancy_1: Англійська мова – 8, ООП – 8, Java – 8, C# - 8;
- Vacancy_3: Англійська мова – 5, ООП – 6, Java – 7, C# - 3, Тестування - 2 ;
- Vacancy_6: ООП – 5, Тестування – 5, Java – 5, C# - 5.

На рисунку 4.3 бачимо таблицю вакансій та вимог до них, де зберігаються назва вакансії, email, назва вимоги та необхідни рівень володіння.

	VACANCY_NAME	VACANCY_EMAIL	SKILL_NAME	VACANC...
1	Vacancy_3	Vacancy_3@mail.ru	Англійська мова	5
2	Vacancy_3	Vacancy_3@mail.ru	C#	3
3	Vacancy_3	Vacancy_3@mail.ru	ООП	6
4	Vacancy_3	Vacancy_3@mail.ru	Java	7
5	Vacancy_3	Vacancy_3@mail.ru	Тестування	2
6	Vacancy_6	Vacancy_6@mail.ru	C#	5
7	Vacancy_1	Vacancy_1@mail.ru	Java	8
8	Vacancy_1	Vacancy_1@mail.ru	ООП	8
9	Vacancy_1	Vacancy_1@mail.ru	Англійська мова	8
10	Vacancy_6	Vacancy_6@mail.ru	ООП	5
11	Vacancy_6	Vacancy_6@mail.ru	Java	5
12	Vacancy_6	Vacancy_6@mail.ru	Тестування	5

Рисунок 4.3 – Таблиця вільних вакансій

Оскільки ініормаціна система призначена для пошуку вільних вакансій для користувача відповідно до його умінь, то для перевірки введемо дані користувача ті, які дозволять перевірити роботу програми. Вхідні дані користувача наведено на рис.4.4.

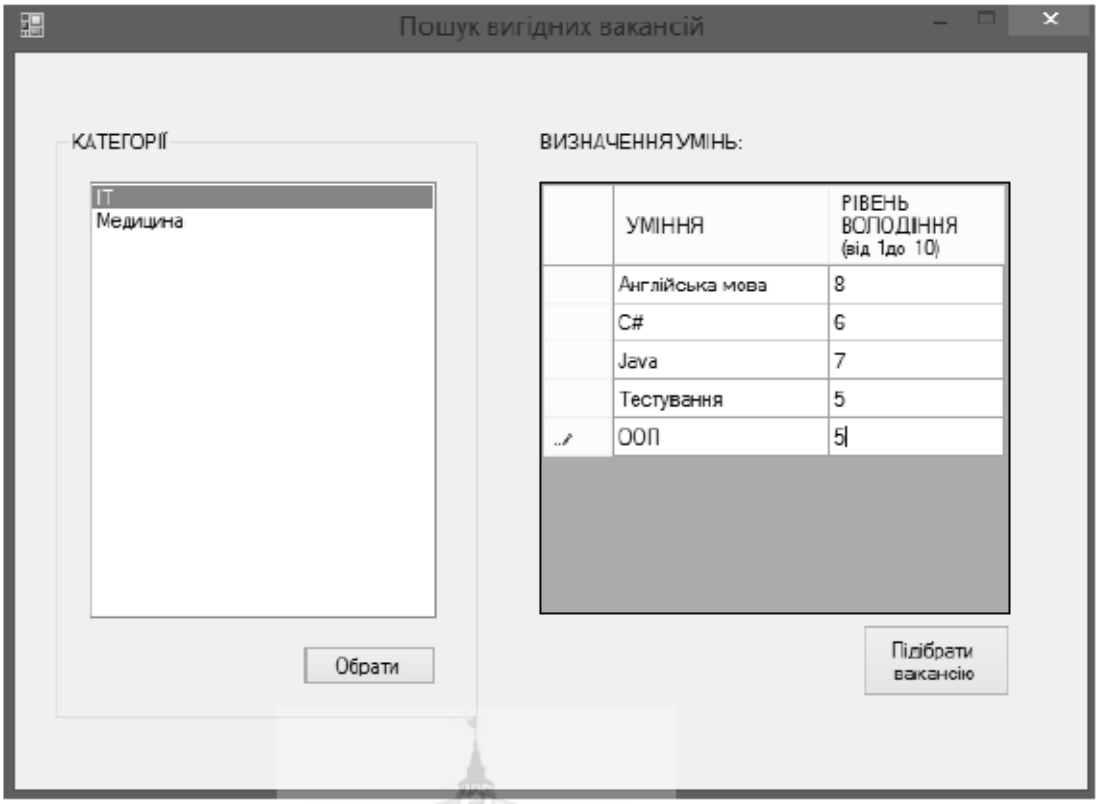


Рисунок 4.4 – Вхідні дані користувача

Отже, таким чином користувач повинен отримати а результаті Vacancy_6 та Vacancy_3 підходять йому, а Vacancy_1 – ні. Результат можемо побачити на екранній формі зображеній на рис.4.5.

