

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра прикладної математики

Пояснювальна записка
до курсової роботи
з дисципліни “Бази даних та інформаційні системи”
на тему:
Бот для підбору вакансій

Виконав:

Студент групи КМ-62

Морозюк Є. О.

Перевірили:

Ковальчук-Химюк Л. О

Терещенко І. О

Київ – 2019

АНОТАЦІЯ

В пояснювальній записці до курсової роботи описано основні етапи розробки інформаційної системи. Метою даної роботи є розробка онлайн-сервісу, який би надавав користувачу можливість знайти роботу за його навичками. Дана робота є лише прототипом, в якому реалізовано CRUD-операції для всіх сутностей, а також механізм для збору і відображення статистики

РЕФЕРАТ

В ході виконання курсового проекту було створено веб-застосунок на мікрофреймворку Flask. База даних, на якій виконано завдання – PostgreSQL. Схема бази даних описана у файлі model.py. За допомогою orm SQLAlchemy була створена міграція, за допомогою якої, власне, і була створена база. Веб-застосунок дозволяє переглядати дані, наявні в інформаційній системі, змінювати та видаляти дані, де це допустимо та додавати дані у всі сутності. Для створення форм було використано WTForms, на яких також, за допомогою налаштування випадаючих списків та обмеженого вибору підтримано цілісність даних. З Python до HTML-сторінок передається інформація та відображається з допомогою шаблонізатора Jinja, а веб-сторінки зверстані з використанням фреймворку Bootstrap-4. Віддалений сервер для розгортання виконаного завдання міститься на Heroku.

ЗМІСТ

ВСТУП	5
ОСНОВНА ЧАСТИНА	6
1 ПОСТАНОВКА ЗАДАЧІ	6
1.1 Категорія користувачів	6
1.2 Класи даних	7
1.4 Матриця елементарних події (сценарії)	10
1.5 Бізнес-правила	12
2 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ	13
3 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ	15
4 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ	16
ВИСНОВКИ	18
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	19

ВСТУП

В Україні щонайменше 2 масштабні сервіси для пошуку роботи. Ці сервіси мають відмінний дизайн, чудовий UX. Проте ці сервіси лише показують список вакансій по ключовому слову. Метою курсової роботи є створити підґрунтя для мікросервісу, що зможе доповнити звичайні сервіси для відображення інформації в дещо більше: в залежності від навичок користувача підбирати вакансію. Крім цього, сервіс оброблює отриману інформацію та відображає деяку статистику: Список навичок і скільки користувачів мають цей навичок.

ОСНОВНА ЧАСТИНА

1 ПОСТАНОВКА ЗАДАЧІ

Основна відмінність простої автоматизації робочого процесу від пропонованого мікросервісу в тому, що будемо здобувати корисну інформацію та робити дослідження. Наприклад, залежність зарплати від досвіду.

Основна функція мікросервісу: збір та обробка інформації .

Вхідні дані мікросервісу: данні о користувачеві, професій, вакансій та навичок.

Приклад структури вхідних даних мікросервісу: форма з полями, для кожної сутності поля унікальні(сутності зазначенні вище).

Результат роботи мікросервісу: відображення інформації.

Задача кластеризації: групування професій за рівнем зарплат.

Мета задачі кластеризації для поставленої задачі: для того, щоб визначити найбільш перспективніші професії з точки зору рівня доходу.

Задача кореляційного аналізу: пов'язати залежність між досвідом роботи та зарплатнею.

Визначити мету кореляційного аналізу та для яких атрибутів: для того, щоб прогнозувати ріст зарплати для користувача.

1.1 Категорія користувачів

У сервісі, що реалізовано в курсовому проєкті є тільки одна категорія користувачів (в даному сервісі це категорія - адміністратор). Звичайно, сервіс можна розширити з додаванням авторизації та

різних ролей. Тоді до різних сутностей та дій будуть різні права доступу.

Наприклад, адміністратор зможе робити будь-які CRUD операції зі всіма доступними сутностями, можна додати звичайного користувача, який зможе лише зареєструватися, переглядати інформацію та статистику.

1.2 Класи даних

Опишемо наявні у системі класи даних у таблицях 1.1 – 1.5.

