МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ

«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ   
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ   
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

Затверджую:

Голова циклової комісії

\_\_\_\_\_\_\_\_\_ / Олександр БАБИЧ

“\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2023 р.

**КУРСОВИЙ ПРОЄКТ**

з дисципліни: “Проєктування автоматизованих інформаційних систем”

Розробка автоматизованої інформаційної системи життєвих досягнень

на тему \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КП 014.40/2021.045**

Виконав: здобувач освіти 4 курсу,

45

групи \_\_\_\_\_\_\_  
спеціальність 121 «Інженерія програмного забезпечення»

спеціалізація «Розробка програмного забезпечення»

Олег ЛЮБЧЕНКО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (прізвище та ініціали)

Світлана ГРИЦЕНКО

Керівник \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 (підпис) (прізвище та ініціали)

Полтава – 2024

**ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ПЛАН**

**ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ПЛАН**

ЗМІСТ

[ВСТУП 5](#_Toc182312819)

[1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ 6](#_Toc182312820)

[1.1. Постановка задачі 6](#_Toc182312821)

[1.2. Розробка прототипу інтерфейсу 6](#_Toc182312822)

[1.3. Опис та побудова діаграми прецедентів роботи із системою 13](#_Toc182312823)

[1.4. Інфологічне проєктування – схема бази даних 15](#_Toc182312824)

[2. ОПИС ПРОГРАМНИХ ЗАСОБІВ 17](#_Toc182312825)

[2.1. Мова програмування 17](#_Toc182312826)

[2.2. Бот API 17](#_Toc182312827)

[2.3. База даних 17](#_Toc182312828)

[2.4. Середовище розробки 18](#_Toc182312829)

[3. РЕАЛІЗАЦІЯ РОЗВ’ЯЗАННЯ ЗАДАЧІ ПРОЕКТУВАННЯ 19](#_Toc182312830)

[3.1. Алгоритм програми, архітектура програми 19](#_Toc182312831)

[3.2. Виклик та завантаження 21](#_Toc182312832)

[3.3. Інструкція користувачу 21](#_Toc182312833)

[ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ 30](#_Toc182312834)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 32](#_Toc182312835)

[ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ 33](#_Toc182312836)

[ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ 37](#_Toc182312837)

# ВСТУП

*Актуальність роботи:* у сучасному світі цифрових технологій і швидкого обміну інформацією, зростає потреба в автоматизації різних аспектів повсякденного життя, включаючи фіксацію та збереження важливих життєвих досягнень. Ця потреба охоплює як особисті досягнення, так і професійні успіхи, оскільки їх облік і перегляд можуть мотивувати до подальшого саморозвитку та підвищення продуктивності. Розробка автоматизованих інформаційних систем, які дозволяють користувачам фіксувати й відслідковувати досягнення, стає актуальною задачею, особливо у контексті месенджерів, які стали невід’ємною частиною повсякденного спілкування.

*Метою* роботи є створення Telegram-бота для автоматизованої інформаційної системи життєвих досягнень. Розроблений бот дозволяє користувачам фіксувати виконання досягнень, додаючи підтвердження у вигляді тексту. За допомогою цього бота користувачі можуть створювати архів своїх досягнень, зберігати спогади та переглядати свій прогрес, що сприятиме формуванню почуття успіху та впевненості. [3]

*Завдання* роботи полягає у розробці та реалізації програмного забезпечення, яке дозволить користувачам зручно зберігати інформацію про досягнення, а також аналізувати та переглядати їх за допомогою функціонала бота. Вирішення поставленого завдання досягається через проектування і реалізацію відповідної структури бази даних. Бот також забезпечує функції пошуку та статистики.

*Об’єктом дослідження* виступає технологія розробки Telegram ботів.

*Предмети дослідження* включають використання мови програмування Python та застосування Telegram-бот API для взаємодії з користувачами. Крім того, зберігання даних організоване через базу даних Firebase Firestore, що дозволяє забезпечити надійність і зручність доступу до інформації про досягнення.

# ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ

## 1.1. Постановка задачі

Задача – створити автоматизовану інформаційну систему для фіксації життєвих досягнень. Розроблювана система має бути Telegram-ботом, основне призначення якого – зберігання інформації про перебіг виконання досягнень, з можливістю їх подальшого перегляду та аналізу.

Потенційними користувачами системи є особи, які бажають текстово фіксувати виконання досягнень та вести особистий архів.

Основна функціональність, яку пропонуватиме система, включає:

* фіксацію текстового опису про виконання досягнення;
* можливість демонстрації виконання друзям;
* перегляд власної та глобальної статистик;
* редагування псевдоніму в межах системи;
* зміна приватності профілю;
* можливість пропонувати досягнення адміністрації.

Для досягнення цього функціоналу система повинна забезпечити стабільне зберігання даних, а також можливість швидкого доступу до них за запитом користувача. У якості бази даних було обрано Firebase Firestore, що забезпечує гнучкість і масштабованість, а також надійну роботу з великими обсягами інформації.

## 1.2. Розробка прототипу інтерфейсу

На етапі проектування інтерфейсу враховано специфіку використання Telegram-бота, де основним інструментом є текстові команди й кнопки для навігації. Прототип інтерфейсу передбачає:

* повідомлення головного меню з описом системи та кнопками для пошуку досягнень, та перегляду статистики;
* заготовлений список слеш-команд, що передбачений інтерфейсом месенджера;
* інші повідомлення з елементами навігації та HTML форматуванням для доступу до інших функцій.

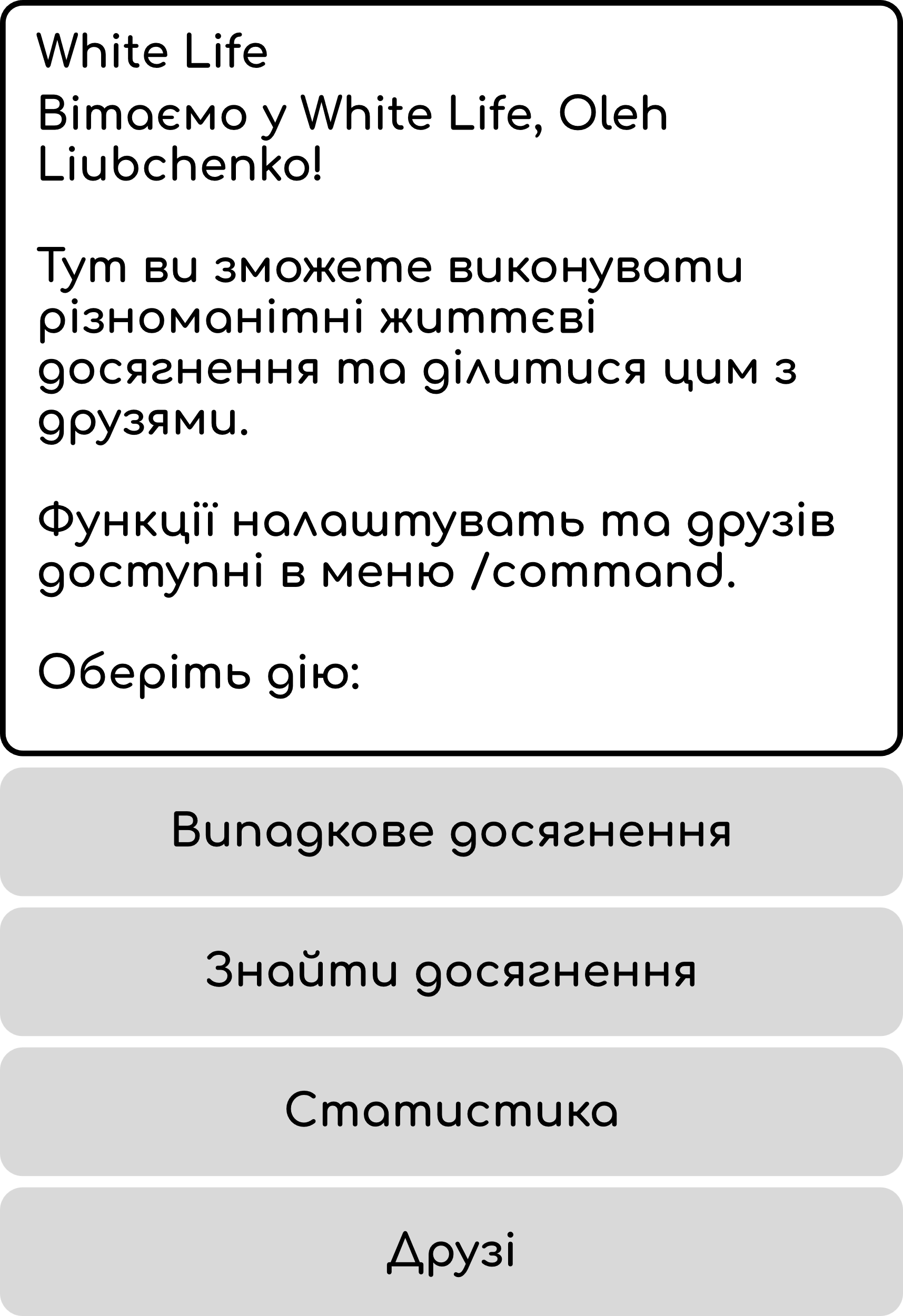
Для розробки прототипу інтерфейсу було використано програму Figma.

Figma – це потужний інструмент для розробки інтерфейсів та створення інтерактивних прототипів, який використовують дизайнери, продакт-менеджери та розробники. Завдяки тому, що Figma працює як в браузері, так і в настільному режимі, вона є доступною на різних платформах, що полегшує роботу над проєктами.

Для створення прототипу інтерфейсу в Figma користувачі можуть розробляти інтуїтивні дизайни за допомогою простих у використанні інструментів для побудови графічних елементів. Функції, такі як автокомпонування та створення компонентів, дозволяють створювати масштабовані та структуровані елементи, які можна багаторазово використовувати, прискорюючи процес розробки та підтримуючи цілісність дизайну.

