



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №2
Технології розробки програмного забезпечення
«ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ
ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.
КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»

Виконала:
студенка групи ІА-24
Орловська А. В.
Перевірив:
Мягкий М. Ю.

Київ 2024

Зміст

Короткі теоретичні відомості.....	3
Хід роботи	5
Діаграми варіантів використання (Use-case diagram).....	6
UML діаграма класів	9
Зображення структури бази даних	10
Посилання на репозиторій.....	13
Висновок	13

Тема: ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ

Мета: Розробити діаграми варіантів використання та класів.

Короткі теоретичні відомості

Діаграма варіантів використання (Use Case Diagram)

Діаграма варіантів використання описує функціональні можливості системи з точки зору її користувачів (акторів).

- **Елементи:**
 - **Актори:** користувачі системи (люди, пристрої чи інші системи).
 - **Варіанти використання (Use Cases):** функції чи послуги, які система надає акторам.
 - **Зв'язки:** відображають взаємодію акторів із варіантами використання.
- **Призначення:** допомагає зрозуміти, що система повинна робити, і визначити її функціональність.

Сценарії варіантів використання (Use Case Scenarios)

Сценарій варіанту використання описує конкретну послідовність дій між актором і системою для досягнення певної цілі.

- **Основні компоненти:**
 - Назва сценарію.
 - Опис: короткий огляд того, що відбувається.
 - Основний потік: стандартна послідовність дій.
 - Альтернативні потоки: відхилення від основного сценарію.
 - Попередні умови: що має бути виконано перед початком сценарію.
 - Результат: стан системи після виконання сценарію.

Діаграми UML (Unified Modeling Language)

UML — це мова моделювання для створення візуальних діаграм, які описують різні аспекти системи.

- **Основні типи діаграм:**
 1. **Структурні:** діаграми класів, компонентів, об'єктів, розгортання.

2. **Поведінкові:** діаграми варіантів використання, активності, послідовності, станів.

3. **Інтеракційні:** діаграми комунікації, часові діаграми.

- **Призначення:** полегшення аналізу, дизайну та документування системи.

Діаграма класів (Class Diagram)

Діаграма класів моделює структуру системи, відображаючи класи, їх атрибути, методи та взаємозв'язки.

- **Елементи:**
 - **Класи:** описують об'єкти (атрибути та методи).
 - **Зв'язки:**
 - Асоціація (association).
 - Агрегація (aggregation).
 - Композиція (composition).
 - Наслідування (inheritance).
 - **Мультиплікатори:** показують кількість об'єктів у зв'язку.
- **Призначення:** деталізує статичну структуру системи.

Концептуальна модель системи

Концептуальна модель системи — це абстрактне уявлення про основні об'єкти домену і зв'язки між ними.

- **Компоненти:**
 - Об'єкти/класи: реальні чи абстрактні сутності, що мають атрибути та операції.
 - Атрибути: властивості об'єктів.
 - Зв'язки: взаємодії між об'єктами.
- **Призначення:** допомагає зрозуміти бізнес-логіку та основні взаємозв'язки в системі.
- **Застосування:** використовується на початкових етапах проєктування для формування загального уявлення про систему.

Хід роботи

Система для колективних покупок (proxy, builder, decorator, facade, composite). Система дозволяє створити список групи для колективної покупки, список що потрібно купити з орієнтовною вартістю кожної позиції та орієнтовною загальною вартістю, запланувати хто що буде купляти. Щоб користувач міг відмітити що він купив, за яку суму, з можливістю прикріпити чек. Система дозволяє користувачу вести списки бажаних для нього покупок, з можливістю позначати списки, які будуть доступні для друзів (як списки, що можна подарувати користувачеві). Система дозволяє добавляти інших користувачів в друзі.

Діаграми варіантів використання (Use-case diagram)

