

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

Комп'ютерних наук, фізики та математики

Кафедра комп'ютерних наук і програмної інженерії

**ПРОЄКТУВАННЯ ТА РОЗРОБКА КЛІЄНТ-СЕРВЕРНОГО WEB-  
ДОДАТКУ "СОЦІАЛЬНА МЕРЕЖА" ЗА ДОПОМОГОЮ DJANGO ТА  
REACT**

**Кваліфікаційна робота**

на здобуття ступеня вищої освіти «бакалавр»

	Виконав: студент 4 курсу 441 групи Спеціальності: 121 Програмна інженерія Освітньо-професійної програми: 122 Комп'ютерні науки Попова Д.О.
	Керівник: Шерман М.І.
	Рецензент: _____

Херсон – 2022

<b>Вступ.....</b>	<b>3</b>
<b>Теорія .....</b>	<b>5</b>
<b>Історія виникнення та розвитку соціальних мереж .....</b>	<b>5</b>
<b>Facebook.....</b>	<b>6</b>
<b>Twitter .....</b>	<b>7</b>
<b>Instagram.....</b>	<b>8</b>
<b>Аналіз соціальних мереж .....</b>	<b>9</b>
<b>Принципи та класифікація.....</b>	<b>9</b>
<b>Аудиторія та її платоспроможність .....</b>	<b>11</b>
<b>Комерція у соціальних мережах.....</b>	<b>12</b>
<b>Проектування .....</b>	<b>13</b>
<b>Інструменти проектування.....</b>	<b>13</b>
<b>Функції та особливості соціальних мереж .....</b>	<b>13</b>
<b>Планування проєкту.....</b>	<b>14</b>
<b>Варіанти використання.....</b>	<b>16</b>
<b>Проектування схеми бази даних .....</b>	<b>16</b>
<b>Діаграмма класів соціальної мережі .....</b>	<b>18</b>
<b>Практика .....</b>	<b>19</b>
<b>моделі реляційної бази даних .....</b>	<b>19</b>
<b>Основні функціональні алгоритми .....</b>	<b>22</b>
<b>Висновок практичної роботи.....</b>	<b>28</b>
<b>Висновок .....</b>	<b>30</b>

## Вступ

**Соціальні мережі** – це інтернет платформи для спілкування та обговорення різноманітних тем, а також інших суспільних взаємодій.

У сучасному світі соціальні мережі можуть використовуватися для роботи та розваг. Вони дозволяють людям координувати взаємодію один з одним і мають широкий спектр функцій.

**Соціальні мережі** з кроку їхньої появи все більше потрапляють у наше життя.

Для багатьох соціальні мережі стали основним місцем, де можна провести час в Інтернеті.

Сьогодні багато людей тяжко представляють життя без соціальних мереж, які дозволяють порозумітися з абсолютно незнайомими людьми. Вони руйнують ці бар'єри не лише через спілкування, а й через відкриту інформацію та дискусії на різні теми.

Сьогодні соціальні мережі перестали бути просто безпечними сервісами для обміну інформацією та досвідом між користувачами — вони стали потужними платформами, які формують глобальний інформаційний порядок денний.

## Актуальність теми

Важливість соціальних мереж сьогодні не можна ігнорувати. Соціальні мережі використовують мільйони людей у всьому світі, що робить їх інструментом інтернет-маркетингу.

Нові статистичні дані від компанії **Global Web Index** за 2020 рік продемонстрували, що на початку 2020 року понад 4,5 мільярда людей у всьому світі користуються інтернетом, що на 7 відсотків більше, ніж роком раніше.

З них 3,8 мільярда користувачів соціальних мереж, що становить майже 60 відсотків населення світу. Також дослідження **Global Web Index** показують, що середній користувач проводив в Інтернеті майже 7 годин на день, що майже на чверть годин більше, ніж у попередньому кварталі. На соціальні мережі припадає понад третина всього часу, проведеного в Інтернеті. На сьогодні користувачі соціальних мереж у середньому витрачають 2,5 години на день, що безумовно свідчить про подальшу перспективність соціальних платформ.

Буквально у всіх основних громадських мережах світу за кінець року важливо збільшилася сумарна чисельність користувачів. У період з липня по вересень 2020 року Instagram зацікавив найбільшу кількість нових користувачів.

За власними даними компанії, рекламна аудиторія *Instagram* за останні три місяці зросла на 76 мільйонів людей, досягнувши 1,16 мільярда користувачів на початку жовтня 2020 року, що на 7 відсотків більше, ніж у попередньому кварталі.

Вплив соціальних мереж на користувачів з кожним роком зростає, а відомі громадські мережі, такі як Twitter, Facebook, Instagram можуть привести на свою платформу політиків і різних зоряного неба. Це дозволяє громадським соціальним мережам стати частиною громадянського суспільства.

### **Мета дослідження**

Метою дослідження є проектування та розробка соціальної мережі. Дослідження складається з трьох частин: *Теоретичної, проєктувальної та практичної*.