Figma також пропонує можливість інтерактивного прототипування – створення функціональних макетів, що імітують поведінку реального додатку. Завдяки цьому є можливим тестування різних сценаріїв користувацької взаємодії, що спрощує оцінку юзабіліті. Інструмент підтримує колаборацію в реальному часі, тож усі зміни одразу відображаються у вікні перегляду.

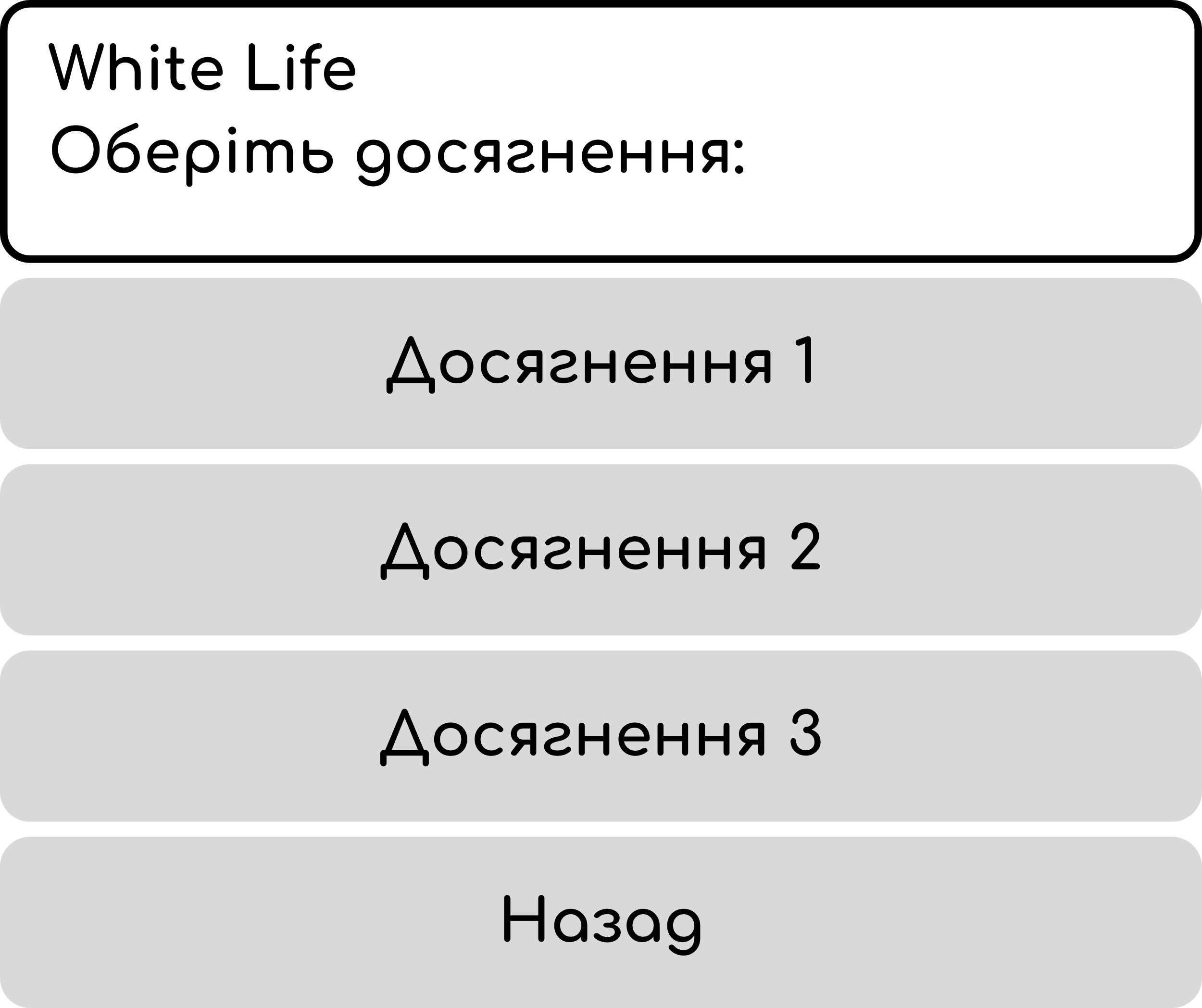
Користувачів зустрічатиме повідомлення головного меню з вітанням, базовим поясненням функціоналу та кнопками взаємодії (Рисунок 1.1). [1]

  
Рисунок 1.1 – Повідомлення головного меню

Далі користувач зможе знайти собі досягнення для виконання, скористувавшись функцією випадкового підбору (Рисунок 1.2), або відкрити повний список доступних досягнень (Рисунок 1.3).

В меню випадкового досягнення також можна буде повторно виконати пошук, перейти до виконання, чи повернутися до головного меню.

