

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ

КУРСОВА РОБОТА
з навчальної дисципліни “Бази даних та інформаційні системи-2”
на тему
СЕРВІС ІНТЕГРАЦІЇ МЕСЕНДЖЕРІВ TELEGRAM/VIBER/WHATSAPP В ЄДИНУ
ПЛАТФОРМУ

Виконала студентка групи КМ-62
ШУМЕЛЬ С. О.

Керівник
ТЕРЕЩЕНКО І.О.

Оцінка:
Кількість балів:

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Створити інформаційну систему, що дозволяє об'єднувати повідомлення з різних месенджерів користувача на єдиній платформі. Для цього виконати передпроектне дослідження (проаналізувати бізнес-процеси, бізнес-правила та елементарні події), визначити сутності, що міститимуться у базі даних, проаналізувати операції, що можуть виконуватись над кожною із сутностей, забезпечити цілісність системи. Додатково виконати кореляційний та кластерний аналізи. Розгорнути завдання на віддаленому сервері.

АНОТАЦІЯ

В пояснювальній записці до курсової роботи описано основні етапи розробки інформаційної системи, інформацію про виконання передпроектного дослідження, детальну постановку задачі, виконано моделювання бізнес-процесів, визначено модель даних, описано сутності інформаційної системи, надано графічне представлення моделі.

РЕФЕРАТ

В ході виконання курсового проекту було створено інформаційну систему на базі веб-фреймворку Flask. База даних, на якій виконано завдання – PostgreSQL 10. Власне модель створюється на базі модуля Python SQLAlchemy, а зі сторони PostgreSQL налаштовано обмеження на унікальність, які виникли після оптимізації бази даних та введення штучних ключів.

Веб-застосунок дозволяє переглядати дані, наявні в інформаційній системі, змінювати, видаляти та додавати дані у всі сутності. Додавання даних валідується за допомогою 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 Jinja2, а веб-сторінки зверстані з використанням фреймворку Bootstrap.

ЗМІСТ

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ	2
АНОТАЦІЯ	3
РЕФЕРАТ	4
ЗМІСТ	5
ВСТУП	6
1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ	7
2 ПОСТАНОВКА ЗАДАЧІ	8
2.1 Категорії користувачів	8
2.2 Класи даних	9
2.3 Бізнес-правила	11
2.4 Матриця елементарних подій (сценарії)	12
3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ	16
4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ	17
5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ	19
ВИСНОВКИ	21
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	22
Додаток А. Use-Case діаграма	23

ВСТУП

Станом на сьогодні існує безліч месенджерів, які об'єднують різні групи людей для онлайн-спілкування. Наприклад, Telegram, Whatsapp, Viber, Slack та інші. Переважна більшість людей зареєстровані в одному з перелічених додатків: це дозволяє за допомогою інтернету швидко обмінюватися потрібною інформацією.

Кожен месенджер націлений на певну аудиторію (групу людей). З чого випливає, що різними месенджерами користуються різні люди. Це створює ряд проблем: одна людина потрапляє в декілька соціальних груп, різні групи мають комунікувати між собою. Це призводить до того, що людина має використовувати декілька месенджерів одночасно аби задовольнити потребу спілкування з людьми. Наприклад, переважна більшість молоді використовує Telegram, у той час коли старше покоління — Viber. Діти потрапляють в першу групу, батьки — в другу. Аби підтримувати онлайн-спілкування між батьками та дітьми, хтось має встановлювати додаток іншої соціальної групи. У той саме час, одна з вище згаданих груп може працювати в ІТ-сфері, де прийнято використовувати додаток Slack. Тобто людина має встановити 3 додатки. Це створює багато незручностей. По-перше, використання й так обмеженої пам'яті телефону. По-друге, моніторинг повідомлень в усіх додатках.

Було вирішено створити платформу для мікросервісу, що зможе об'єднати усі месенджери разом. Згодом на базі виконаних досліджень можна буде проаналізувати використання різних месенджерів різними групами, оцінити залежність між тематикою повідомлення та месенджером.