Для *Теоретичної* частини було поставлено такі завдання:

- Розкрити поняття «соціальна мережа» і зрозуміти процес становлення цього феномена;
- Історія виникнення;

- Виявити поняття «соціальна мережа» та усвідомити процес становлення цього парадоксу;
- Розглянути класифікації соціальних мереж;
- Проаналізувати основну аудиторію соціальних мереж.

Для Проектувальної частини було поставлено такі завдання:

- Виділити основні архітектурні сутності соціальних мереж;
- Створити *UML* діаграми, які описуватимуть структуру соціальної мережі.

Для Практичної частини було поставлено такі завдання:

- Розглянути моделі реляційної бази даних на базі ORM Django (models);
- Розглянути основні функціональні алгоритми (controllers);
- Висновок практичної роботи

### **Об'єкт дослідження**

Об'єктом дослідження роботи є соціальні мережі.

### **Предмет дослідження**

Предметом дослідження роботи є принципи проектування та класифікації соціальних мереж.

## **Теорія**

### **Історія виникнення та розвитку соціальних мереж**

У 1954 році поняття «соціальна мережа» було введено англійським соціологом *Джеймсом Барнесом*. У ті дні сформульовані поняття *Джеймса Барнеса* були далекі від нинішньої технічної реалізації глобальної мережі Інтернету і тлумачилось так:

*Соціальна мережа* – це багатокomпонентна структура, яка складається з вузлів, які є соціальними об'єктами та взаємовідносинами між ними. Інакше кажучи, *Соціальна мережа* – це група людей, між якими є певні соціальні взаємовідносини, двостороння чи одностороння взаємодія. А 1959-1961 р. Угорські вчені Альфред Реньї та Пол Ердош опублікували вісім статей, що вивели принципи соціальних мереж. Водночас, соціальні мережі стали основною причиною того, що кількість часу, що проводиться нинішніми користувачами в Інтернеті, на сьогоднішній день зростає.

Перші соціальні платформи почали з'являтися вже у 90-х роках минулого століття. Потім ринок вибороли: ICQ, Egroups/OneList, Evite. Такі онлайн-сервіси, як правило, не є реальними соціальними мережами, але стали незамінною основою для подальшого розвитку онлайн-сервісів для спілкування та взаємодії у мережі.

Вже 1995 року було офіційно зареєстровано створення першої повноцінної соціальної мережі — це був *classmate* американської онлайн-платформи. Проект мав великий успіх, у результаті протягом наступних кількох років стали з'являтися десятки таких онлайн-сервісів.

У 2003 році було запущено професійну соціальну мережу *LinkedIn* з 740 мільйонами користувачами, вона є популярною до сьогодні.

## **Facebook**

Facebook було засновано у 2004 році студентами Гарварду. Спочатку мережа призначалася лише для студентів. У міру зростання популярності користувачами мережі могли стати студенти Єльського, Бостонського,

Массачусетського та інших університетів. Через рік користувачами Facebook могли стати студенти більшості університетів США та Канади.

## **Twitter**

У 2006 році було створено соціальну мережу Twitter, яка на той час називалася Twtr. У тому ж році Джек Дорси (творець соціальної мережі Twitter) зробив перший у світі «твіт», у якому містилася фраза Just setup my twtr. Восени аудиторія платформи зростає до кількох тисяч користувачів, які активно соціалізували своїх друзів та знайомих.

У 2007 році Twitter було перетворено на окрему компанію. Джек Дорсі, розробник сервісу, став генеральним директором. Засновники та розробники активно розвивали функціонал платформи: у середині 2007 року була додана можливість відстеження інших користувачів соціальної мережі.

У тому ж році були додані можливість пошуку за профілем користувача, система коментарів, система хештегів та інші функції, які покращили систему та популяризували сервіс. А у квітні 2009 року сервіс мікроблогів Twitter вплинув на політичний процес у Молдові. У зв'язку з недемократичними змінами у державі місцеві молодіжні організації запустили на своїх твіттер-каналах флешмоб — усіх незадоволених запросили на центральну площу Кишинева із запаленими свічками в руках. У результаті в центрі столиці зібралося близько 10 тисяч людей.

Завдяки Твіттеру стали набирати популярності політичні теми — лідери різних країн, послы, міністри та інші офіційні особи стали користуватися соціальними мережами, особливо Твіттером. Завдяки цьому Twitter поки є найполітизованою соціальною мережею. Тому твіттер у багатьох країнах є механізмом суспільного життя.

## **Instagram**

6 жовтня 2010 року Instagram став доступним для завантаження в AppStore.

За тиждень Instagram став найзавантаженішим додатком на різних платформах.

За перший місяць кількість користувачів соціальної мережі сягнула 1 мільйона. Після отримання настільки запаморочливих успіхів, засновники активно взялися активно доробляти та покращувати *Instagram*.