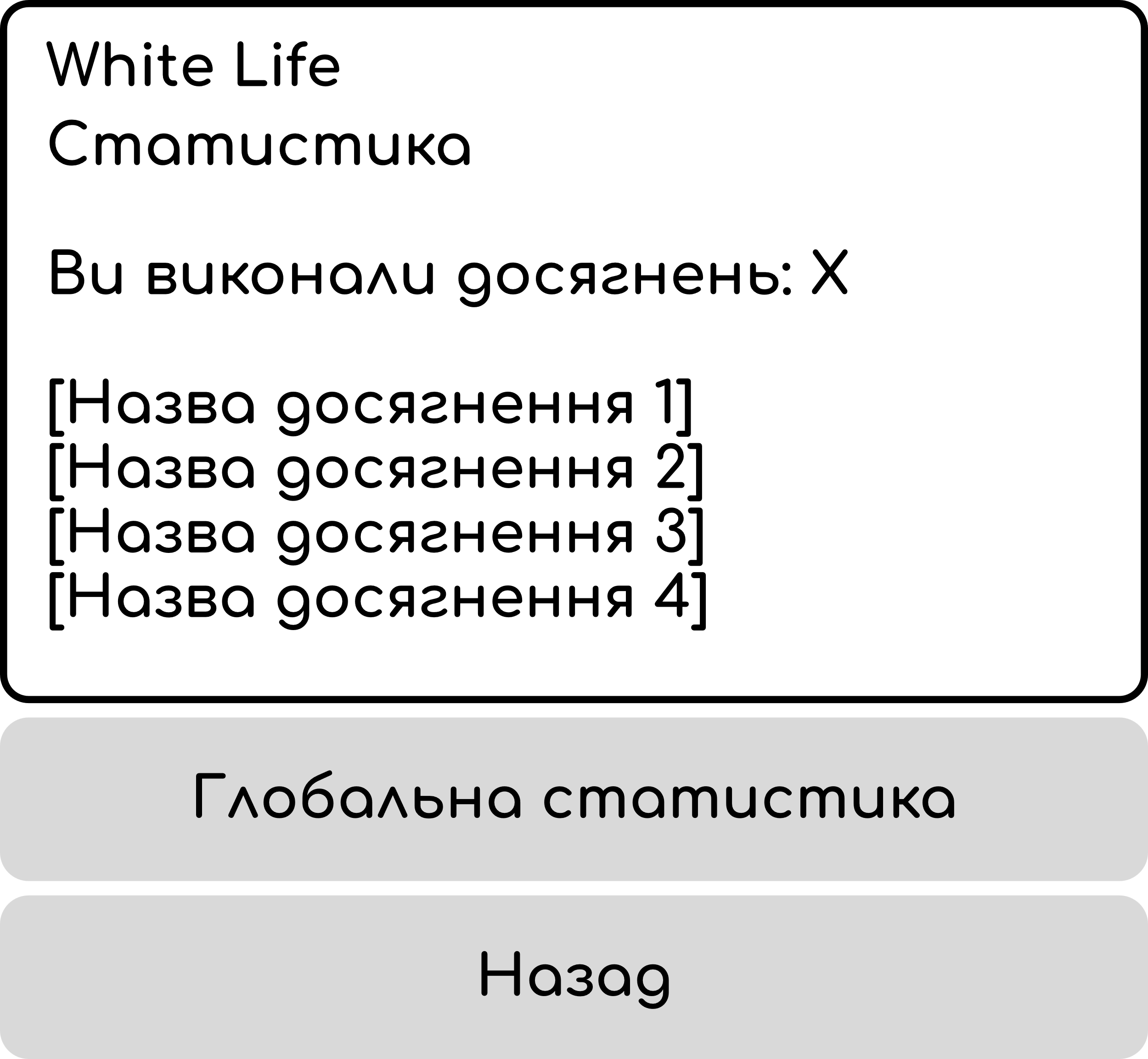
  
Рисунок 1.2 – Деталі досягнення, отриманого методом випадкового пошуку

  
Рисунок 1.3 – Список досягнень в методі ручного пошуку

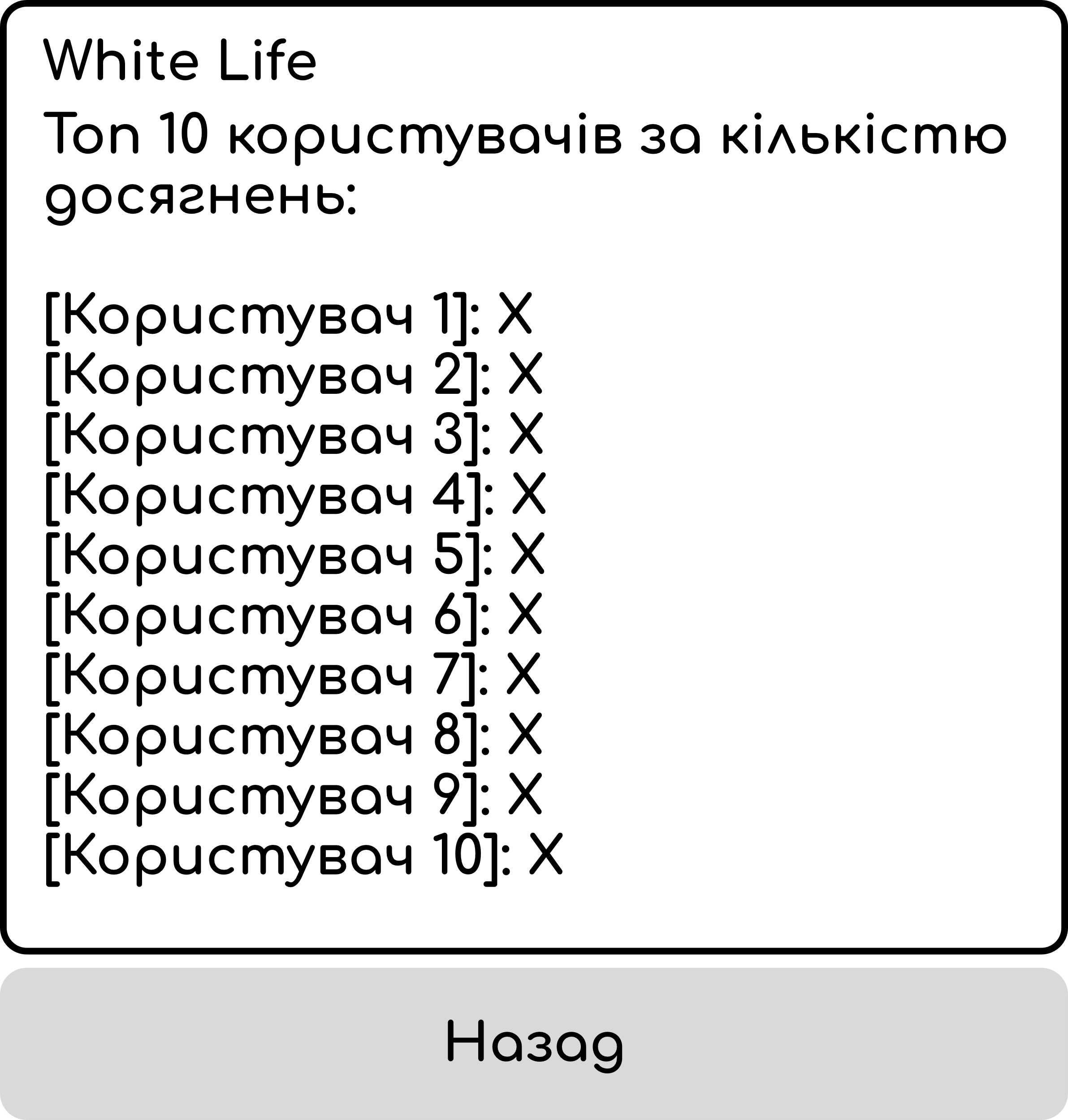
При ручному пошуку, використанні повного списку досягнень, користувач зможе відкрити деталі досягнення (Рисунок 1.4) та виконати його, або ж повернутися назад.

  
Рисунок 1.4 – Деталі досягнення, отриманого методом ручного пошуку

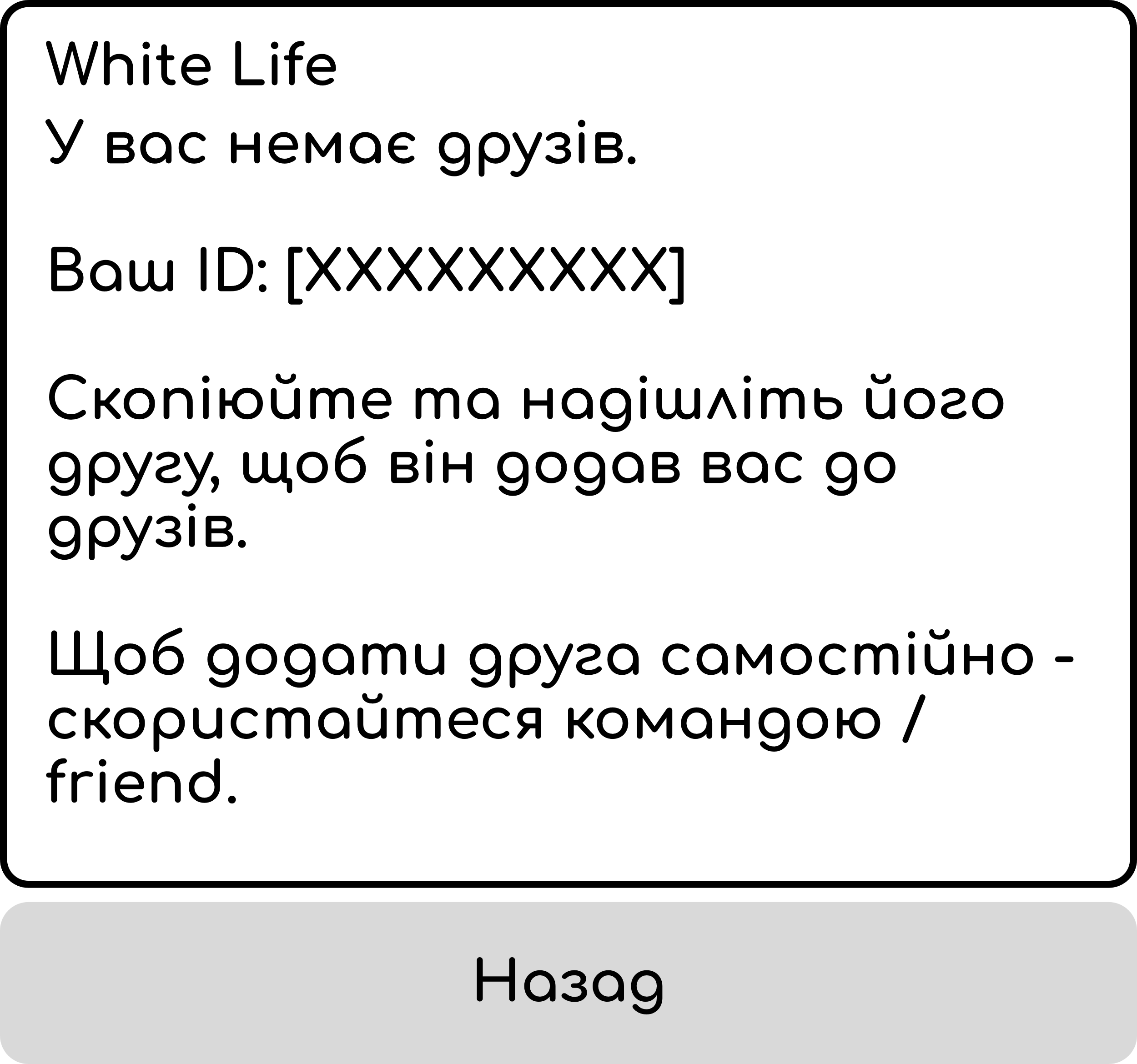
Якщо користувач забажає переглянути власну статистику, то його зустріне повідомлення з кількістю виконаних досягнень, їхнім списком, а також кнопками навігації для перегляду глобальної статистики, чи повернення до головного меню (Рисунок 1.5).

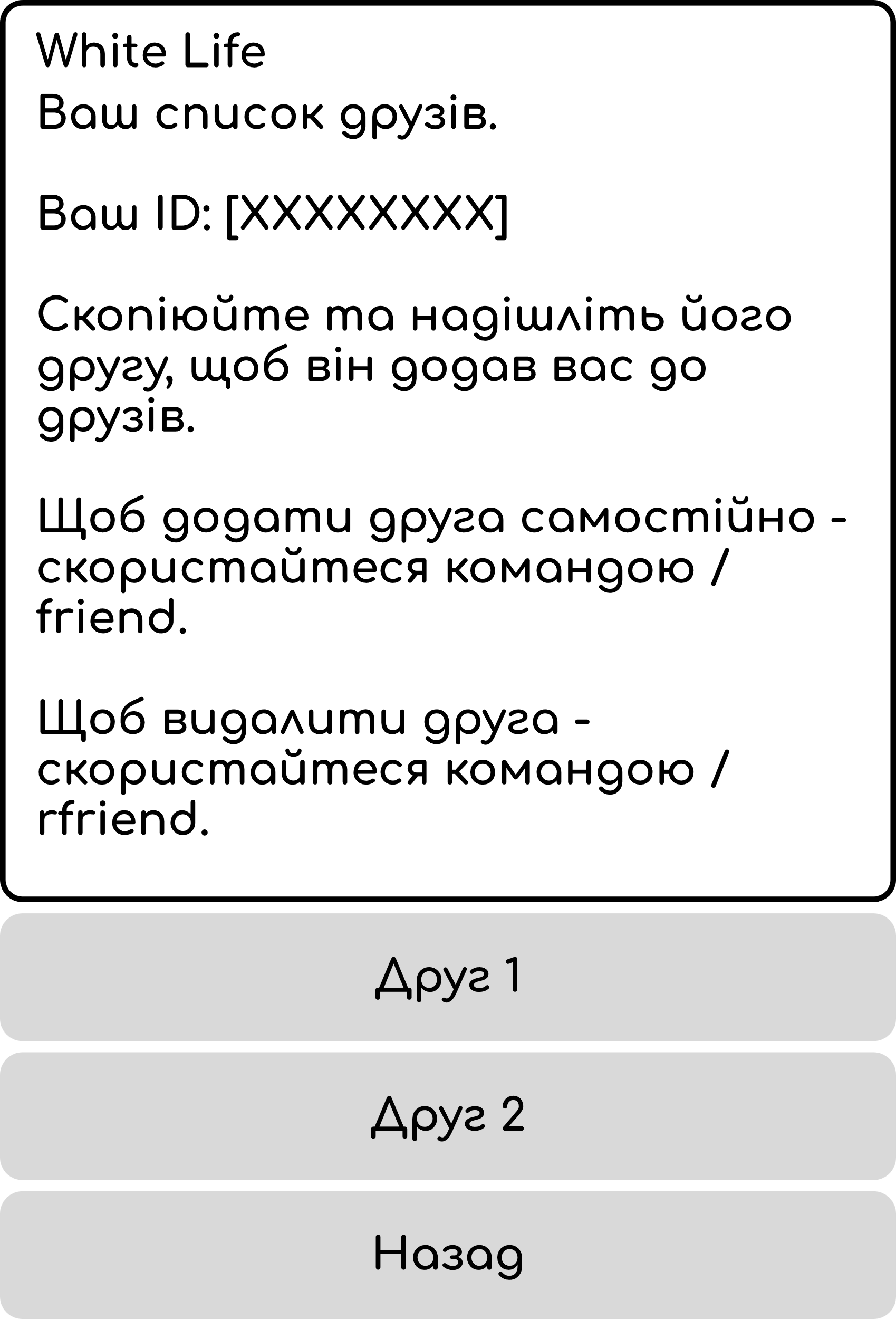
  
Рисунок 1.5 – Статистика користувача

Перегляд глобальної статистики передбачає собою повідомлення зі списком 10-ти користувачів, які будуть відсортовані за кількістю виконаних досягнень (Рисунок 1.6).