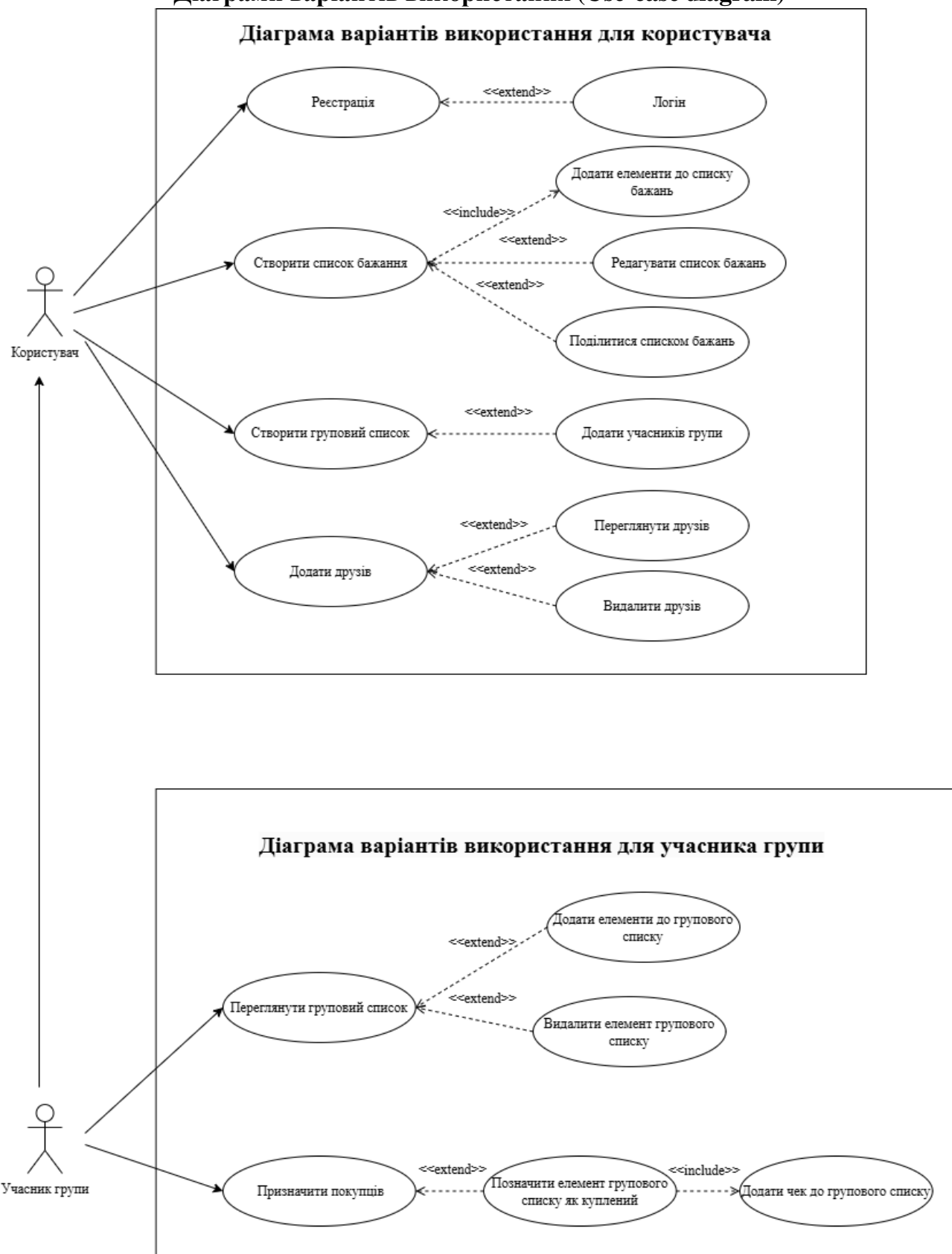


Рис. 1.1 - Use-case diagram

Варіанти використання (прецеденти)

1. Створення списку групи для колективної покупки

Передумови: Користувач авторизований у системі.

Постумови: Список групи для колективної покупки успішно створено.

Взаємодіючі сторони: Користувач, система для колективних покупок.

Короткий опис: Цей варіант описує процес створення списку групи для організації колективної покупки.

Основний потік подій:

1. Користувач входить у систему для колективних покупок.
2. Користувач натискає кнопку "Створити список групи".
3. Користувач заповнює назву групи, опис та додає учасників.
4. Користувач натискає кнопку "Зберегти". Система зберігає дані про групу та підтверджує успішне створення.
5. Користувач отримує сповіщення про успішне створення списку групи.

Винятки:

- **Виняток №1:** Пропущені обов'язкові поля (назва групи, учасники). Система відображає повідомлення про помилку та пропонує виправити.

2. Додавання списку покупок для колективної покупки

Передумови: Користувач створив або приєднався до групи.

Постумови: Список покупок успішно додано.

Взаємодіючі сторони: Користувач, система для колективних покупок.

Короткий опис: Цей варіант описує процес створення списку покупок з орієнтовними цінами.