Об'єктом дослідження у курсовій роботі є функціонал наявних інформаційних систем, існуючі рішення, які піддаються інтеграції.

Метод дослідження — пошук та систематизація наявних рішень для створення інформаційних систем та мікросервісів.

1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ

Відповідно до постановки задачі, маємо процеси реєстрації та авторизації користувача, перегляд повідомлень та створення повідомлень у разі використання сервісу у ролі клієнта, а також операції CRUD до всіх сутностей у разі авторизації у ролі адміністратора. У короткому представленні ієрархія процесів наведена на рисунку 1.1.

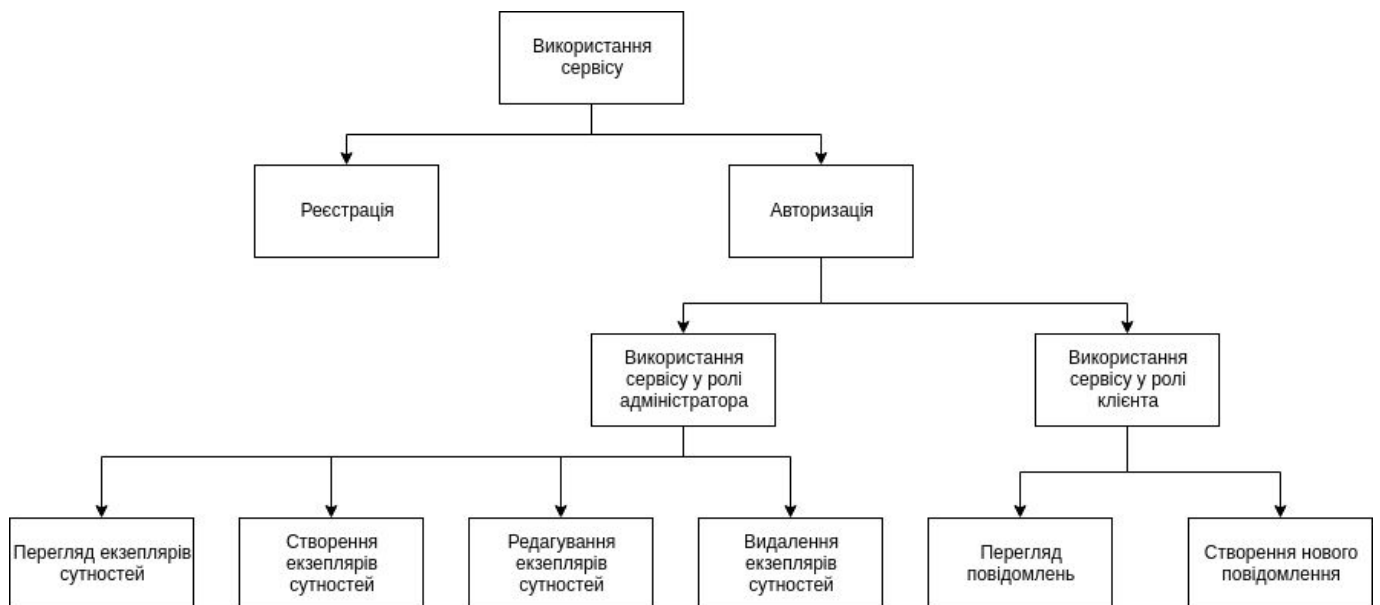


Рисунок 1.1 – Ієрархія процесів

2 ПОСТАНОВКА ЗАДАЧІ

Створити мікросервіс для інтеграції месенджерів в єдину платформу.

Основна функція мікросервісу: відображувати повідомлення користувача з різних месенджерів.

Вхідні дані мікросервісу: данні повідомлення (ім'я відправника, заголовок повідомлення, текст повідомлення, месенджер, категорія повідомлення).

Приклад структури вхідних даних мікросервісу: текстове повідомлення.
Словник: користувач – повідомлення.

Результат роботи мікросервісу: таблиця з повідомленнями користувача.

Далі розглянемо основні категорії користувачів сервісу, класи даних та елементарні події, що відбуваються у ході користування системою.