Таблиця 1.1 – Клас даних «Користувач»

Сутність	Користувач	
Опис сутності	Людина, яка є основним користувачем системи	
Атрибути сутності	Опис атрибуту	Пов'язана сутність із атрибутом
ID	Універсальний ідентифікатор	Навички користувача
Ім'я	Ім'я користувача	—
Прізвище	Прізвище користувача	—
Дата народження	Дата народження	—
Місто	Місце проживання	—

Таблиця 1.2 – Клас даних «Навички»

Сутність	Навички	
Опис сутності	Навички, які може мати користувач або професія	
Атрибути сутності	Опис атрибуту	Пов'язана сутність із атрибутом
ID	Універсальний ідентифікатор	Навички користувача та професії
Назва	Як називається навичок	-

Таблиця 1.3 – Клас даних «Навички користувача»

Сутність	Навички користувача	
Опис сутності	Співаки та гурти	
Атрибути сутності	Опис атрибуту	Пов'язана сутність із атрибутом
ID	Універсальний ідентифікатор	-
ID навичка	Посилання на навичок	Навичок
ID користувача	Посилання на користувача	Користувач

Таблиця 1.4 – Клас даних «Навички професії»

Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	-
ID навичка	Посилання на навичок	Навичок
ID професії	Посилання на професію	Професія

Таблиця 1.5 – Клас даних «Вакансія»

Сутність	Вакансія	
Опис сутності	Альбом, що об’єднує пісні	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Мелодія
Назва	Як називається вакансія	—
Обов’язки	Обов’язки	-
Опис	Опис вакансії	-
Зарплата	Зарплата вакансії	-
Дата	Дата створення вакансії	-

ID професії	Посилання на професію	Професії
-------------	-----------------------	----------

1.4 Матриця елементарних події (сценарії)

Таблиця 1.6 – Процес створення нового запису в сутності

Назва процесу	Додавання запису до сутності
Сутності	Відповідно кожна із сутностей, зазначена у таблицях 1.1–1.5 цього розділу
Вхідні атрибути сутності	В процесі додавання вхідні атрибути пусті, бо запис ще не вставлено
Опис функціоналу	Дані з форми на веб-сторінці записуються у поля класу, що відповідають сутності та відправляються на сервер
Змінені атрибути сутності	Рядок вставлених даних

Таблиця 1.7 – Процес редагування запису в сутності

Назва процесу	Редагування запису
Сутності	Відповідно кожна із сутностей, зазначена у таблицях 1.1–1.5 цього розділу
Вхідні атрибути сутності	В процесі додавання вхідні атрибути вже заповнені

Опис функціоналу	Дані з форми на веб-сторінці записуються у поля класу, що відповідають сутності та відправляються на сервер
------------------	---

Продовження таблиці 1.8 – Процес редагування запису в сутності

Назва процесу	Додавання запису до сутності
Змінені атрибути сутності	Рядок вставлених даних

Таблиця 1.9 – Процес видалення запису із сутності

Назва процесу	Видалення запису до сутності
Сутності	Відповідно кожна із сутностей, зазначена у таблицях 1.1–1.5 цього розділу
Вхідні атрибути сутності	Ідентифікатор запису
Опис функціоналу	Дані з форми на веб-сторінці записуються у поля класу, що відповідають сутності та відправляються на сервер
Змінені атрибути сутності	Видалення рядку із бази даних
Назва процесу	Видалення запису із сутності

1.5 Бізнес-правила

1. Користувач повинен мати прізвище, ім'я, дату народження та місто проживання;
2. Користувач не може додати навичок, якого немає в базі даних;
3. Користувач не може додати до професії навичок, якого нема в базі;
4. Користувач не може створити вакансію з професією, якої нема в базі;
5. Користувач має можливість переключатися між вкладками за допомогою навігаційного меню.
6. Користувач може переглядати статистику;
7. Користувач має можливість редагувати всі сутності;

2 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

Відповідно до work-flow із постановки задачі, маємо процеси пошуку вакансій. Задача кластеризації професій по рівню зарплат привела до процесів дослідження вакансій. У короткому представленні ієрархія процесів наведена на рисунку 2.1.