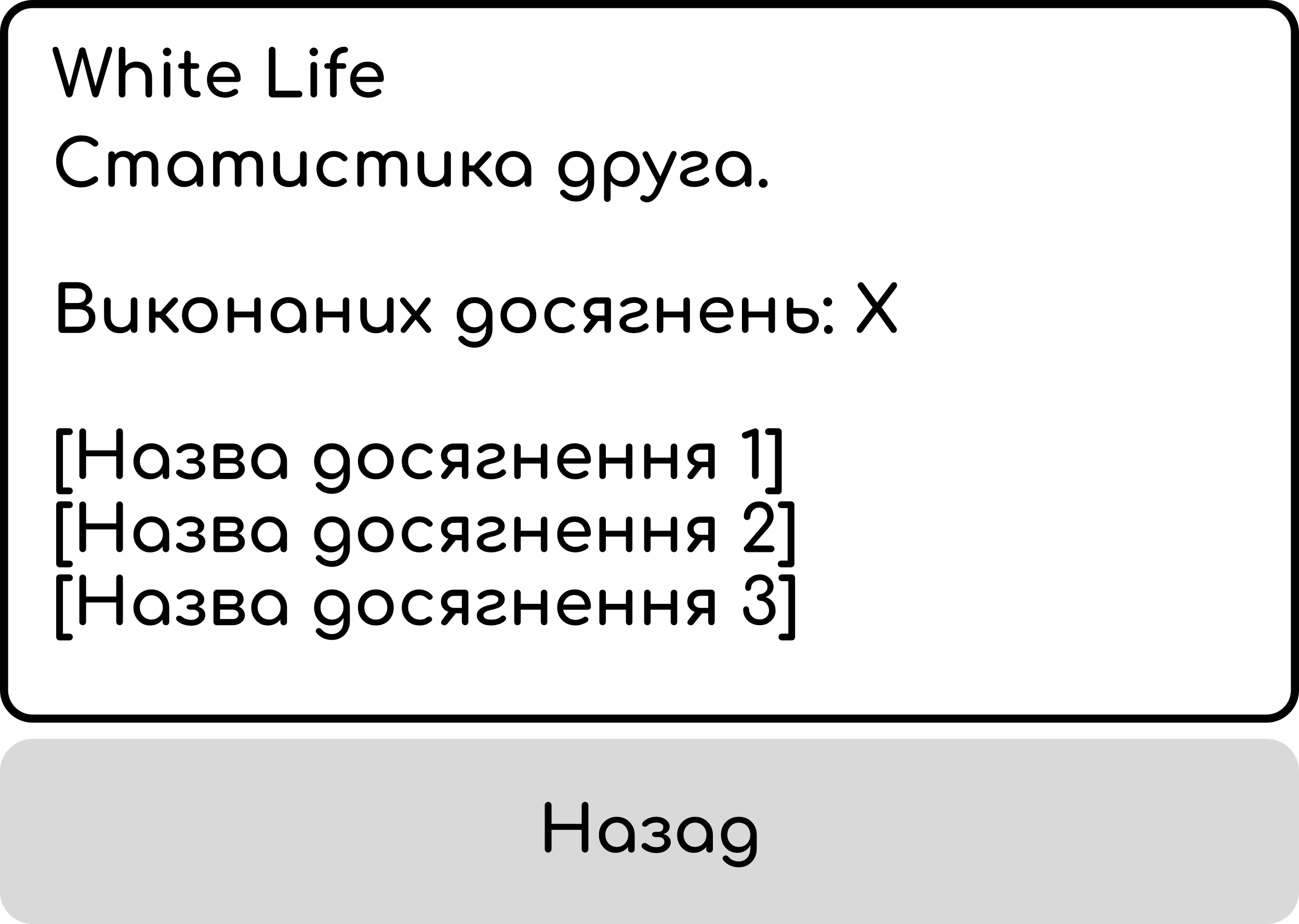
  
Рисунок 1.6 – Глобальна статистика

У програмі буде присутня система друзів, в меню якої можна перейти за допомогою кнопки в головному. Наповнення повідомлення цього меню змінюється в залежності від наявності у користувача друзів (Рисунок 1.7 та Рисунок 1.8). Тут буде відображатися ідентифікатор користувача, який дозволить іншим надіслати дружній запит, підказки щодо команд на цю тему та список друзів у вигляді кнопок.

  
Рисунок 1.7 – Меню друзів у випадку їхньої відсутності

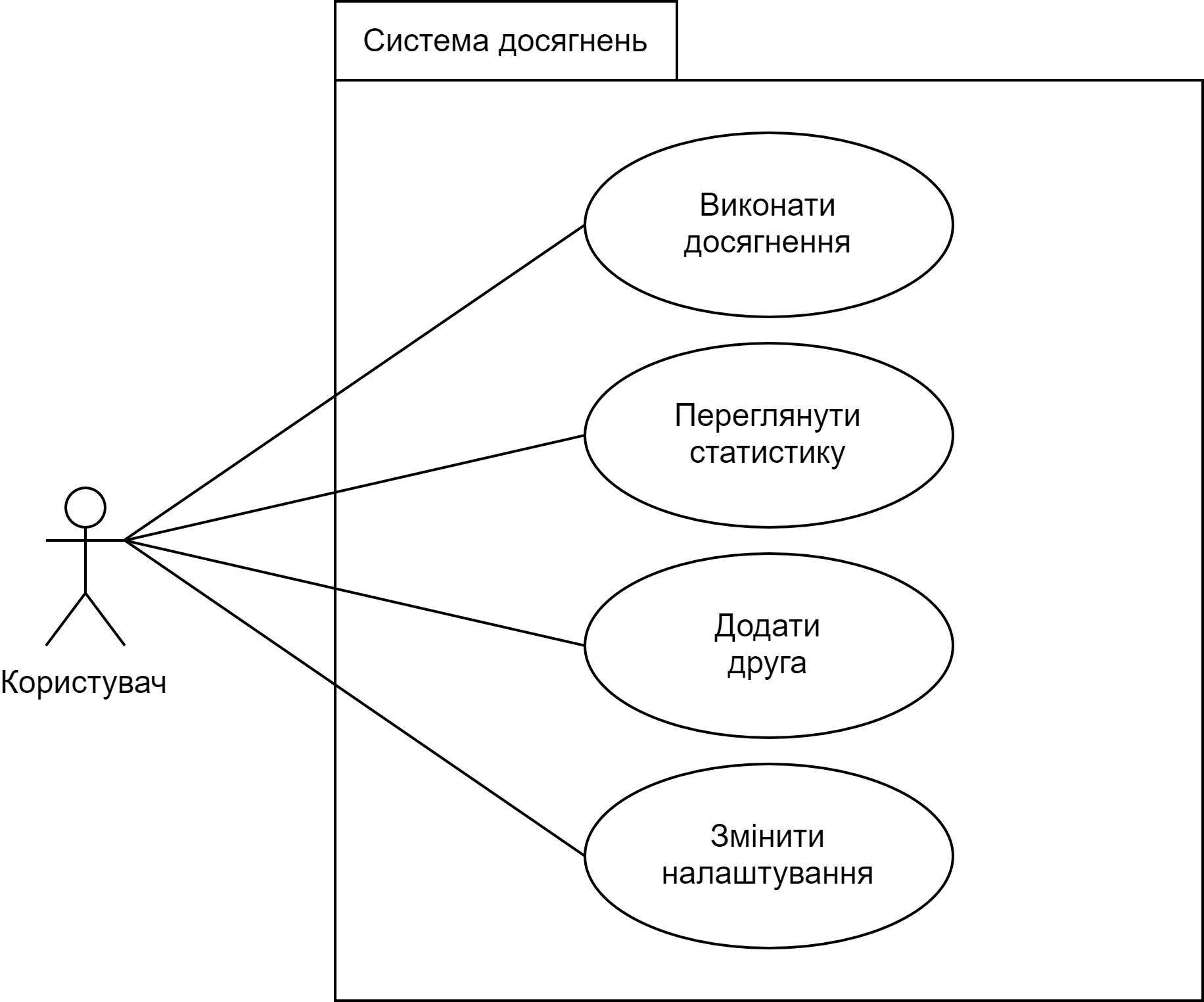
  
Рисунок 1.8 – Меню друзів у випадку їхньої наявності

При натисканні користувачем кнопки з іменем друга, перший зможе переглянути статистику останнього (Рисунок 1.9).