2.1 Категорії користувачів

У сервісі, що реалізовано в курсовому проекті є дві категорії користувачів.

Клієнт може створювати повідомлення та переглядати таблицю зі своїми повідомленнями.

Адміністратор може робити CRUD операції на сервісі, не дістаючись при цьому до інтерфейсу PostgreSQL, робота з базою автоматизована.

2.2 Класи даних

Опишемо наявні у системі класи даних у таблицях 2.1 –2.4.

Таблиця 2.1 – Клас даних «Користувач»

Сутність	Користувач	
Опис сутності	Основний користувач системи	
Атрибути сутності (1)	Опис атрибуту (2)	Пов’язана сутність із атрибутом (3)
ID	Універсальний ідентифікатор	Повідомлення, месенджер
Ім’я (нікнейм)	Ім’я користувача в системі	—
Пароль	Пароль користувача для авторизації в системі	—
Локація	Країна проживання користувача	—

Таблиця 2.2 – Клас даних «Месенджер»

Сутність	Месенджер	
Опис сутності	Дані про месенджер	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Повідомлення, користувач
Назва	Назва месенджера	—

Таблиця 2.3 – Клас даних «Категорія»

Сутність	Категорія	
Опис сутності	Категорія, якій належить повідомлення	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Повідомлення
Назва	Назва категорії	—
Чисельність	Кількість повідомлень даної категорії	—

Таблиця 2.4 – Клас даних «Повідомлення»

Сутність	Повідомлення	
Опис сутності	Повідомлення користувача	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	—
Заголовок	Заголовок повідомлення	Альбом
Текст	Текст повідомлення	Жанр
Дата створення	Коли повідомлення було відправлено	—
Користувач	Користувач, який відправив повідомлення	Користувач

Продовження таблиці 2.4

Месенджер	Месенджер, з якого було інтегровано повідомлення	Месенджер
Категорія	Категорія, якій належить повідомлення	Категорія

2.3 Бізнес-правила

Визначимо бізнес-правила для інформаційної системи:

- 1) користувач повинен мати унікальне ім'я в системі та пароль, за допомогою яких відбувається авторизація
- 2) користувач, якого немає в системі має змогу зареєструватися
- 3) для реєстрації на сервісі користувач має вести унікальний логін та пароль, а також країну свого проживання
- 4) користувач для надсилання повідомлення обов'язково має обрати месенджер
- 5) користувач може обрати тільки месенджер, який є у базі
- 6) користувач для надсилання повідомлення обов'язково має обрати категорію, якій належить повідомлення
- 7) користувач може обрати тільки категорію, яка є у базі
- 8) для входу в систему у ролі адміністратора використовується раніше визначений логін та пароль
- 9) адміністратор повинен мати можливість редагування всіх таблиць, які відповідають сутностям мікросервісу
- 10) користувач може переглядати тільки свої повідомлення

- 11) повідомлення не обов'язково повинно мати месенджер та категорію
- 12) користувач обов'язково повинен мати країну проживання
- 13) користувач повинен мати можливість виходу з системи (повернення до початкової сторінки входу на сервіс)
- 14) редагує повідомлення, адміністратор може лише змінювати месенджер та категорію

2.4 Матриця елементарних події (сценарії)

Тепер розглянемо процеси, наведені у ієрархії в розділі 1 курсового проекту. Кожен процес може чинити вплив на атрибути сутностей.

Розглянемо кожен з них у таблицях 2.5–2.11

Таблиця 2.5 – Процес реєстрації

Назва процесу	Реєстрація
Сутності	Користувач
Вхідні атрибути сутності	Всі атрибути, крім штучного ключа
Опис функціоналу	На сервер POST-запитом відправляються дані з форми реєстрації. Виконується операція Select з фільтрацією по імені користувача з метою збереження унікальності логіну. У разі вдалої перевірки на унікальність, в таблиці користувачів вставляється новий екземпляр сутності Користувач.
Змінені атрибути сутності	Новий рядок у сутності Користувач

Таблиця 2.6 – Процес авторизації

Назва процесу	Авторизація
Сутності	Користувач
Вхідні атрибути сутності	Логін та пароль
Опис функціоналу	На сервер POST-запитом відправляються дані з форми авторизації. Виконується запит Select з бази даних. Відбувається перевірка наявності в базі данного логіну та відповідності паролю. У разі невдалого входу з'являється помилка
Змінені атрибути сутності	Сутність не змінює атрибутів.