Основний потік подій:

1. Користувач відкриває групу в системі для колективних покупок.
2. Користувач натискає кнопку "Додати список покупок".
3. Користувач вводить назви товарів, орієнтовні ціни та кількість кожної позиції.

4. Користувач натискає кнопку "Зберегти". Система обчислює загальну орієнтовну вартість і зберігає список.
5. Користувач отримує підтвердження про успішне створення списку покупок.

Винятки:

- **Виняток №1:** Відсутні дані про товари. Система виводить повідомлення про помилку та пропонує вказати хоча б одну позицію.

3. Відмітка про виконання покупки з додаванням чека

Передумови: Користувач приєднаний до групи і має доступ до списку покупок.

Постумови: Покупка успішно відмічена як виконана, додано чек.

Взаємодіючі сторони: Користувач, система для колективних покупок.

Короткий опис: Цей варіант описує процес відмітки виконаної покупки з додаванням чека.

Основний потік подій:

1. Користувач відкриває список покупок у групі.
2. Користувач обирає позицію, яку купив, і натискає кнопку "Відмітити як виконану".
3. Користувач вказує фактичну вартість товару і завантажує чек (за бажанням).
4. Користувач натискає кнопку "Зберегти". Система оновлює статус позиції у списку.
5. Користувач отримує сповіщення про успішне оновлення.

Винятки:

- **Виняток №1:** Чек не прикріплено (якщо це обов'язково). Система відображає повідомлення із запитом на додавання чека.