Корпорація проводила значну роботу над соціальною мережею: нові фільтри, система хештегів та категорій, публікація відео, фото та відео у різних форматах та фільтрах. У цей же рік соціальна мережа стала можливою для використання на Android.

9 квітня 2012 року було підписано угоду між Кевіном Сістромом (власником та розробником Instagram) та Facebook. Результатом стало придбання Instagram компанією Facebook за 1 мільярд доларів. За роки, що минули з того часу, як Facebook придбала Instagram, програма була значно покращена за допомогою історій, IGTV, прямих трансляцій та багатьох інших успішних функцій. Найголовніше, Facebook розробив для Instagram систему монетизації – таргетовану рекламу, яка згодом окупила усі втрати. З 2012 року аудиторія Instagram значно зростає.

На цю мить в інстаграмі сумарно зареєстровано 1 мільярд акаунтів.

## **Висновок**



Кожна соціальна мережа має свою унікальну історію виникнення, успіхів та невдач. Деякі з них залишилися популярними досі, а деякі були приречені на забуття. Проте всі історії невдач і успіхів — це дорогоцінний досвід, без якого ми не побачили б соціальні мережі в тому вигляді, в якому вони існують сьогодні.

## **Аналіз соціальних мереж**

### **Принципи та класифікація**

Вихідним завданням аналізу соціальних мереж є їх класифікація на окремі групи, що дозволить виділити спільні зв'язки між різними групами соціальних мереж, їх потенціал, аудиторію, ринкову спрямованість та можливості переходу групи мереж, соціальні до стану інших.

Різноманітність послуг в Інтернеті ускладнює їхню ідентифікацію, але загальними характеристиками можуть бути:

- Послуги;
- Основним джерелом контенту соціальної мережі є користувач;
- Основним зображенням, характеристикою та даних (ім'я, фамілія, сімейний статус інтереси) на вебсервісі є профіль користувача;
- Надсилання особистих повідомлень;
- Обмін ресурсами (файлами, зображеннями та іншими медіаформатами);
- Групи користувачів, що дозволяє об'єднувати профілі користувачів із загальних інтересів;
- Ґрунтуються на вебінтерфейсах;

Класифікації соціальних мереж:

- *Особисте та ділове спілкування* (Facebook, LinkedIn, VK);
- *Навчальні мережі* (wikipedia.org);
- *Хостинг медіаконтенту* (YouTube, Vevo, DailyMotion, Vimeo);
- *Стрімінгові сервіси* (Twitch, Periscope, LiveStream, Mixer);
- *Мережі колективних обговорювань* (Mail.ru, Quora);
- *Мережі блогів\мікроблогів* (Twitter, LiveJournal);
- *Політичні мережі* (Politico);
- *Комерційні мережі* (Xing, TeamFinding.com);
- *Універсальні розважальні мережі* (IMDb, Last.fm);

Самі послуги розділені по регіонах (організації, країни, доменні імена верхнього рівня), оскільки деякі послуги можуть бути недоступні в деяких країнах. Ці класифікації дозволяють класифікувати послуги відповідно до їх технічних можливостей, не впливаючи на рівень соціальної взаємодії між користувачами. Цей підхід загалом підходить для розробників веб-сервісів та людей, які цікавляться технічною стороною проекту. Вихідним фактором соціальних мереж мають бути технічна структура, обмеження та архітектурні залежності веб-сервісу.

*За типом соціальні мережі можуть поділятися на:*

- Глобальні;
- Тематичні;

Соціальні мережі можуть бути класифіковані за інтересами:

- Аудиторією;
- Професією;
- Соціальним статусом;

По значенню вузлів мережі можуть бути зв'язаними з:

- Користувачами;
- Контентом;

### **Аудиторія та її платоспроможність**

Потенційна аудиторія соціальних мереж відіграє важливу роль, адже від цього залежать такі показники, як:

- Подальший розвиток соціальної мережі;
- система монетизації;
- платоспроможність;

Більшість аудиторії Facebook становлять жінки, але порівняно з Instagram і Twitter аудиторія Facebook зріліша: більшості з них від 25 до 35 років.

За даними MediaScore за 2017 рік, аналіз платоспроможності виглядає так:

- «ВКонтакте»;
- «Однокласники»;
- «Інстаграм»;
- "Фейсбук"

Однак ці дані неспроможні дати точну інформацію про кредитоспроможності користувачів, оскільки враховують цінову категорію товарів. Однак, згідно з дослідженнями, розрахувати точну мережу розчинників практично неможливо.

Але все ж таки загальна тенденція всіх соцмереж — дорослішання аудиторії, що значно підвищує кредитоспроможність.

На основі яких характеристик користувачі соціальних мереж роблять покупки:

- Відгук користувачів щодо товару;

- Популярність товарів у друзів;
- Активність в тематичних групах соціальних мереж перед покупкою;
- спостереження за контентом, який публікують бренди у соціальних мережах;

Вплив соціальних мереж на інтернет комерцію складно переоцінити.