Таблиця 2.7 – Процес перегляду екземплярів сутності

Назва процесу	Перегляд екземплярів сутності
Сутності	Всі сутності
Вхідні атрибути сутності	Всі атрибути
Опис функціоналу	Виконується операція Select, поля отриманого результату передаються на веб-сторінку.
Змінені атрибути сутності	Сутність не змінює атрибутів.

Таблиця 2.8 – Процес створення екземплярів сутностей

Назва процесу	Створення екземплярів сутностей
Сутності	Всі сутності
Вхідні атрибути сутності	Всі атрибути, крім штучних ключів
Опис функціоналу	На сервер POST-запитом відправляються дані з форми для створення сутності. Відбувається перевірка унікальності всіх полів, які того потребують. Після чого вставляється в таблицю сутності новий екземпляр
Змінені атрибути сутності	Новий рядок у сутності

Таблиця 2.9 – Процес редагування екземплярів сутностей

Назва процесу	Редагування екземплярів сутностей
Сутності	Всі сутності
Вхідні атрибути сутності	Рядок із полями запису, отриманого за його ID.
Опис функціоналу	На сервер GET-запитом передається ID запису, його полями заповнюємо форму. Після цього у формі адміністратор змінює значення атрибутів. Дані зчитуються із полів форми та відправляються на сервер POST-запитом.
Змінені атрибути сутності	Процес не змінює атрибутів.

Таблиця 2.10 – Процес видалення екземлярів сутностей

Назва процесу	Видалення екземлярів сутностей
Сутності	Всі сутності
Вхідні атрибути сутності	Унікальний ідентифікатор ID.
Опис функціоналу	Виконується GET-запит. Отримуємо запис сутності за ID, що надійшов після натискання на відповідну кнопку, та видаляємо запис за отриманим ID.
Змінені атрибути сутності	Рядок за заданим ідентифікатором відсутній.

Таблиця 2.11 – Процес перегляду повідомлень клієнтом

Назва процесу	Перегляд повідомлень клієнтом
Сутності	Повідомлення, Месенджер, Категорія, Користувач, Виконавець
Вхідні атрибути сутності	Повідомлення - Заголовок, Текст, Дата створення. Месенджер - ім'я. Категорія - ім'я. Користувач - унікальний ідентифікатор ID.
Опис функціоналу	Виконується операція Select, поля отриманого результату передаються на веб-сторінку. При цьому виконується операція JOIN між сутностями за штучними ключами.
Змінені атрибути сутності	Процес не змінює атрибутів.

3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

Use-Case — це дії, що користувач, або адміністратор, може робити з інтерфейсом сервісу, та те, як сервер може відповідати — наприклад, надати повідомлення про помилку, тому елементи Use-Case-діаграми відповідають вкладкам/кнопкам/повідомленням на створеному веб-сайті.

Повністю змодельований Use-Case для процесів, що можна виконувати наразі у веб-застосунку, зображено у додатку А.

4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Змоделюємо сутності та зв'язки.

- a) Користувач відправляє повідомлення. Користувач може мати багато повідомлень.
- b) Повідомлення має заголовок, текст, дату створення, користувача, що відправив повідомлення, месенджер за допомогою якого було віправлено повідомленні та категорію, якій належить повідомлення. Повідомлення може мати лише одного користувача, один месенджер та одну категорію.
- c) Категорія може належати багатьом повідомленням.
- d) Користувач може бути зареєстрований в багатьох месенджерах.
- e) Месенджер може мати багато користувачів
- f) Месенджер може мати багато повідомлень

Таким чином, користувача ідентифікуємо за нікнеймом. Повідомлення можна визначити за датою створення та користувачем, месенджер та категорію – за назвою.