UML діаграма класів

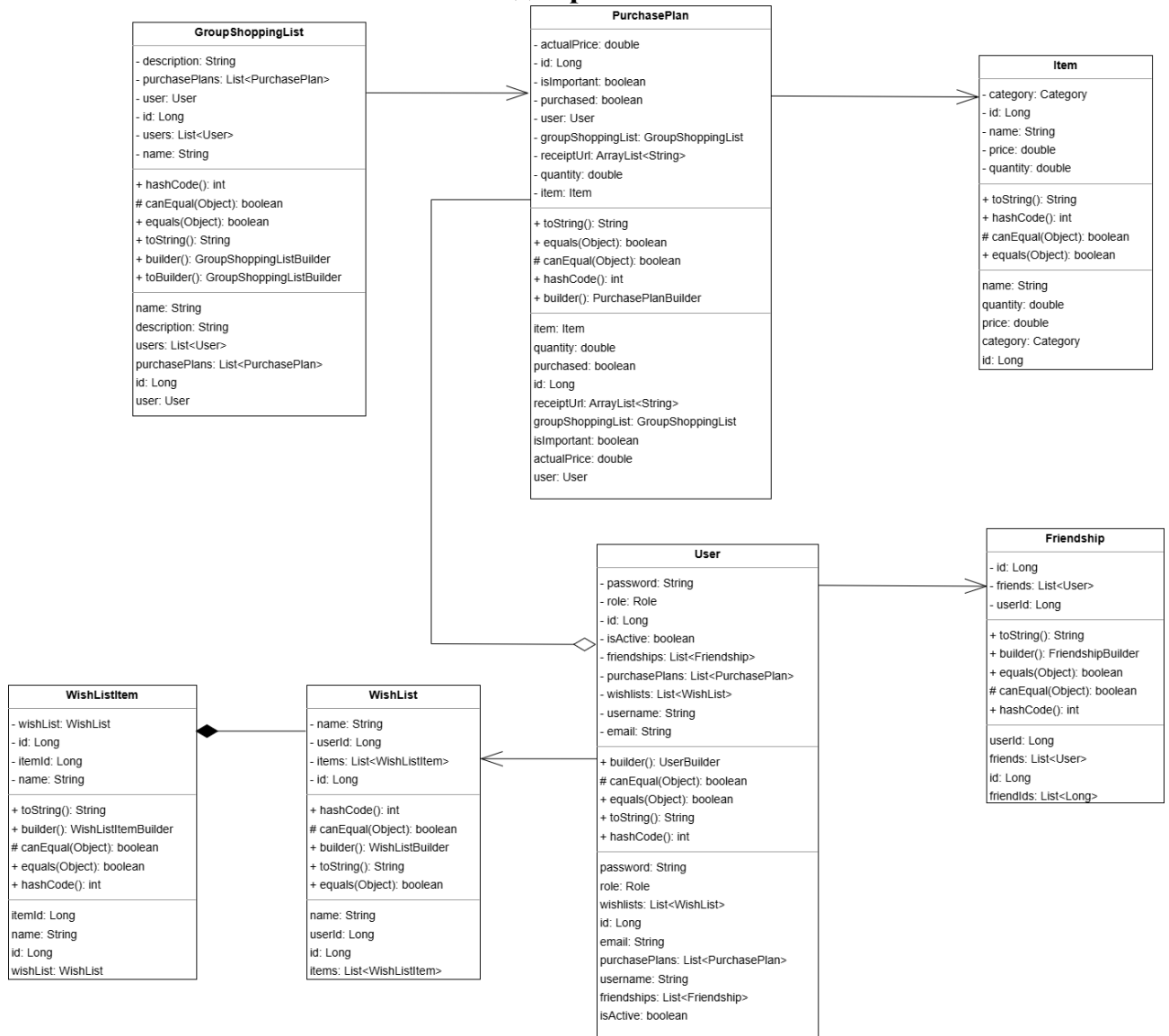


Рис. 2.1 – UML діаграма класів

1. **GroupShoppingList**- представляє груповий список покупок.
2. **PurchasePlan**- представляє план покупки, пов'язаний із певним товаром.
3. **Item**- представляє товар, який можна додати до списку покупок.
4. **WishList**- представляє список бажань користувача.
5. **WishListItem**- представляє конкретний товар у списку бажань.
6. **User**- представляє користувача системи.
7. **Friendship** - описує дружбу між користувачами.