Вони дозволяють малому бізнесу розвиватися, набирати медіа популярність та просувати свій вплив на ринку мінімальними вкладеннями.

Великому бізнесу та корпораціям соціальні мережі дозволяють вигадувати креативні рекламні компанії, завдяки яким позиції на ринку посилюються

### **Комерція у соціальних мережах**

Соціальна комерція, користувачів соціальних мереж з кожним роком стає дедалі більше. Отже, зростає потреба в охопленні та визначенні різних груп населення.

Для цього маркетологи, спеціалісти у галузі реклами та розробники продуктів звертають увагу на такі споживчі характеристики:

- Платоспроможність;
- Географічне положення;
- Вік;
- Гендерна приналежність;
- Інтереси;

Маркетологи створюють методи привернення уваги споживача, а саме:

- Групи в соціальних мережах;
- Конкурси та опитувальники;
- Пряма реклама;

- Інтегрована реклама;
- Створення сторінок та груп в соціальних мережах

Однією з найбільших перешкод для ведення малого бізнесу у соціальних мережах є анонімність користувачів соціальних мереж. Учасник соціальної мережі повідомляє лише частину інформації, яку вважає достатньою для публікації. Це дуже мало оцінки споживачем. Перешкода тут полягає в тому, що приховування деяких деталей може вплинути на вашу репутацію та інші ділові рішення.

## **Проектування**

### **Інструменти проектування**

Для розробки проекту було використано:

- **Хостинг:**
  - GitHub.com
- **UML:**
  - StarUML;
  - Altova;
  - Visio (Visual Studio UML);
  - Draw.IO
- **Мови програмування:**
  - Python;
  - JavaScript;
  - SQL;

### **Функції та особливості соціальних мереж**

Одна з головних функціональних особливостей соціальних мереж є пошук людей та встановлення соціальної взаємодії між користувачами мережі. За допомогою функціональних можливостей соціальної мережі користувач може створити свій віртуальний портрет – профіль, в якому є можливість вказати:

- Ім'я;
- Інтереси;
- Дату народження;
- Місце навчання;
- Місце роботи;
- Життєві цілі

Завдяки системі профілів є можливість використовувати механізми пошуку за іменем, гендером, датою народження для пошуку користувачів, друзів, колег та ін. *Функції соціальних мереж:*

- Система профілів;
- Взаємодія користувачів (система коментарів, система приватних чатів);
- Обмін медіаконтентом (фото, відео);
- Система груп та спільнот.

Основними особливостями соціальної мережі є:

- Уніфікація інтерфейсу (Усі сторінки соціальної мережі мають однаковий інтерфейс; відрізняється лише контент);
- Уніфікація функціоналу (функціонал соціальної мережі визначає активність користувача);
- Інтерфейс зворотного зв'язку (лайки, репости, «поділитися у соціальних мережах»)

## **Планування проєкту**

У цьому розділі ми почнемо проектувати власну соціальну мережу, яка матиме назву *ISpeaks*. Соціальна мережа буде являти собою вебсайт, в якому буде реалізовані базові соціальні та технічні потреби для соціальної мережі, а саме буде розроблено:

- Система повідомлень;
- Система дозволу до контенту (*приватний, публічний*)
  - Створювати повідомлювання можуть тільки зареєстровані користувачі;
  - Видаляти повідомлення можуть тільки зареєстровані користувачі;
  - Використати соціальні оцінки контенту («лайк») можуть тільки зареєстровані користувачі;
  - Робити «репост» контенту можуть тільки зареєстровані користувачі;
- Система особистих повідомлень
  - Розмежування можливості писати особисті повідомлення (користувачі, які були додані у «чорний список» не можуть відсилати особисті повідомлення).
- Система акаунтів;
  - Реєстрація;
  - Вхід до облікового запису (*Login*);
  - Вихід з облікового запису (*Logout*);
  - Профіль користувача;
  - Система відновлення паролів;
- Система відстежування (*Follows*);
- Система друзів (*Friends*);
- Система особистих статусів (користувачі мають можливість вказати у профілі свій статус);

- Стрічки активності (дозволяє користувачам, які є друзями, відстежувати дії один одного);
- Система закладок (zareєстровані користувачі соціальних мереж мають можливість додати контент до закладок);

## Варіанти використання

Перед тим як переходити до етапу технічного проєктування необхідно визначити варіанти використання функціоналу різними видами *користувачів*. *Користувач* стосовно моделі системи є сутністю, яка використовує функціонал системи у своїх цілях. Модель користувача може мати безліч ролей. Ролі відрізняються один від одного сукупністю доступного для користувача функціоналу системи соціальної мережі.

### Ролі користувачів:

- Користувач;
- Модератор;
- Адміністратор

### Розподілений функціонал:

- Користувач має звичайну роль, яка дозволяє йому використовувати функціонал соціальної мережі в цілях споживача системи;
- Модератор має роль наглядача в групах соціальних мереж. Має можливість блокувати, коригувати та попереджувати користувачів системи;
- Адміністратор має роль наглядача соціальної мережі в цілому.