Концептуальна діаграма зображена на рисунку 4.1.

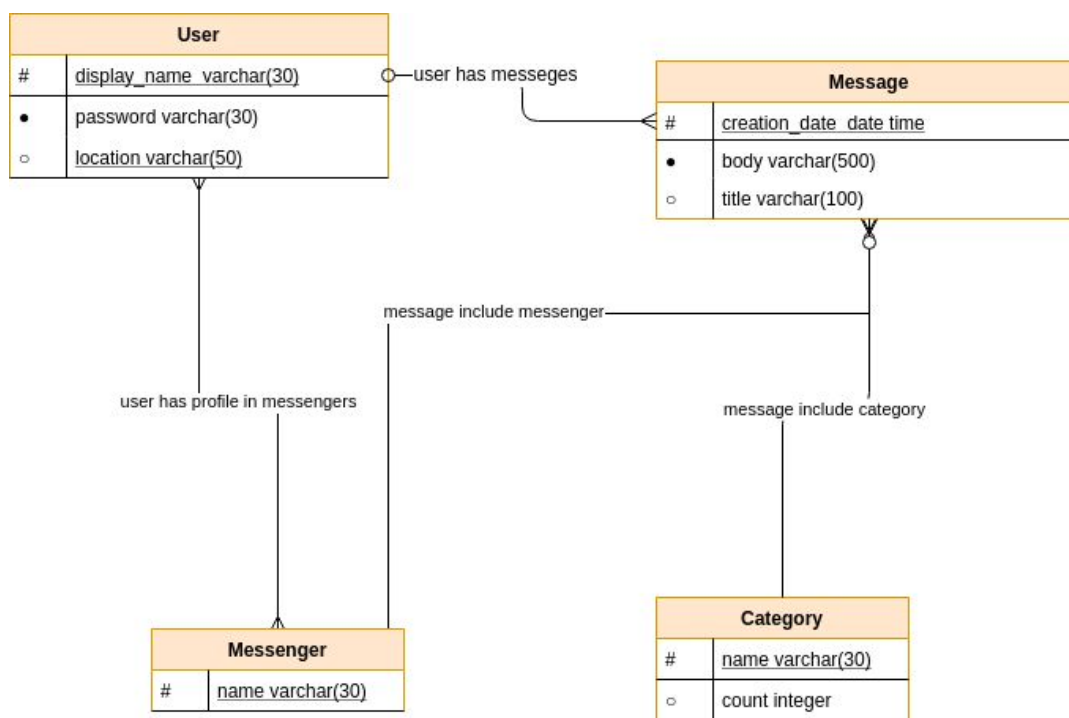


Рисунок 4.1 – Концептуальна діаграма бази даних

5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Наступний крок – перетворити концептуальну модель на логічну та оптимізувати її. При такому перетворенні вводять штучні ключі, а на атрибути, що були ключовими у концептуальній моделі накладається обмеження унікальності. Оптимізована логічна модель зображена на рисунку 5.1.