  
Рисунок 1.9 – Статистика друга

## 1.3. Опис та побудова діаграми прецедентів роботи із системою

Діаграму прецедентів роботи із системою зображено схемою (Рисунок 1.10).

  
Рисунок 1.10 – Діаграма прецедентів роботи із системою

ПРЕЦЕДЕНТ: ВИКОНАННЯ ДОСЯГНЕННЯ

Ектор: користувач.

Передумова: необхідність додати інформацію про виконання досягнення.

Післяумова: додано текстову інформацію про перебіг виконання досягнення.

Сценарій:

1. Знайти досягнення методом випадкового підбору, чи вручну.

2. Передати текст про перебіг виконання досягнення.

3. Інформація про виконання збереглася.

ПРЕЦЕДЕНТ: ПЕРЕГЛЯД СТАТИСТИКИ

Ектор: користувач.

Передумова: необхідність переглянути власну та глобальну статистики.

Післяумова: отримано інформацію власної та глобальної статистик.

Сценарій:

1. Перейти до меню статистики.
2. Отримати персональну статистику.
3. Перейти далі до меню глобальної статистики.
4. Отримати глобальну статистику.

ПРЕЦЕДЕНТ: ДОДАВАННЯ ДРУГА

Ектор: користувач.

Передумова: необхідність додати друга в системі.

Післяумова: додано друга.

Сценарій:

1. Отримати від друга його ідентифікатор.
2. Використати команду /friend, вказавши ідентифікатор.
3. Дочекатися підтвердження запиту від друга.
4. Перейти до меню друзів.
5. Побачити доданого друга в списку.

ПРЕЦЕДЕНТ: ЗМІНА НАЛАШТУВАНЬ

Ектор: користувач.

Передумова: необхідність змінити системний псевдонім та статус приватності.

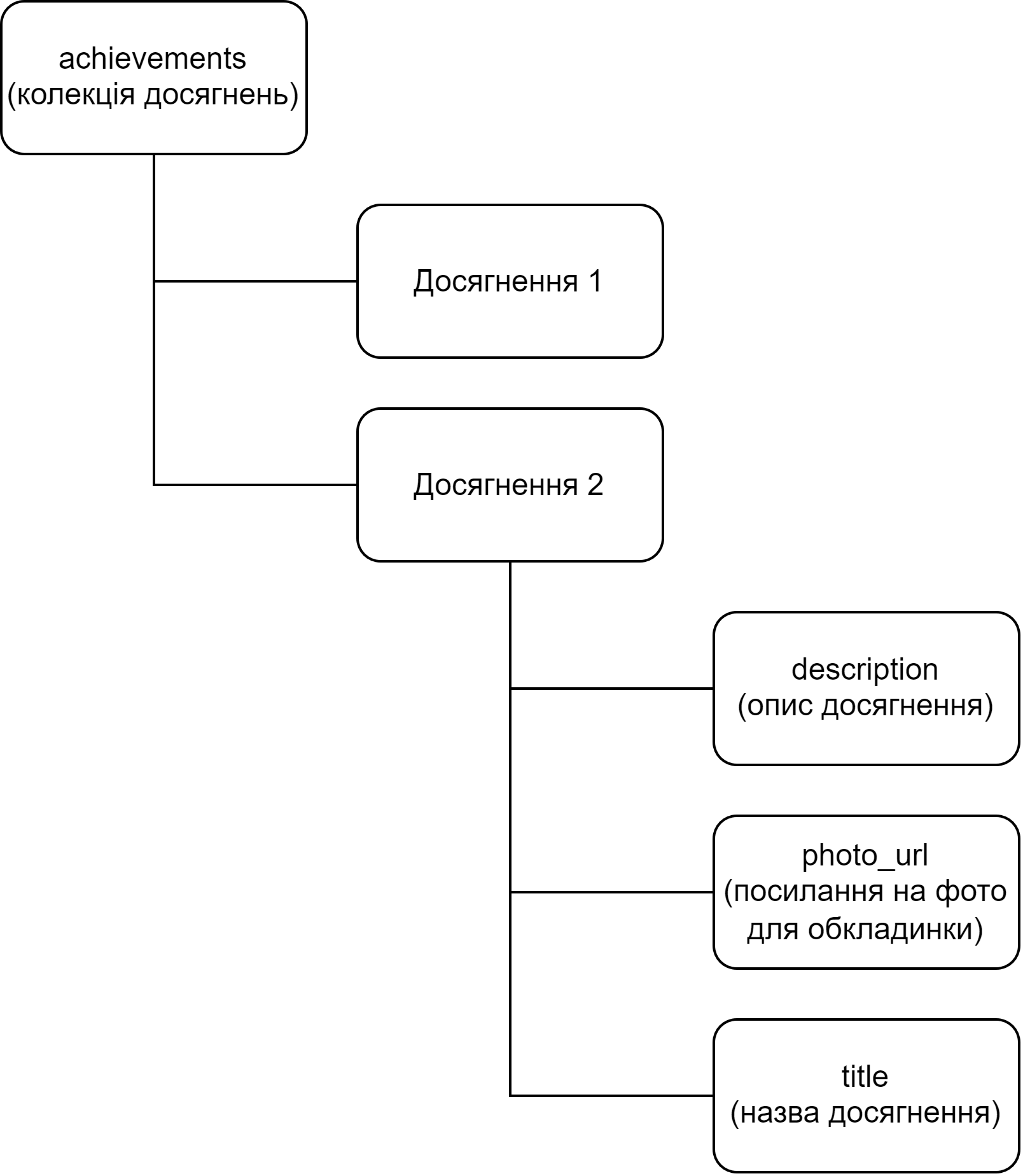
Післяумова: оновлено системний псевдонім та змінено статус приватності на протилежний.

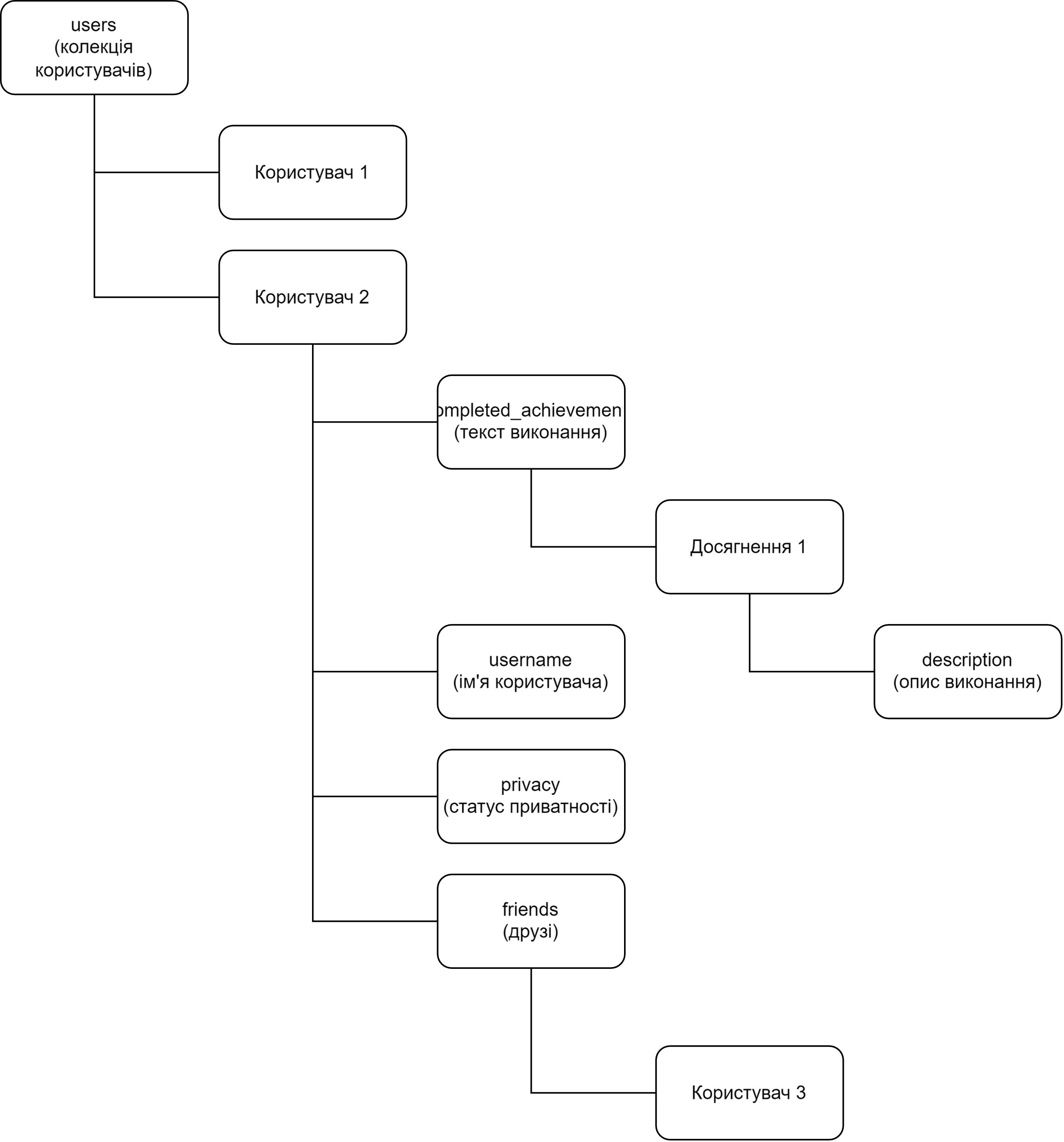
Сценарій:

1. Використати команду /username, вказавши новий псевдонім.
2. Отримати підтвердження про зміну.
3. Використати команду /privacy.
4. Отримати підтвердження про зміну статусу приватності на протилежний.

## 1.4. Інфологічне проєктування – схема бази даних

Конструкцію бази даних зображено схемами (Рисунок 1.11 та Рисунок 1.12).