Відобразити таку поведінку можна за допомогою use case діаграм.

## Проектування схеми бази даних



Реляційні бази даних мають недолік щодо розробки соціальних мереж. Через низьку пропускну здатність, реляційні бази даних погано поєднуються з великою кількістю одночасних запитів. Тому для цих цілей найчастіше використовують (у таких соціальних мережах як *Facebook*, *Twitter*) графові бази даних (по класифікації відносяться до *NoSQL* баз даних).

На початковому етапі важко визначати схему бази даних, оскільки змінювати схему «на льоту» завжди проблемно. І тому існує два підходи до проєктування бази даних:

- Функціональний підхід – застосовується тоді, коли відомі мінімальні функції деякої групи об'єктів предметної області, які мають бути описані.
- Предметний підхід – у схему включається найбільш суттєві об'єкти та взаємозв'язки. Схема бази даних жорстко не фіксується.

Для нашої соціальної мережі було б доцільніше орієнтуватися на конкретні завдання чи базові функціональні потреби користувачів. Для цих цілей можна поєднати графову та реляційну моделі, що підтримує роботу з підтримки бази даних у актуальному стані.

Почнемо проєктування бази даних з таблиці *повідомлення*.

Таблиця має мати такі поля:

- *Id* – первинний ключ;
- *Parent\_id* – рекурсивний зв'язок на запис із поточної таблиці;
- *User* – зовнішній ключ до таблиці користувача;
- *Content* – текстове поле для контенту;
- *Image* – бінарне поле для зберінання зображення;
- *Timestamp* – дата створення повідомлення

Таблиця *користувача* має такі поля:

- Id – первинний ключ;
- Username – ім'я користувача;
- Password – пароль користувача;
- Email – пошта користувача;
- Date\_created – дата створення облікового запису;
- Date\_updated – остання дата зміни облікового запису

\*Пароль користувача не зберігається у базі даних у відкритому вигляді. Це необхідно для того, щоб у разі XSS-атаки, хакер не зміг скористатися паролем і отримати доступ до облікового запису користувача. За замовчуванням *Django* хешує паролі користувачів із «сіллю» для цих цілей.

Таблиця *облікового запису* має такі поля:

- Id – первинний ключ;
- User – зовнішній ключ до таблиці користувача;
- Followers – кількість фоловерів;
- Location – інформація про місце знаходження;
- Gender – інформація про гендер користувача;
- Date\_created – дата створення профілю;
- Date\_updated – дата останнього оновлення профілю;

## **Діаграма класів соціальної мережі**

Програма на python складається з таких класів:

- UserClass – клас аутентифікації та пошуку користувача;
- RoleClass – роль користувача у соціальній мережі;
- PermissionsClass – система розмежування контенту;
- PostClass – створити, змінити та видалити відгук;
- ShareClass – «репост» відгуку

## **Практика**

**моделі реляційної бази даних**

Для опису схеми реляційної бази даних використовується інтегрована в Django Framework ORM (*Object Relation Map*).

*Django ORM* — інструмент фреймворку Django, який дозволяє взаємодіяти з базами даних, об'єкти python, а не SQL-запити. Він відноситься до типу ORM, що реалізує шаблон Active Record. Загальна суть шаблону в тому, що кожній таблиці у додатку відповідає одна модель. Моделі визначають структуру даних, що зберігаються, включаючи типи полів і, можливо, їх максимальний розмір, значення за замовчуванням, параметри списку вибору, текст довідки для документації, текст міток для форм і т.д.

### Модель твіта

```
15 class Tweet(models.Model):
16     id = models.AutoField(primary_key=True)
17     user = models.ForeignKey(
18         User,
19         on_delete=models.CASCADE,
20         related_name='tweets',
21         related_query_name='tweets_query')
22     likes = models.ManyToManyField(
23         User,
24         related_name='likes',
25         related_query_name='likes_query',
26         through=TweetLike)
27     parent = models.ForeignKey('self', null=True, blank=True, on_delete=models.SET_NULL) # Retweet field
28     text_content = models.TextField(blank=True, null=True)
29     file_content = models.FileField(upload_to='files/', blank=True, null=True)
30     date_created = models.DateTimeField(auto_now_add=True)
31     date_updated = models.DateTimeField(auto_now=True, null=True)
```

Ця модель є основою в проекті, за допомогою цієї функціональності користувачі можуть взаємодіяти між собою.