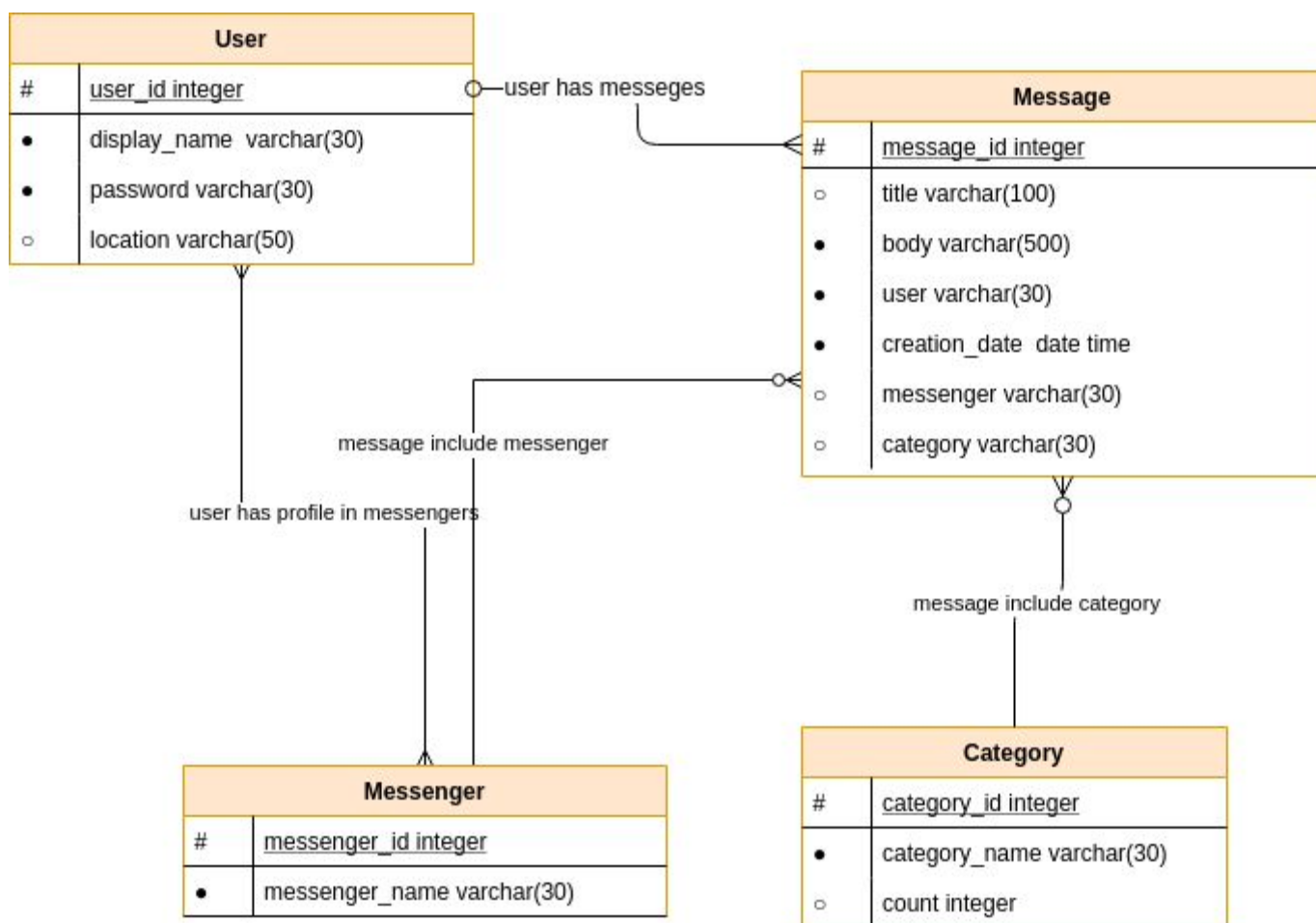


Рисунок 5.1 – Оптимізована логічна модель даних

Далі логічна модель перетворюється на фізичну. Формується модель, що зображена на рисунку 5.2.