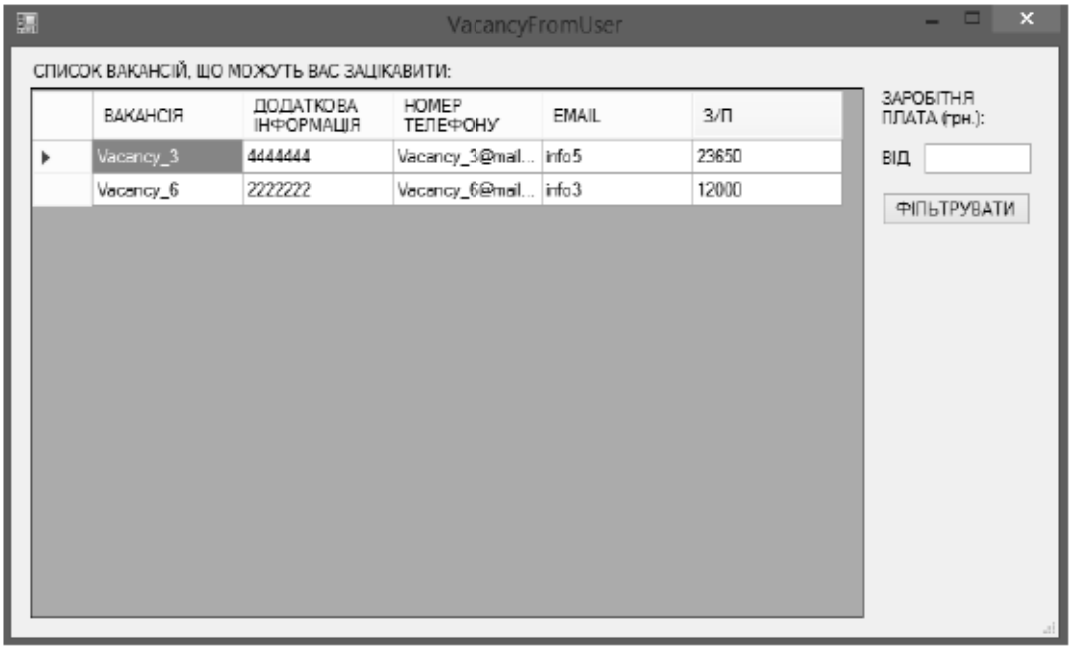


Рисунок 4.5 – Результат роботи програми

Результати роботи програми було збережено до БД у таблицю suitable vacancy for users, що зображено на рис.4.6.

	VACANCY_NAME	VACANCY_EMAIL	USER_LOGIN
1	Vacancy_6	Vacancy_6@mail.ru	1111
2	Vacancy_3	Vacancy_3@mail.ru	1111

Рисунок 4.6 – Збережені результати у таблиці suitable vacancy for users

4.5 **Висновки** до розділу

Інформаційна система пошуку вакансій є корисною для швидкого пошуку бажаної вакансії. Така система призначена для користувачів різного віку та різного рівня володіння ПК. Тому було описано ряд випадків коли система обмежує користувача та повідомляє про помилку. Такими є, наприклад, заборона вводу різних симфолів, окрім латинських літер, у поле «Ім'я».

В якості контрольного прикладу були створені вакансії з вимогами. На рисунках зображено хід роботи пошуку вакансій для користувача, а саме занесення даних до БД і як вони відображуються у системі.

ВИСНОВКИ

Метою даної роботи було створення математичного та програмного забезпечення для пошуку вільних вакансій, що підходять користувачеві на основі аналізу указаних ним умінь. Актуальність даної роботи достатньо висока, адже кожен випускник школи, ліцею, університету, а також люди що хочуть змінити роботу матимуть бажання скористатись розробленою інформаційною системою. Також і для роботодавців дана система значно полегшить життя і зменшись час, який раніше було витрачено на подання оголошень на різні сайти та друковані видання.

Під час виконання роботи було проаналізовано різні підходи у реалізації автоматизованої системи. Основною задачею системи є: можливість збереження та відтворення даних, що надає користувач та адміністратор. За допомогою математичного апарату, було реалізовано підбір вакансій, що підходять користувачеві відносно його умінь.

Програма має достатньо потужний функціонал та користувальницький інтерфейс. Всі елементи у програмі є достатньо зрозумілими і очевидними для користувача. Також у програмі реалізовано ряд перевірок, які допоможуть не досвідченим користувачам користуватися програмою та не пошкодити системи, що забезпечує цілісність інформації.

В цілому, програма орієнтована на компанії, які займаються працевлаштуванням.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Шевченко В.Н., Золотых Н.Ю., Линейное и целочисленное линейное программирование. — Нижний Новгород: Издательство Нижегородского госуниверситета им. Н.И. Лобачевского, 2004. — 154 с.

2. Панченко Т.В. Генетические алгоритмы - эволюционные методы поиска: Астрахань : Издательский дом«Астраханский университет», 2007. — 88 с.

3. L. A. Zaden, The concept of a linguistic variable and itsapplication to approximate reasoning . New York: Academic Press, 1975. - 507 p.

4. Трубін В. А., Енциклопедія кібернетики. Київ: Головна редакція Української радянської енциклопедії, 1973р.том 2 - с. 232-234

5. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. М.: Радио и связь, 1981. — 286 с

6. Руководство MICROSOFT по проектированию архитектуры приложений (2 издание): [Електронний ресурс]. – Режим доступу: https://msdn.microsoft.com/en-us/library/dd673617.aspx

7. Richard Earp, Database Design Using Entity-Relationship Diagrams. Auerbach Publications, 2003. - 242 p.

8. Матриця подій : [Електронний ресурс]. – Режим доступу: http://amis.fpm.kpi.ua/dbis/9-disciplines/dbis/2-elm

9. Черемных С.В, Моделирование и анализ систем. IDEF-технологии. М.: Финансы и статистика, 2006. — 188 с.

10. Диаграммы потоков данных: [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/2195/55/lecture/1632?page=2