  
Рисунок 1.11 – Схема бази даних, колекція досягнень

  
Рисунок 1.12 – Схема бази даних, колекція користувачів

# 2. ОПИС ПРОГРАМНИХ ЗАСОБІВ

## 2.1. Мова програмування

Python – мова програмування, обрана для розробки Telegram-бота через її високу популярність, зручність та наявність великої кількості бібліотек, що спрощують створення і підтримку ботів. Python має простий синтаксис, що дозволяє швидко та ефективно реалізовувати складні алгоритми обробки даних, а також забезпечує зручну інтеграцію з API Telegram і Firebase. Python використовується для написання всього програмного коду бота.

## 2.2. Бот API

Для роботи з API Telegram використовується бібліотека python-telegram-bot, яка включає модулі telegram і telegram.ext. Ця бібліотека дозволяє зручно інтегрувати Python-код з платформою Telegram, надаючи прості у використанні функції для розробки інтерфейсу бота, обробки команд користувачів і надсилання повідомлень.

Модуль telegram забезпечує доступ до основних об'єктів, таких як Message, User і Chat, що дозволяє реалізувати функції надсилання і отримання інформації. Модуль telegram.ext надає більш розширені можливості, включаючи обробку слеш команд та використання функцій callback для реагування на натискання кнопок. [4][5]

## 2.3. База даних

Firebase Firestore – хмарна база даних від Google, що забезпечує надійне зберігання та швидкий доступ до даних, що особливо важливо для Telegram-ботів, де потрібна оперативна обробка інформації. Firebase Firestore підтримує гнучке зберігання даних у вигляді документів і колекцій, що дозволяє зручно структурувати інформацію про користувачів і їхні досягнення. [2]

## 2.4. Середовище розробки

Visual Studio Code (VS Code) є однією з найпопулярніших і потужних крос-платформових інтегрованих середовищ розробки (IDE). Розроблена компанією Microsoft, ця програма здобула широку популярність серед розробників усіх рівнів навичок та в різних галузях програмування.

Однією з головних переваг Visual Studio Code є його легкість використання та налаштування. Він має інтуїтивний та зручний інтерфейс, що дозволяє розробникам швидко освоїти його функціонал. Крім того, VS Code підтримує велику кількість розширень, які дозволяють налаштувати його під індивідуальні потреби кожного розробника.

VS Code надає підтримку для багатьох мов програмування, включаючи C, C++, Python, JavaScript, HTML, CSS та багато інших. Це робить його універсальним інструментом для розробки різних типів проектів. Крім того, VS Code підтримує інтеграцію з різними системами керування версіями, такими як Git, що спрощує спільну роботу над проектами в команді.

У Visual Studio Code також є багато корисних функцій, які полегшують розробку програм. Наприклад, він має вбудовану підсвітку синтаксису, автодоповнення коду, відладку, пошук та заміну тексту, вбудований термінал та багато інших інструментів, що допомагають розробникам ефективно працювати над своїми проектами.

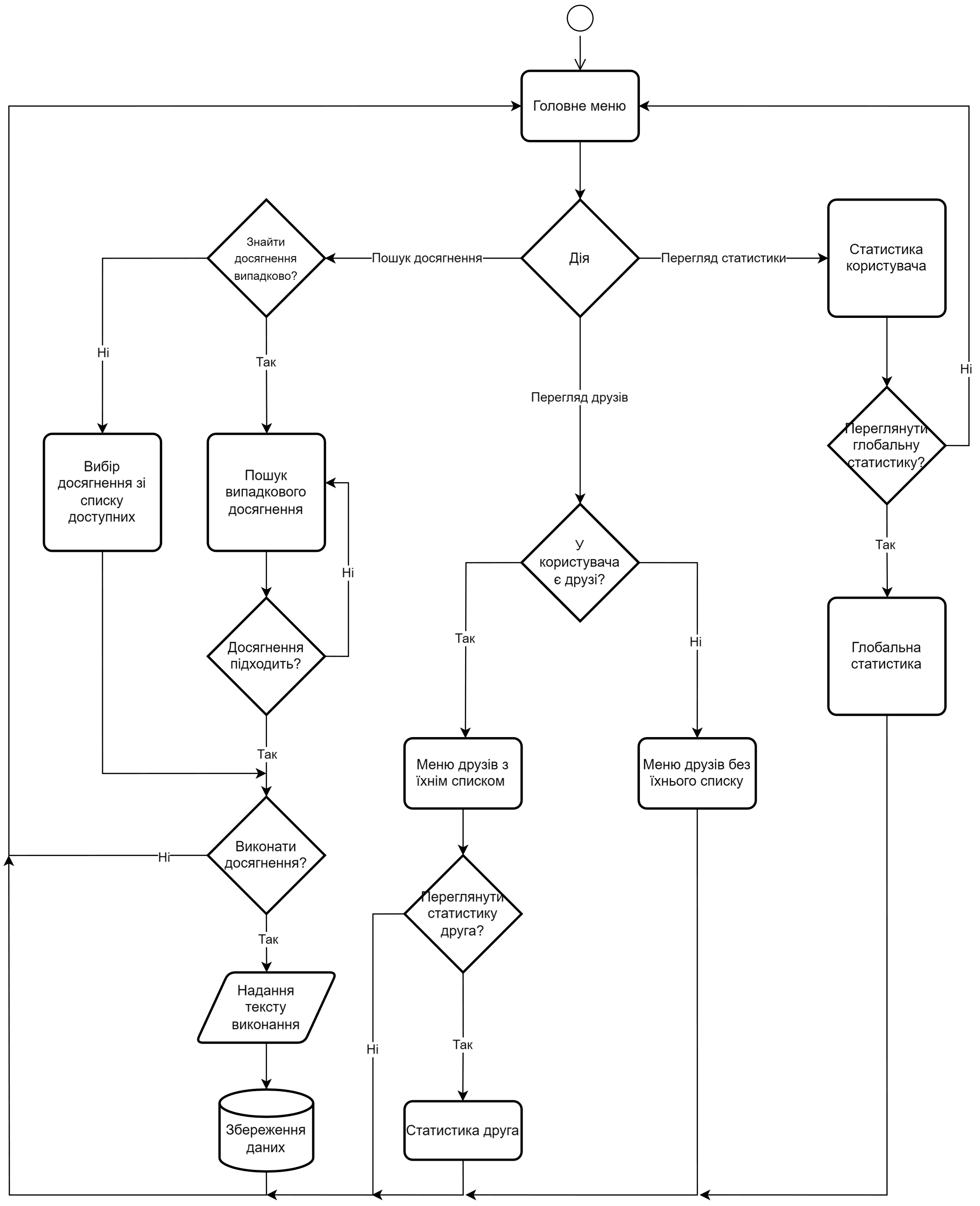
Однією з важливих особливостей Visual Studio Code є його розширюваність. Він базується на архітектурі з плагінами, що дозволяє розробникам створювати власні розширення та теми, розширюючи функціонал програми. Це дозволяє адаптувати VS Code до власних потреб та працювати з різними типами проектів і мов програмування.

Загалом, Visual Studio Code – це потужне та зручне середовище розробки, яке надає розробникам широкий набір інструментів для ефективної роботи над проектами різної складності. Його легкість використання, розширюваність та підтримка багатьох мов програмування роблять його популярним вибором серед програмістів по всьому світу. [6]

# 3. РЕАЛІЗАЦІЯ РОЗВ’ЯЗАННЯ ЗАДАЧІ ПРОЕКТУВАННЯ

## 3.1. Алгоритм програми, архітектура програми

Алгоритм роботи програми, в межах використання кнопок навігації, зображений на рисунку 3.1.

  
Рисунок 3.1 – Алгоритм роботи програми, в межах кнопок навігації

Алгоритм роботи програми, при використанні слеш команд, зображено рисунками 3.2, 3.3, 3.4, 3.5, 3.6.