Модель твіту має такі основні поля, як:

- **ID** – ідентифікатор твіту, обов'язкове поле;

- **user** – зовнішній ключ типу «багато-до-одного» (\*-1), необхідний для зв'язку *користувачів* та *твітів*;
- **likes** – це зовнішній ключ «багато-до-багатьох» (\*-\*), який має містити ідентифікатор користувача, якому сподобався твіт;
- **parent** – це зовнішній ключ «багато-до-одного» (\*-1), який посилається сам на себе (рекурсивний зв'язок у реляційній таблиці). Це поле необхідне для створення ретвіта;
- **text\_content** – текстове поле. Містить текстовий вміст твіту;
- **file\_content** – текстове поле, яке містить шлях до файлу;
- **date\_created** – текстове поле, яке містить дату створення твіту;
- **date\_updated** – текстове поле, яке містить дату зміни твіту;

### Допоміжна модель твіту

```

7 class TweetLike(models.Model):
8     user = models.ForeignKey(User, on_delete=models.CASCADE)
9     tweet = models.ForeignKey("Tweet", on_delete=models.CASCADE)
10    timestamp = models.DateTimeField(auto_now_add=True)
11
12    def __str__(self):
13        return 'user: %s; tweet: %s' % (self.user.id, self.tweet.id)
14

```

Ця модель є *through-моделлю*, яка містить зіставлення ідентифікаторів між моделями **Tweet** та **User**.

Використовується для обслуговування зв'язку «багато-до-багатьох» (\*-\*) між таблицями **User** і **Tweet**.

Має такі основні поля, як:

- **user** – містить ідентифікатори з моделі **User**;
- **tweet** – містить ідентифікатори з моделі **Tweet**;
- **timestamp** - дата створення запису в таблиці

### Модель профіля користувача

```

12 class Profile(models.Model):
13     user = models.OneToOneField(User, on_delete=models.CASCADE)
14     location = models.CharField(max_length=220, null=True, blank=True)
15     bio = models.TextField(blank=True, null=True)
16     timestamp = models.DateTimeField(auto_now_add=True)
17     updated = models.DateTimeField(auto_now=True)
18     followers = models.ManyToManyField(User, related_name='following', blank=True)

```

Модель профілю користувача.

Має такі основні поля, як:

- **user** – зовнішній ключ до моделі користувача (тип зв'язку «один-до-одного»);
- **location** – розташування користувача;
- **bio** – гендер користувача;
- **timestamp** – дата створення профілю;
- **updated** - дата оновлення профілю;
- **followers** - зовнішній ключ типу «багато-до-багатьох» (\*-\*). Містить користувачів, підписаних на профіль

### Допоміжна модель профілю користувача

```

7 class FollowerRelation(models.Model):
8     user = models.ForeignKey(User, on_delete=models.CASCADE)
9     profile = models.ForeignKey("Profile", on_delete=models.CASCADE)
10    timestamp = models.DateTimeField(auto_now_add=True)
11

```

Ця модель є *through-моделлю*, яка містить зіставлення ідентифікаторів між моделями **User** та **Profile**. Завдяки цій моделі є можливість зберігати підписки на профіль користувача.

Має такі основні поля, як:

- **user** – містить ідентифікатори з моделі **User**;
- **profile** – містить ідентифікатори з моделі **Profile**;
- **timestamp** - дата створення запису в таблиці

## Основні функціональні алгоритми

Основні функціональні алгоритми проекту ми опишемо за допомогою `django views`.

**Django Views** є ключовим компонентом програм, створених за допомогою `django`. У найпростішому випадку це функція або клас Python, який приймає веб-запит і повертає веб-відповідь.

### *отримання доступу до ресурсу*

```

29 @api_view(['GET']) # К контроллеру можно получить доступ только с помощью http метода GET
30 def tweets_list_view(request, *args, **kwargs):
31     # Вывод списка твитов
32     qs = Tweet.objects.all()
33     serial = TweetSerializer(qs, many=True) # Сериализация объекта твитов в примитивный тип Python
34     return Response(serial.data, status=200) # Возвращает JSON с данными и статусом 200 - запрос выполнен успешно

```

В даному коді ми отримуємо з бази даних поточні твіти та повертаємо їх у форматі JSON з статусом 200.

### *отримання конкретного твіту*

```

37 @api_view(['GET']) # К контроллеру можно получить доступ только с помощью http метода GET
38 def tweet_detail_view(request, tweet_id, *args, **kwargs):
39     # Вывод детальной информации о твите
40     qs = Tweet.objects.filter(id=tweet_id)
41     if not qs.exists():
42         # Если твит не найден - возвращаем информацию об ошибке в JSON формате
43         return Response({'message': 'tweet not found'}, status=404)

```

У даному коді ми отримуємо **id** твіт через **url**, за яким ми знаходимо його в базі даних. Якщо твіт не знайдено, тоді повертаємо повідомлення про помилку зі статусним кодом 404.

Якщо твіт знайдено, формуємо з об'єкта **python** відповідь у форматі **JSON** з статусним кодом 200.