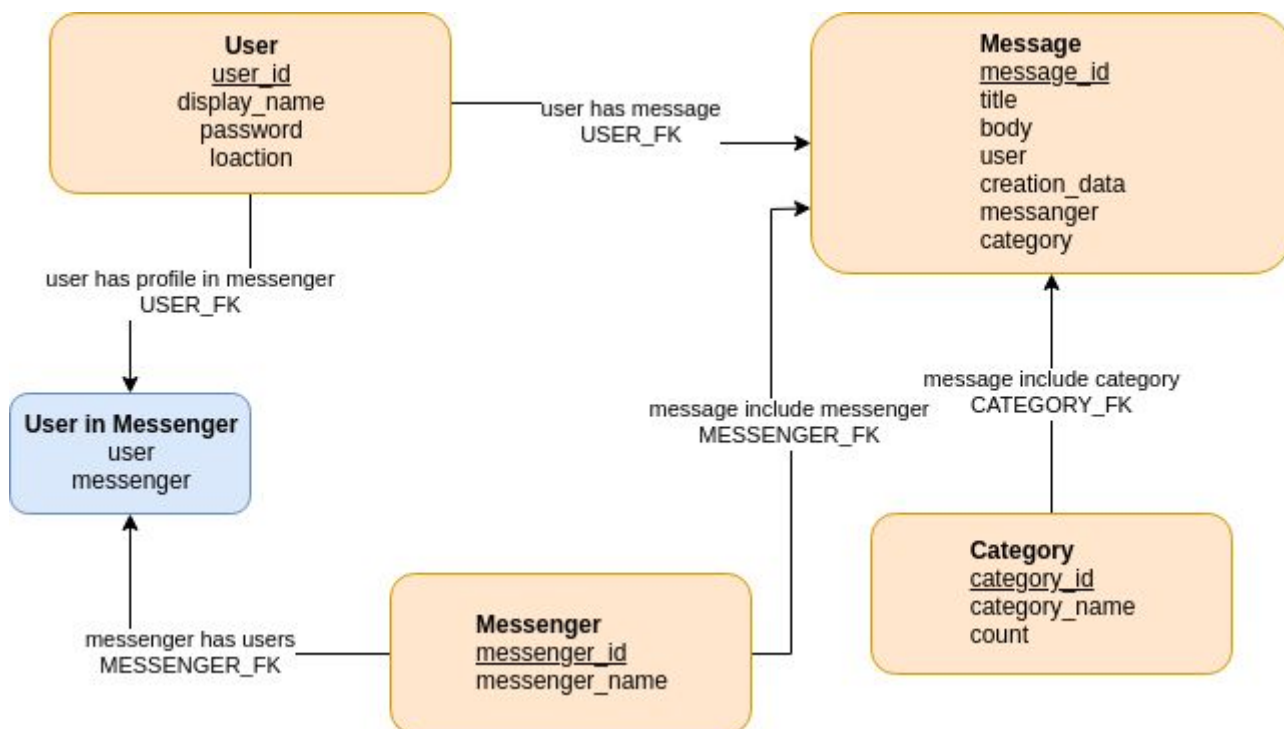


Рисунок 5.2 – Фізична модель даних

Таким чином, отримали модель даних, де кожна сутність містить штучні ключі. Штучні ключі в отриманому сервісі редагуватись користувачем не будуть: відбуватиметься автоінкремент 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 обмежимо користувача у вводі інформації, окрім валідації полів.

ВИСНОВКИ

В ході виконання курсової роботи було визначено наступні етапи впровадження проекту: побудова ієрархії процесів, обґрунтування постановки задачі, визначення сценаріїв перебігу елементарних процесів, визначення класів даних, сутностей та атрибутів, побудова моделей даних, інтеграція відомих рішень та їх покращення новим функціоналом.

Як результат виконання проекту досягнуто: інформаційна система, що дозволяє переглядати, додавати, змінювати та видаляти зміст сутностей. База даних, що взята за основу – PostgreSQL, на Python – SQLAlchemy. Веб-застосунок побудований на базі фреймворків Flask та Bootstrap-4. Ресурс розгорнутий на хостингу Heroku.

Безапережно, розроблена система потребує доопрацювання. Забезпечення аналітики даних (кореляційний, кластерний аналізи, машинне навчання) та застосування UX-дизайну для покращення інтерфейсу додатку — є необхідними задачами перед релізом даного сайту. Також необхідно удосконалити сторінку користувача: додати розміщення фотографій та даних про користувача.

Для повноцінної роботи сервісу необхідно через API месенджерів інтегрувати дані з акаунтів користувачів, які вони матимуть змогу підключити до свого профіля на створеному сервісі. Для user-friendly інтерфейсу можна додати логотипи месенджерів для використання їх при заповненні сутностей даними та відображенні їх у таблицях.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. PostgreSQL 10.11 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/10/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/>.

Додаток А. Use-Case діаграма