Зображення структури бази даних

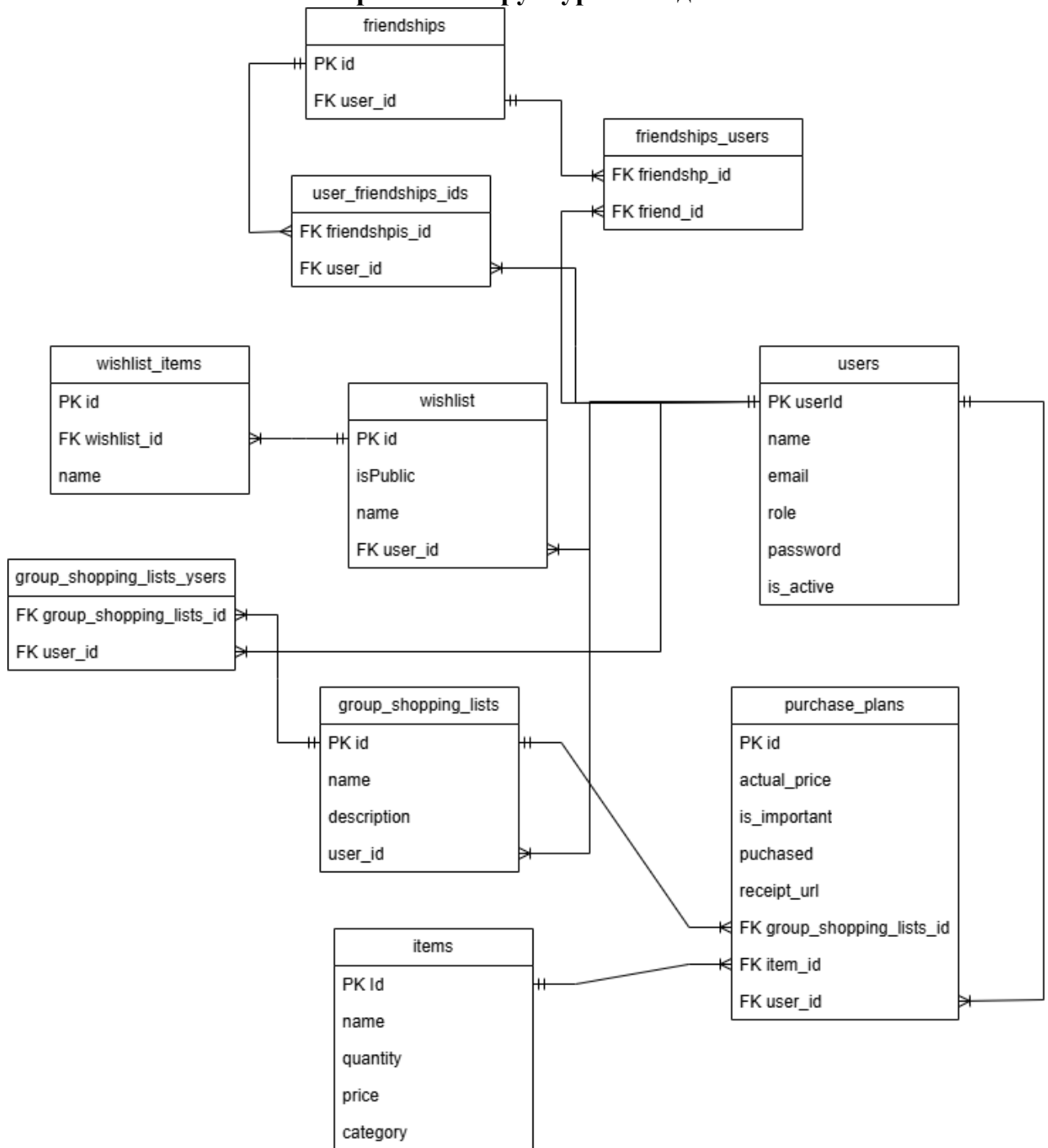


Рис. 3.1 – Структура бази даних

1. Таблиця users

- **Призначення:** Містить інформацію про користувачів системи.
- **Колонки:**
 - userId (PK): Унікальний ідентифікатор користувача.
 - name: Ім'я користувача.

- email: Адреса електронної пошти користувача.
- role: Роль користувача в системі.
- password: Пароль користувача.
- is_active: Статус активності користувача.

2. Таблиця friendships

- **Призначення:** Моделює дружні зв'язки між користувачами.
- **Колонки:**
 - id (PK): Унікальний ідентифікатор дружби.
 - user_id (FK): Ідентифікатор користувача, який ініціював дружбу.
 - Зв'язок із таблицею users.

3. Таблиця friendships_users

- **Призначення:** Використовується для збереження друзів кожного користувача.
- **Колонки:**
 - id (PK): Унікальний ідентифікатор.
 - friend_id (FK): Ідентифікатор друга.
 - Зв'язок із таблицею users.

4. Таблиця wishlist

- **Призначення:** Зберігає списки бажань користувачів.
- **Колонки:**
 - id (PK): Унікальний ідентифікатор списку бажань.
 - isPublic: Прапорець, чи є список публічним.
 - name: Назва списку бажань.
 - user_id (FK): Ідентифікатор власника списку.
 - Зв'язок із таблицею users.

5. Таблиця wishlist_items

- **Призначення:** Містить позиції з бажань користувачів.
- **Колонки:**
 - wishlist_items_id (PK): Унікальний ідентифікатор позиції.
 - wishlist_id (FK): Ідентифікатор списку бажань.
 - name: Назва товару.
 - Зв'язок із таблицею wishlist.

6. Таблиця group_shopping_lists

- **Призначення:** Містить інформацію про списки покупок, створені групами.
- **Колонки:**
 - id (PK): Унікальний ідентифікатор списку.
 - name: Назва списку покупок.
 - description: Опис списку.
 - user_id (FK): Ідентифікатор користувача, який створив список.
 - Зв'язок із таблицею users.

7. Таблиця group_shopping_lists_users

- **Призначення:** Зберігає дані про користувачів, що належать до певного списку покупок.
- **Колонки:**
 - group_shopping_lists_id (FK): Ідентифікатор списку.
 - user_id (FK): Ідентифікатор користувача.
 - Зв'язок із таблицями group_shopping_lists і users.

8. Таблиця items

- **Призначення:** Містить інформацію про товари.
- **Колонки:**
 - id (PK): Унікальний ідентифікатор товару.

- name: Назва товару.
- quantity: Кількість товару.
- price: Орієнтовна ціна товару.
- category: Категорія товару.

9. Таблиця purchase_plans

- **Призначення:** Зберігає дані про плани покупок.
- **Колонки:**
 - id (PK): Унікальний ідентифікатор плану.
 - actual_price: Фактична ціна товару.
 - is_important: Чи є покупка важливою.
 - purchased: Статус виконання покупки.
 - receipt_url: URL чека.
 - group_shopping_lists_id (FK): Ідентифікатор списку покупок.
 - item_id (FK): Ідентифікатор товару.
 - user_id (FK): Ідентифікатор користувача, який відповідає за покупку.
 - Зв'язок із таблицями group_shopping_lists, items і users.

Посилання на репозиторій:

<https://github.com/annaorlovskaaa/collectivePurchases.git>

Висновок: у ході виконання даної лабораторної роботи, було розроблено діаграми використання, класів, а також схему бази даних.