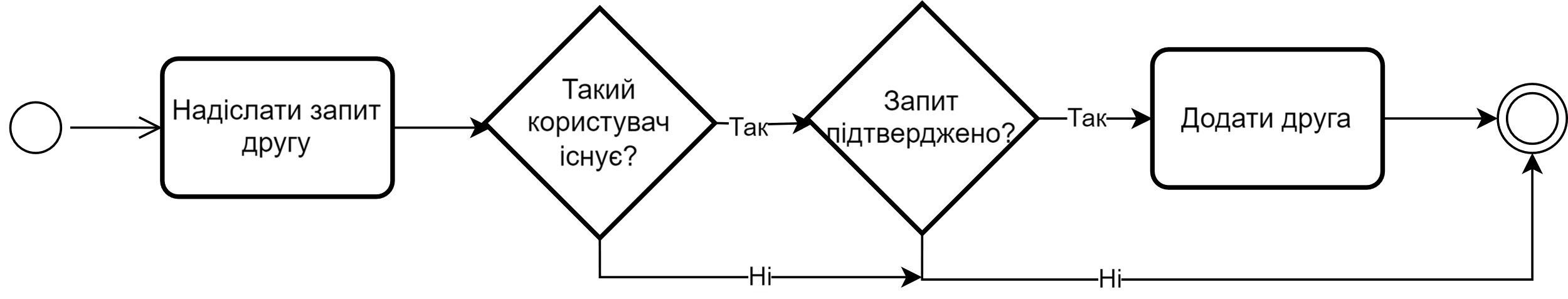
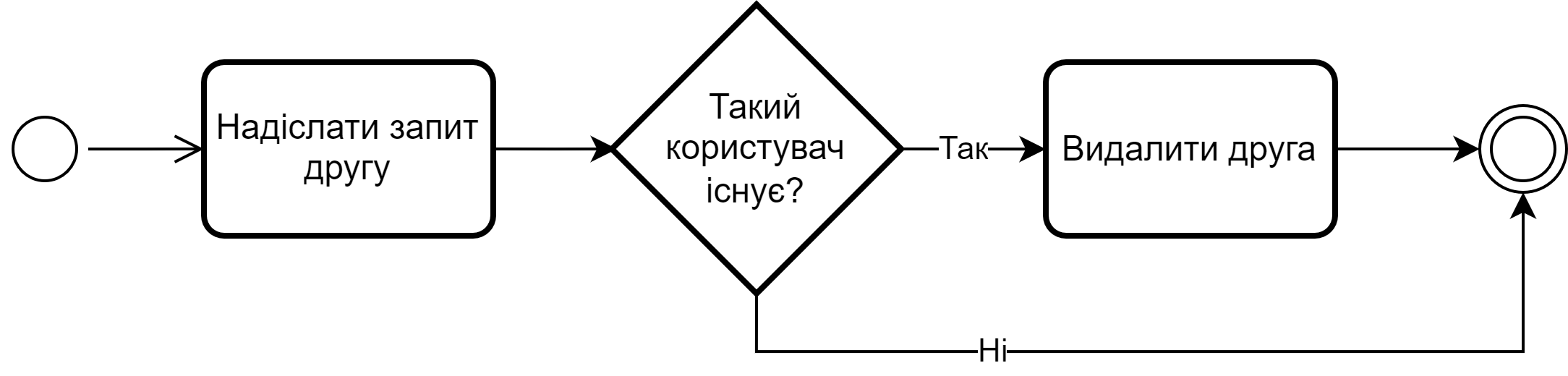
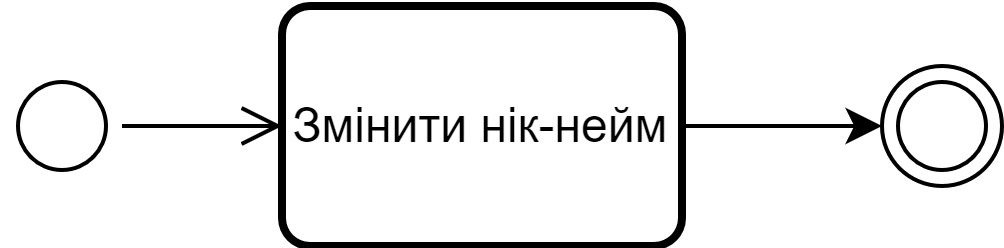
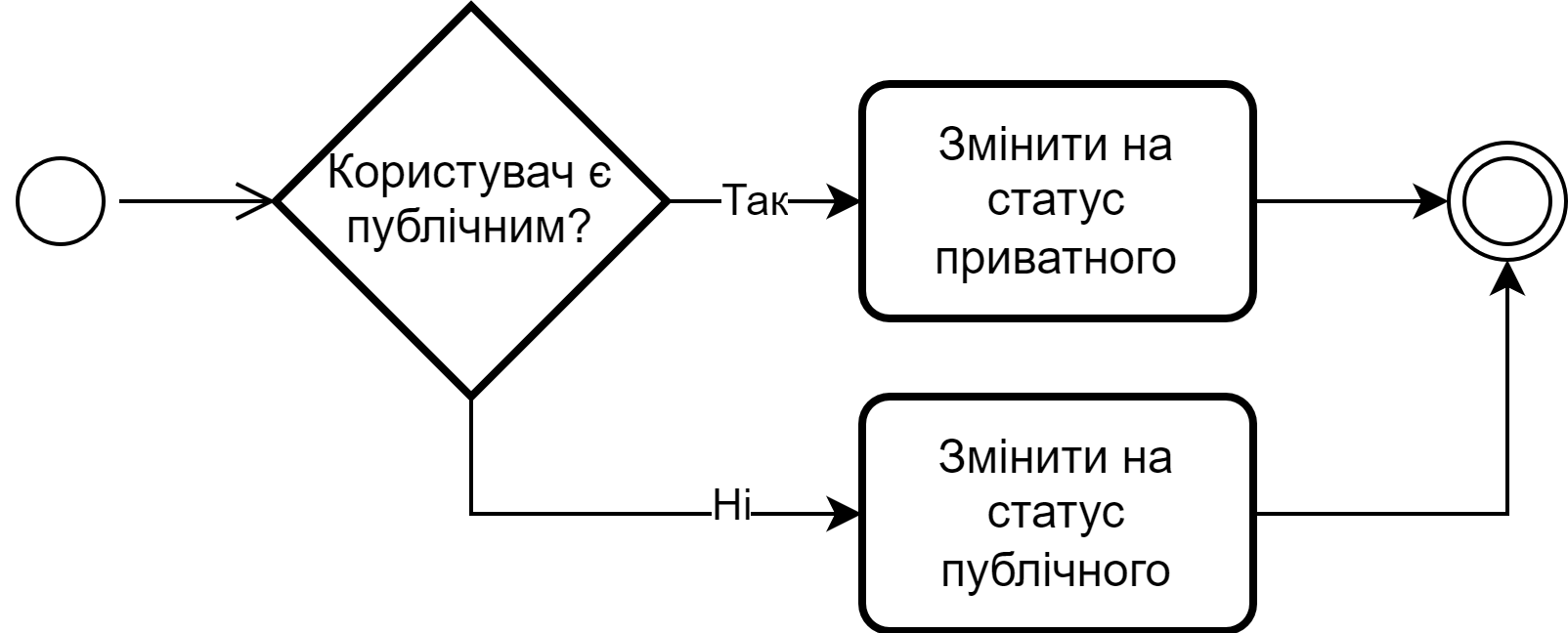
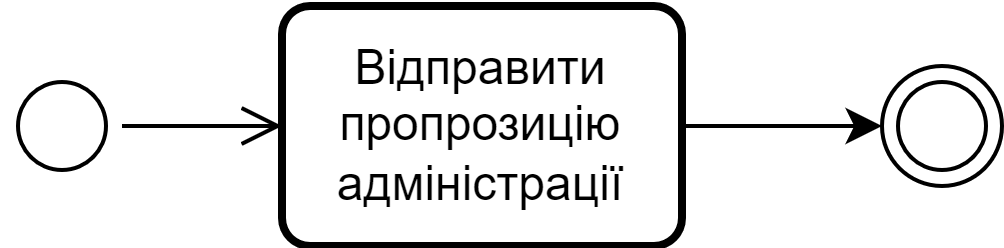


Рисунок 3.2 – Алгоритм роботи програми в межах команди /friend

  
Рисунок 3.3 – Алгоритм роботи програми в межах команди /rfriend

  
Рисунок 3.4 – Алгоритм роботи програми в межах команди /username

  
Рисунок 3.5 – Алгоритм роботи програми в межах команди /privacy

  
Рисунок 3.6 – Алгоритм роботи програми в межах команди /suggest

## 3.2. Виклик та завантаження

Алгоритм виклику програми:

1. Перехід за посиланням https://t.me/whlifebot.
2. Натискання кнопки «РОЗПОЧАТИ» в інтерфейсі месенджера.

## 3.3. Інструкція користувачу

Інструкція по роботі з програмою

1. Виконуємо команду /start, щоб потрапити до головного меню (Рисунок 3.7).

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис  
Рисунок 3.7 – Повідомлення головного меню

1. Виконуємо пошук досягнення за допомогою кнопки «Випадкове досягнення» і отримуємо його (Рисунок 3.8), або ж відкриваємо список всіх доступних досягнень кнопкою «Знайти досягнення» (Рисунок 3.9), після чого обираємо те, яке нас зацікавило, та отримуємо деталі (Рисунок 3.10).

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис  
Рисунок 3.8 – Деталі досягнення, отриманого випадковим способом  
  
Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис  
Рисунок 3.9 – Список доступних досягнень  
  
Зображення, що містить текст, знімок екрана, транспортний засіб

Автоматично згенерований опис  
Рисунок 3.10 – Деталі досягнення, отриманого ручним способом

1. Після отримання деталей досягнення, для його виконання, натискаємо кнопку «Виконати». Після чого бот просить описати процес виконання, відправляємо повідомлення з потрібною інформацією (Рисунок 3.11) та отримуємо деталі досягнення із введеним текстом про виконання (Рисунок 3.12).

Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис  
Рисунок 3.11 – Процес передачі тексту про виконання досягнення  
  
Зображення, що містить текст, знімок екрана

Автоматично згенерований опис  
Рисунок 3.12 – Деталі досягнення із текстом про виконання

1. Для перегляду власної статистики, в головному меню натискаємо кнопку «Статистика», після чого отримуємо її (Рисунок 3.13), тепер також є можливим перегляд глобальної статистики (Рисунок 3.14).

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис  
Рисунок 3.13 – Статистика користувача  
  
Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис  
Рисунок 3.14 – Глобальна статистика

1. Для перегляду власного списку друзів, натискаємо кнопку «Друзі» в головному меню і отримуємо повідомлення про відсутність друзів, якщо користувач не додав їх заздалегідь (Рисунок 3.15), в іншому випадку, отримуємо список з друзями (Рисунок 3.16). Також можливо переглянути статистику друга, натиснувши на кнопку з його іменем (Рисунок 3.17).

Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.15 – Меню друзів у випадку їхньої відсутності

Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис  
Рисунок 3.16 – Меню друзів у випадку їхньої наявності

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис

Рисунок 3.17 – Перегляд статистики друга

1. В цьому повідомленні, під форматом «спойлеру» приховано ідентифікатор користувача, передавши який другу, він буде здатен додати вас до друзів. Щоб додати користувача, виконуємо команду /friend та вказуємо ідентифікатор (Рисунок 3.18). Після чого отримуємо підтвердження про відправку запиту. Тепер інший користувач здатен його прийняти, або відхилити (Рисунок 3.19).