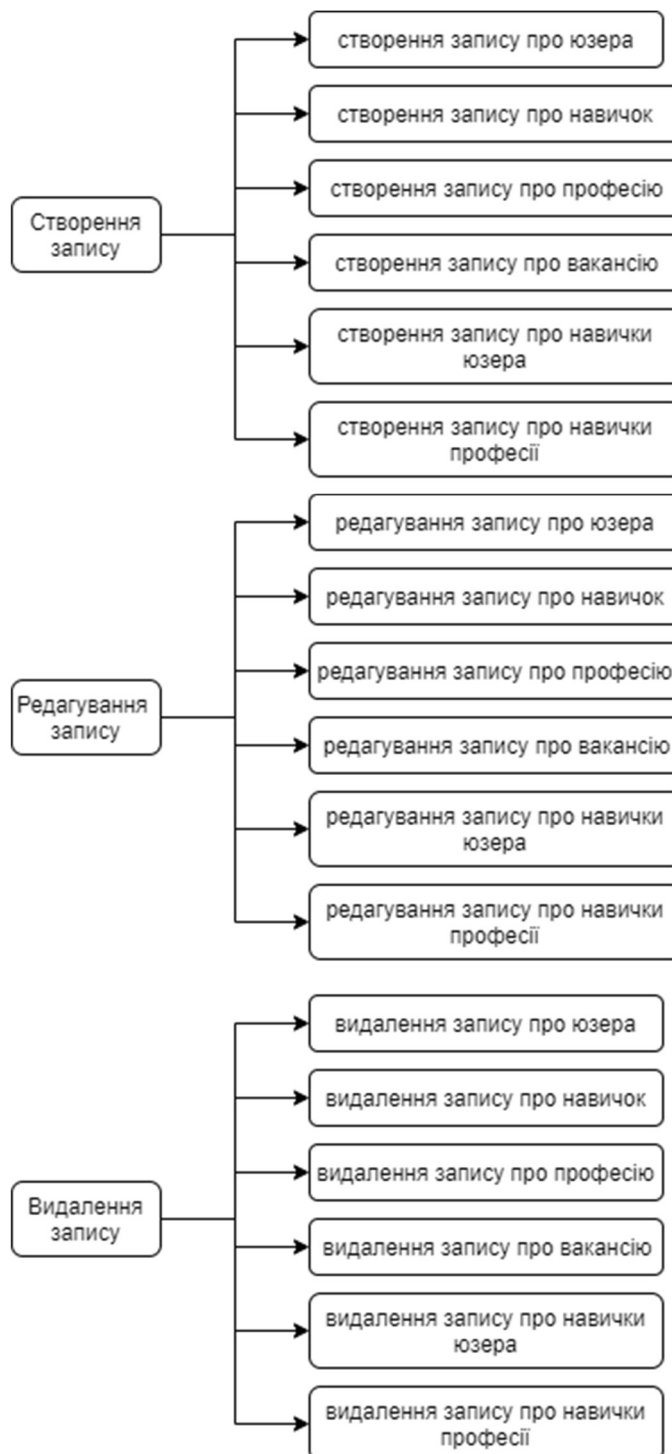


Рисунок 2.1 – Ієрархія процесів

Змодельований Use-Case для процесів, що можна виконувати наразі у веб-застосунку, зображено на рисунку 2.2.