### *створення об'єкта твіту*

```

50 @api_view(['POST']) # К контроллеру можно получить доступ только с помощью http метода POST
51 @permission_classes([IsAuthenticated]) # Не аутентифицированный пользователь не может получить доступ к контроллеру
52 # Будет возвращён http код 403 - пользователь не аутентифицирован
53 def tweet_create_view(request, *args, **kwargs):
54     ...
55     Контроллер создания твита:
56     - Поддерживает только метод HTTP POST
57     - Не аутентифицированные пользователи создавать твиты не могут
58     - Поддержка AJAX запросов
59     - Используем TweetCreateSerializer для создания объекта твита
60     ...
61     serial = TweetCreateSerializer(data=request.data or None) # Заполняем сериализатор данными, которые пришли в HTTP body
62     if serial.is_valid(): # Проверка данных полей сериализатора на валидность
63         tweet_obj = serial.save(user=request.user) # Создаём новый объект твита, указываем отправившего твит пользователя.
64         serial = TweetSerializer(tweet_obj)
65         return Response(serial.data, status=201) # Возвращает JSON с данными и статусом 201 - успешно создан новый объект
66     if serial.errors: # Данные полей сериализатора не валидны. Проверяем наличие ошибок
67         return Response({'error_messages': serial.errors}, status=400) # Возвращаем информацию об ошибке.
68
69     return Response({}, status=404) # Поведение не обработано...

```

Цей код обробляє **POST**-запит створення твіту. Приймає він у тілі HTTP-запиту рядок типу `text_content=tweet_content&...`

Далі десеріалізує контент твіту та перевіряє його на валідність.

Якщо контент валідний, створюємо об'єкт твіту та повертаємо статусний код 200, інакше повертаємо знайдені помилки при валідації до користувача.

Також варто зазначити, що функціонал доступний лише для авторизованих користувачів.

*видалення об'єкта твіту*



```

72 @api_view(['DELETE', 'POST']) # К контроллеру можно получить доступ только с помощью http метода DELETE и POST
73 @permission_classes([IsAuthenticated])
74 def tweet_delete_view(request, *args, **kwargs):
75     ...
76     Контроллер удаления твита:
77         - Контроллер получает HTTP body 'id=xxx'
78         - id - обязательный параметр
79     ...
80     tweet_id = request.POST.get('id')
81     # qs = Tweet.objects.filter(Q(id=int(tweet_id)) & Q(user=request.user))
82     qs = Tweet.objects.filter(id=int(tweet_id))
83     if not qs.exists():
84         return Response({'message': 'You cannot delete this tweet'}, status=404)
85
86     tweet = qs.first()
87     if request.user.id != tweet.user.id:
88         return Response({'message': 'Cross user delete error'}, status=400)
89
90     tweet.delete()
91     return Response({'message': 'Tweet was deleted'}, status=200)

```

Цей код обробляє лише **POST** та **DELETE** запити створення твіту. У тілі HTTP-запиту отримуємо **id** яким знаходимо і видаляємо твіт. Якщо твіт не знайдено, повертаємо помилку користувачеві.

### *управління дії над твітами*

```

94 @api_view(['POST']) # К контроллеру можно получить доступ только с помощью http метода DELETE и POST
95 @permission_classes([IsAuthenticated])
96 def tweet_action_view(request, *args, **kwargs):
97     ...
98     Контроллер обработки действий над твитом.
99         - Контроллер получает HTTP body 'id=xxx&action=yyy&content=zzz'
100         - id - обязательный параметр
101         - Действия над твитом: like, unlike, retweet
102     ...
103     action_serial = TweetActionSerializer(data=request.data) # Заполняем сериализатор данными, которые пришли в HTTP body
104     if action_serial.is_valid(): # Проверка данных полей сериализатора на валидность
105         data = action_serial.validated_data # Получаем проверенные данные
106         tweet_id = data.get('id')
107         tweet_action = data.get('action')
108         tweet_content = data.get('content')
109         qs = Tweet.objects.filter(id=tweet_id)
110
111         if not qs.exists():
112             # Твит не найден! возвращаем информацию об ошибке со статусом 404 - ресурс не найден.
113             return Response({'message': 'Sorry, tweet not found!'}, status=404)
114         tweet = qs.first()

```

```

115     if tweet_action.lower() == 'socialscore':
116         ipdb.set_trace()
117         user = request.user
118         if tweet.likes.contains(user):
119             tweet.likes.remove(user)
120         else:
121             tweet.likes.add(user)
122         tweet_serial = TweetSerializer(tweet) # Сериализуем объект твита в примитивный тип Python
123     elif tweet_action.lower() == 'retweet':
124         parent_tweet = tweet
125         new_tweet = Tweet.objects.create(
126             text_content=tweet_content,
127             user=request.user,
128             parent=parent_tweet)
129         tweet_serial = TweetSerializer(new_tweet)
130         return Response(tweet_serial.data, status=201)
131
132     if action_serial.errors: # Если поля сериализатора не валидны - возвращаем сообщение об ошибке с кодом 400 - запрос не обработан
133         return Response({'message': action_serial.errors}, status=400)
134
135     return Response(tweet_serial.data, status=200) # Возвращаем данные обработанного твита в формате JSON со статусным кодом 200

```

Цей код обробляє лише **POST-запит** обробки дій над твітами. Приймає він у тілі HTTP-запиту рядок типу *id=xxx&action=ууу&content=zzz*. *id* - обов'язковий параметр.