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис  
Рисунок 3.18 – Відправка запиту на додання в друзі  
  
Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис  
Рисунок 3.19 – Отримання запиту на додання в друзі

1. За допомогою інших слеш-команд (Рисунок 3.20) можна виконувати такі дії, як: зміна нік-нейму (системного псевдоніму), зміна статусу приватності профілю, для його приховання з глобальної статистики та пропонувати адміністрації додати важливе, на вашу думку, досягнення. Всі слеш-команди доступні в спеціальному меню інтерфейсу Telegram.

Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.20 – Меню доступних слеш-команд

# ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

Під час виконання курсового проєкту було поставлено завдання розробити автоматизовану інформаційну систему для фіксації та збереження життєвих досягнень користувачів на базі Telegram-бота. Результатом виконаної роботи став повнофункціональний Telegram-бот, який дозволяє користувачам додавати записи про виконання досягнень, зберігаючи їх у вигляді тексту, переглядати архіви та отримувати статистику, що сприяє кращому розумінню власних успіхів.

Усі поставлені завдання були виконані в повному обсязі. У рамках роботи було реалізовано функції додавання перебігу виконання досягнень, зберігання даних у Firebase Firestore, а також розроблено інтерфейс взаємодії користувачів з ботом.

Цей проєкт є актуальним, оскільки в умовах сучасного суспільства зростає потреба в інструментах для ведення персональних записів і мотивуючого самовдосконалення. Практична значимість полягає у зручності зберігання і доступу до важливих досягнень, що можуть служити для саморозвитку та самоствердження користувачів.

Під час роботи над проектом було корисно закріпити знання з навчальних дисциплін.

Протягом роботи над проектом були здобуті практичні навички роботи з Telegram API та Firebase Firestore, вдосконалено вміння у використанні Python для створення серверних програм та покращено розуміння інструментів, таких як Figma та Visual Studio Code.

У перспективі планується розширити функціонал Telegram-бота, додавши нові можливості, такі як:

* спільний доступ – дозволити користувачам виконувати групові досягнення;
* інтеграція з календарем – для нагадувань про важливі події або досягнення;
* аналіз даних і персональні рекомендації – для кращої візуалізації прогресу та підвищення мотивації користувачів;
* категорії – для персоналізації досвіду користувача.

Цей проект має перспективу стати цінним інструментом самовдосконалення для користувачів, допомагаючи їм усвідомлювати власний прогрес і розвивати позитивні звички, що є важливими у швидкому темпі сучасного життя.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Figma Learn – Help Center [Електронний ресурс] – Режим доступу до ресурсу: https://help.figma.com/
2. Firebase Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://firebase.google.com/docs
3. Telegram Bots [Електронний ресурс] – Режим доступу до ресурсу: https://core.telegram.org/bots
4. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: https://core.telegram.org/bots/api
5. python-telegram-bot Wiki [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/python-telegram-bot/python-telegram-bot/wiki
6. Documentation for Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: https://code.visualstudio.com/docs
7. Методичні рекомендації до виконання курсових проєктів для здобувачів освіти спеціальності Інженерія програмного забезпечення / Відокремлений структурний підрозділ «Полтавський політехнічний фаховий коледж НТУ «ХПІ»; [уклад.: С.Є.Гриценко]. – Полтава: ВСП ППФК НТУ «ХПІ», 2023. – 27 с.

# ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

1. ПРИЗНАЧЕННЯ І МЕТА СТВОРЮВАНОЇ СИСТЕМИ

1.1. Загальні відомості

Повна назва системи «Автоматизована інформаційна система життєвих досягнень». Коротка назва White Life.

Робота виконується на основі завдання до курсового проєкту з дисципліни Проєктування автоматизованих інформаційних систем.

Термін виконання роботи: 1 вересня 2024 року – 11 листопада 2024 року.

Роботи зі створення АІС життєвих досягнень здаються поетапно згідно з календарним планом.

1.2. Призначення системи

У АІС White Life повинна оброблятися інформація про перебіг виконання досягнень.

Функціонал АІС:

* збереження інформації про перебіг виконання досягнення користувачем;
* формування статистики користувача;
* пошук досягнень для користувача ручним та випадковим способами;
* формування глобальної статистики;
* система друзів зі сповіщеннями про виконання досягнень та переглядом їхньої статистики;
* налаштування приватності профілю користувача;
* можливість користувачу запропонувати досягнення адміністрації;
* зміна системного псевдоніму.

1.3. Мета створення

АІС White Life створюється з метою надати користувачам інструмент для стимуляції їхньої життєвої активності методом пропонування різних життєвих досягнень.

2. ХАРАКТЕРИСТИКА ОБ’ЄКТІВ АВТОМАТИЗАЦІЇ ТА ВИМОГИ ДО СИСТЕМИ

2.1. Відомості про об’єкт автоматизації

Об’єктом автоматизації є:

* процес формування статистики для користувача;
* процес формування глобальної статистики;
* процес випадкового пошуку досягнень.

Особливих умов до експлуатації АІС не висувається

2.2. Вимоги до системи в цілому

АІС повинна бути централізованою, тобто всі дані повинні розташовуватися в центральному сховищі, на сервері бази даних. В системі пропонується виділити наступні функціональні підсистеми:

* підсистема збору, обробки та завантаження даних;
* підсистема зберігання даних;
* підсистема формування даних.

2.3. Вимоги до функцій (задач), що виконуються системою

Система повинна забезпечувати:

* роботу користувачів в режимі - 24 годин на день, 7 днів на тиждень (24х7);
* виконання своїх функцій – збір, обробка та завантаження даних;
* зберігання даних, надання форматованої інформації.

2.4. Вимоги до надійності

Програмний виріб повинен надійно функціонувати в апаратному і програмному середовищі. Відмови, що виникають з-за збоїв апаратних засобів або з-за невідповідності середовища функціонування вимогам, не повинні вести до втрати інформації на накопичувачах.

2.5. Вимоги до інформаційної і програмної сумісності

Для розробки використовувати Firebase Firestore, середовище Visual Studio Code, мову програмування Python, бібліотеку python-telegram-bot.

2.6. Спеціальні вимоги

Спеціальні вимоги до даного програмного продукту не висуваються.

3. СКЛАД І ЗМІСТ РОБІТ ПО СТВОРЕННЮ СИСТЕМИ

Роботи по створенню системи виконуються в три етапи:

* технічне завдання;
* розробка проекту і проектної документації;
* захист проекту.

На етапі «Технічне завдання» має бути виконаний етап розробки, і етап затвердження технічного завдання:

* постановка завдання;
* визначення і уточнення вимог до технічних засобів;
* визначення вимог до програми;
* визначення стадій, етапів і термінів розробки програми і документації на неї;
* вибір мов програмування;
* узгодження і затвердження технічного завдання.

На етапі «Розробка проекту і проектної документації» мають бути виконані перераховані етапи робіт:

* розробка програми (програмування (кодування) і відладки програми);
* розробка програмної документації (розробка пояснювальної записки, технічного завдання, опису програми відповідно до вимог чинних методичних рекомендацій);
* випробування програми (тестування програми; коригування програми і програмної документації за результатами тестування).

На етапі захисту проекту має бути виконана розробка презентації курсового проєкту і представлення його комісії.

4. ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ СИСТЕМИ

При прийманні курсового проєкту перевіряється – комплектність, зміст і оформлення програмної документації.

Захист проводиться перед комісією з трьох викладачів у присутності студентів поточної академічної групи.

# ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ

*from* telegram *import* Update, InlineKeyboardButton, InlineKeyboardMarkup, InputMediaPhoto

*from* telegram.ext *import* Application, CommandHandler, CallbackQueryHandler, MessageHandler, ConversationHandler, ContextTypes, filters

*import* random

*import* firebase\_admin

*import* asyncio

*from* firebase\_admin *import* credentials, firestore

*from* config *import* TOKEN, FIREBASE, ADMIN\_ID

cred = credentials.Certificate(FIREBASE)

firebase\_admin.initialize\_app(cred)

db = firestore.client()

app = Application.builder().token(TOKEN).build()

WAITING\_FOR\_COMPLETION\_DESCRIPTION = 1

WAITING\_FOR\_FRIEND\_CONFIRMATION = 2

async def *start*(update: Update, context: ContextTypes.DEFAULT\_TYPE):

    user = update.effective\_user

    user\_ref = db.collection('users').document(str(user.id))

*if* not user\_ref.get().exists:

        user\_ref.set({

            'username': user.full\_name,

            'completed\_achievements': {}

        }, merge=True)

    achievement\_id = context.args[0] *if* context.args *else* None

*if* achievement\_id:

        achievement = db.collection('achievements').document(achievement\_id).get()

*if* achievement.exists:

            achievement\_data = achievement.to\_dict()

*await* send\_random\_achievement\_details(update.message, achievement\_data)

*return*

    keyboard = [

        [InlineKeyboardButton("🎲 Випадкове досягнення", callback\_data='random')],

        [InlineKeyboardButton("🔍 Знайти досягнення", callback\_data='find')],

        [InlineKeyboardButton("📊 Статистика", callback\_data='stats')],

        [InlineKeyboardButton("👥 Друзі", callback\_data='friends')],

    ]

    reply\_markup = InlineKeyboardMarkup(keyboard)

    user\_data = user\_ref.get().to\_dict()

    username = user\_data.get('username', 'Користувач')

*await* update.message.reply\_text(f"👋  •  Вітаємо у White Life, {username}!\n\n✅  •  Тут ви зможете виконувати різноманітні життєві досягнення та ділитися цим з друзями.\n\n⚙️  •  Функції налаштувать та друзів доступні в меню /command.\n\n🔽  •  Оберіть дію:", reply\_markup=reply\_markup)

if \_\_name\_\_ == "\_\_main\_\_":

app.run\_polling()

Повний код програми доступний на веб-сервісі GitHub за посиланням, що зображене на рисунку Б.1.

  
Рисунок Б.1 – Посилання на репозиторій