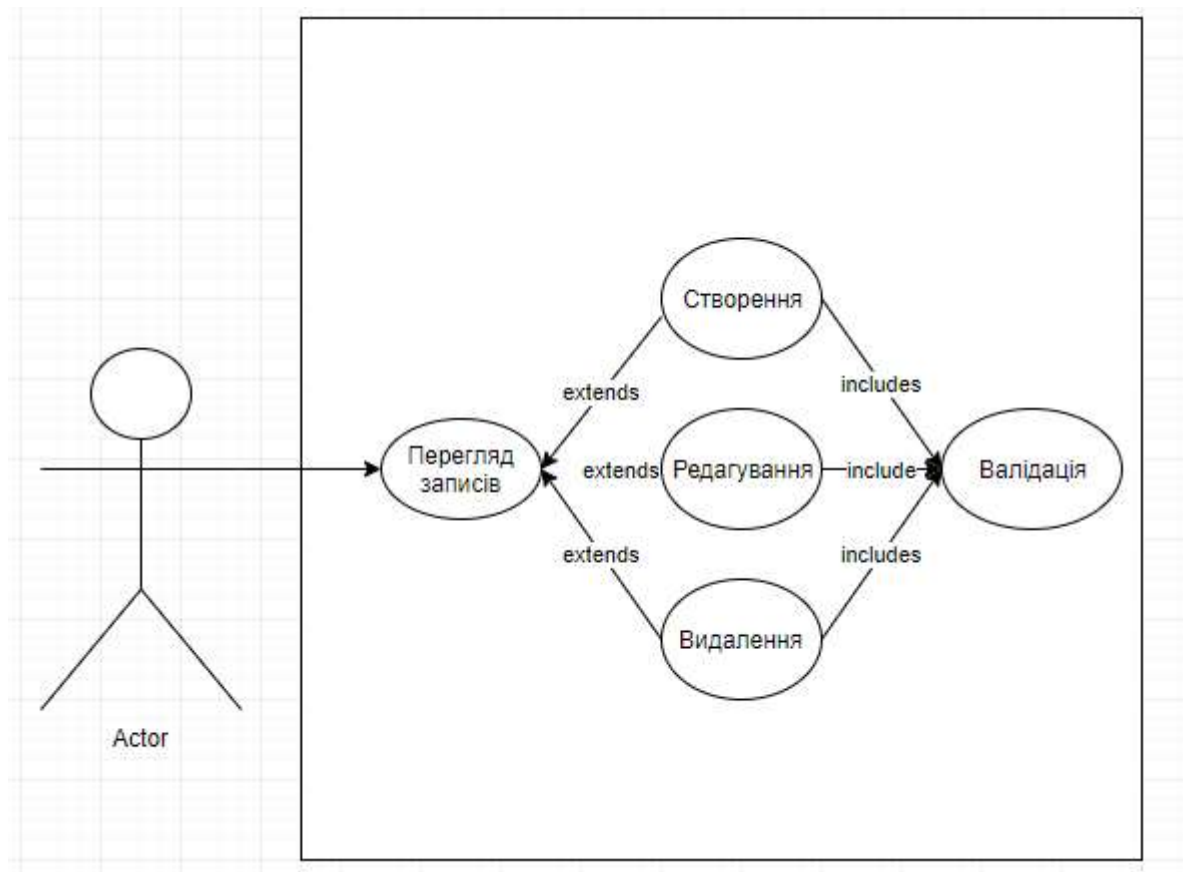


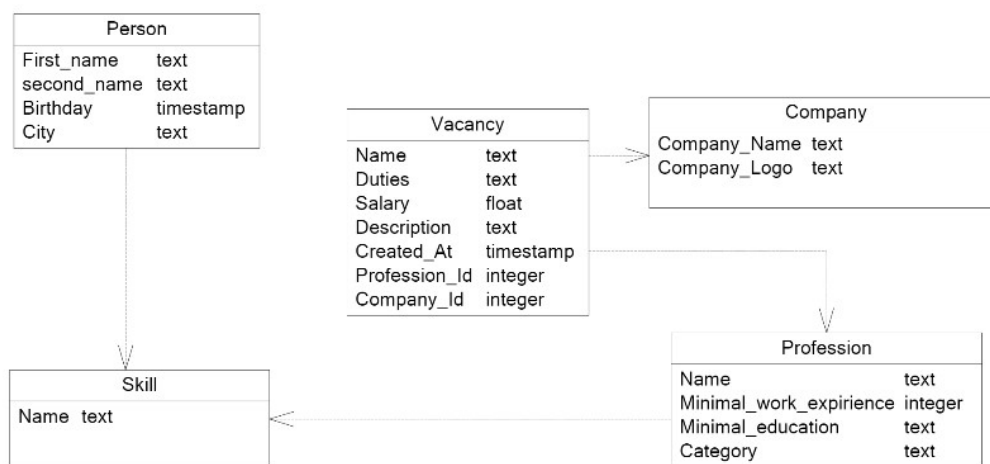
Рисунок 2.2 – Use-case diagram для веб-застосунку

3 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Під час проектування було визначено 4 основні сутності біля яких буде зконцентрована робота сервісу:

1. Користувач – сутність, яка зберігає інформацію про користувача сервісу.
2. Навичок - сутність, яка є зберігає в собі ідентифікатор та назву. Користувач має вміння, професія – потребує навичок.
3. Професія - сутність, яка зберігає інформацію про професію: назва, навички, освіта, категорія.
4. Вакансія - сутність, яка зберігає інформацію про вакансію: назву, обов'язки, зарплата, опис, професія.

Концептуальна діаграма зображена на рисунку 3.1.



4 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Далі концептуальна модель перетворюється на фізичну. Для цього в інтерфейсі Power Designer обираємо PostgreSQL (в цій версії програми наявна версія 8 БД). Формується модель, що зображена на рисунку 5.2. Разом із моделлю Power Designer надає код, в який залишається додати обмеження унікальності, що виникли на етапі оптимізації логічної моделі.