Дії над твітом:

- **like;**
- **unlike;**
- **retweet**

Якщо поля серіалізатора не валідні – повертаємо повідомлення про помилку з кодом 400 – запит не оброблено.

Повертаємо дані обробленого твіту у форматі JSON зі статусним кодом 200, якщо одна з дій над твітом виконана успішно.

*Реєстрація користувача*

```

34 def register_view(request, *args, **kwargs):
35     form = UserCreationForm(request.POST or None)
36     if form.is_valid():
37         user = form.save(commit=True)
38         user.set_password(form.cleaned_data.get("password1"))
39         # send a confirmation email to verify their account
40         login(request, user)
41         return redirect("/")
42     context = {
43         "form": form,
44         "btn_label": "Register",
45         "title": "Register"
46     }
47     return render(request, "accounts/auth.html", context)

```

Якщо контролер приймає POST-метод, то спочатку робимо валідацію поля, а потім створюємо обліковий запис.

Якщо контролер приймає GET-метод, то повертаємо порожню форму користувачеві.

### *Login користувача*

```

8 def login_view(request, *args, **kwargs):
9     form = AuthenticationForm(request, data=request.POST or None)
10    if form.is_valid():
11        user_ = form.get_user()
12        login(request, user_)
13        return redirect("/")
14    context = {
15        "form": form,
16        "btn_label": "Login",
17        "title": "Login"
18    }
19    return render(request, "accounts/auth.html", context)

```

Якщо контролер приймає POST-метод, то спочатку робимо валідацію поля, а потім login облікового запису.

Якщо контролер приймає GET-метод, то повертаємо порожню форму користувачеві.


*Logout користувача*

```
21 def logout_view(request, *args, **kwargs):
22     if request.method == "POST":
23         logout(request)
24         return redirect("/login")
25     context = {
26         "form": None,
27         "description": "Are you sure you want to logout?",
28         "btn_label": "Click to Confirm",
29         "title": "Logout"
30     }
31     return render(request, "accounts/auth.html", context)
```


контроллер приймає тільки GET-метод HTTP запиту.  
за допомогою функції `logout` виходимо із сесії

**Висновок практичної роботи**

[Tweet](#)


 **@brandnewuser**

Retweet via @brandnewuser

 **Justin Mitch @cfe**  
Hello World


[View](#)

[1 Likes](#) [Unlike](#) [Retweet](#) [View](#)


 **Justin Mitchel @justin**

Hello world!

[0 Likes](#) [Unlike](#) [Retweet](#) [View](#)

 **Justin Mitchel @justin**

Retweet via @justin

 **Justin Mitch @cfe**  
hello there!

[View](#)

[0 Likes](#) [Unlike](#) [Retweet](#) [View](#)



Justin Mitch @cfe

hello there!

[2 Likes](#) [Unlike](#) [Retweet](#)

Justin Mitch @cfe

2 followers

1 following

Newport Beach

I love to build things.

[Follow](#)

Justin Mitch @cfe

2 followers

1 following

Newport Beach

I love to build things.

[Follow](#)

Justin Mitch @cfe

Hello World

[1 Likes](#) [Unlike](#) [Retweet](#) [View](#)

## Висновок

Результатом цієї роботи є аналіз, класифікація, проєктування та створення соціальної мережі за допомогою інструментів програмування Django та React. Зроблений аналіз дозволяє нам зробити висновок, що соціальні мережі дають

нам можливість для розвитку спілкування з користувачами різних соціальних статусів, поглядів та ін.

Сьогодні світ соціальних мереж охоплює сотні мільйонів користувачів з різних країн та міст. Соціальні мережі здатні об'єднати людей абсолютно різних захоплень та політичних поглядів, визначити порядок денний та ін. На сьогодні соціальні мережі – частина нашої культури, яка дозволяє вести громадянський діалог.

### Список використаних джерел

- Історія та розвиток соціальних медіа [Електронний ресурс] – режим доступу: <https://habr.com/post/72136/> HYPERLINK "https://habr.com/post/72136/" HYPERLINK "https://habr.com/post/72136/"
- Історія виникнення та розвитку соціальних мереж [Данько Ю. А. Соціальні мережі як форма сучасної комунікації: плюси і мінуси.
- Калініна Г. М., Малюга М. М. Краудфандінг як інструмент розвитку соціального капіталу в цифровому суспільстві. (2019): 113-118.
- Крэг Ларман. Застосування UML 2.0 та шаблонів проектування.
- Мова UML. Посібник користувача – Граді Буч, Джеймс Рамбо, Івар Якобсон.
- Бази даних та UML. Проектування – Роберт Дж. Мюллер.
- Уніфікований процес розробки програмного забезпечення – А. Якобсон, Г. Буч, Дж. Рамбо
- Django for beginners: build websites with Python and Django - William S. Vincent
- Learning React: functional web development - Alex Banks.