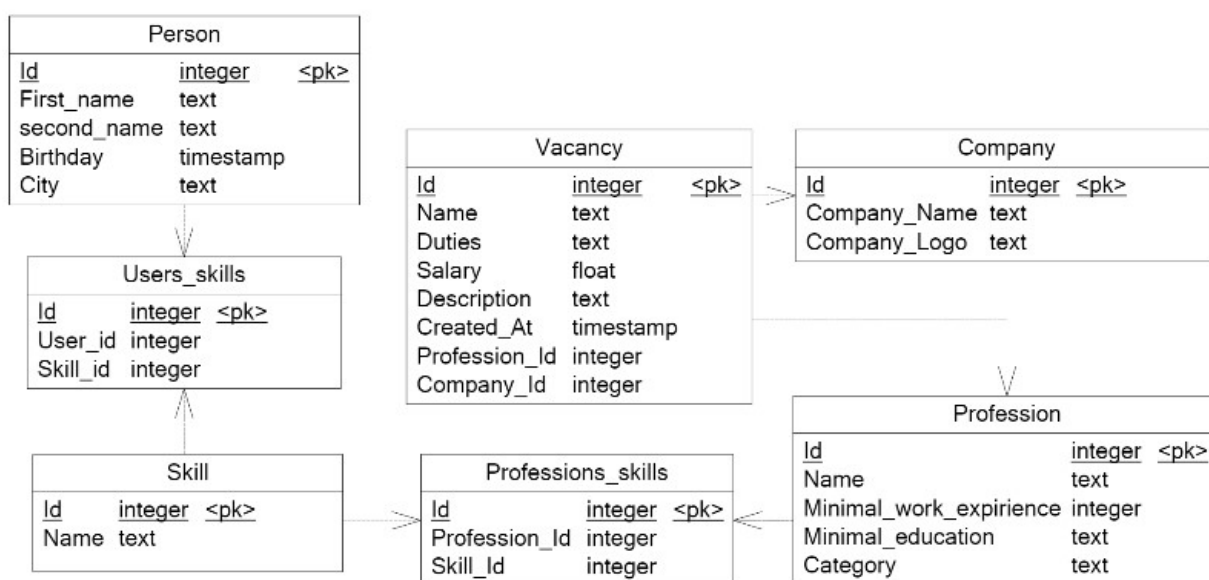


Рисунок 4.1 – Фізична модель даних

Таким чином, отримали модель даних, де кожна сутність містить штучні ключі. Штучні ключі в отриманому сервісі редагуватись користувачем не будуть: відбуватиметься генерація ключа випадковим образом у влаштованих процедурах. Тепер завдяки тому, що маємо штучні ключі, WTF-форми на сайті працюватимуть коректно, будуть наявні HiddenField().

Кожну таблицю моделі можна буде пов'язати з будь-якою іншою, виконавши перехресні записи із JOIN. Цілісність за посиланнями підтримуватиметься перехресними запитами з кожної форми, якщо у відповідній формі сутності є батьківська. Якщо батьківських сутностей немає – не обмежуємо користувача у вводі інформації, окрім валідації полів.

ВИСНОВКИ

В ході даної роботи був розроблений сервіс який надавав користувачеві можливість виконання CRUD-операцій над сутностями: користувач, навик, професія, вакансія і т.д.. В якості платформи був використаний сервіс Heroku, а стек технологій був наступний:

- dbms - PostgreSQL
- server - Flask
- orm mapping - SQLAlchemy
- interface - JavaScript, Bootstrap 4

Варто зазначити, що мова Python та мікрофреймворк Flask виявилися дуже зручною для розробки подібних невеликих веб-застосунків. Це зумовлено із структурою мови Python. SQLAlchemy значно спростила розробку інтерфейсу взаємодії Python, Flask та БД PostgreSQL. Завдяки orm можна достатньо легко зробити міграцію БД або початкову ініціалізацію. Сервіс Heroku ідеально підходить на випадки, якщо треба протестувати сервіс в Інтернеті або ж для невеликих веб-застосунків, як в даному випадку. Heroku має наявності зручний інтерфейс для взаємодії з ним а також розширення для розробників, які використовують в своїх системах БД.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. PostgreSQL Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.elephantsql.com/docs/index.html>.
2. SQLAlchemy 1.3 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sqlalchemy.org/en/13/>.
3. WTForms Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://wtforms.readthedocs.io/en/stable/>
4. Deploy app to the heroku [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@dushan14/create-a-web-application-with-python-flask-postgresql-and-deploy-on-heroku-243d548335cc>