Knowlity

Специфікація вимог до програмного забезпечення

1.0

19.05.2025

Дронов Ілля

**Історія версій**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Опис** | **Автор** | **Коментар** |
| 19.05.25 | Версія 1.0 | Дронов Ілля | Створення документу |

**Зміст**

[1. Вступ 4](#_Toc198562600)

[1.1 Огляд продукту 4](#_Toc198562601)

[1.2 Мета 5](#_Toc198562602)

[1.3 Межі 5](#_Toc198562603)

[1.4 Посилання 5](#_Toc198562604)

[1.5 Означення та абревіатури 6](#_Toc198562605)

[2. Загальний опис 6](#_Toc198562606)

[2.1 Перспективи продукту 6](#_Toc198562607)

[2.2 Функції продукту 7](#_Toc198562608)

[2.3 Характеристики користувачів 8](#_Toc198562609)

[2.4 Загальні обмеження 8](#_Toc198562610)

[2.5 Припущення та залежності 9](#_Toc198562611)

[3. Конкретні вимоги 9](#_Toc198562612)

[3.1 Вимоги до зовнішніх інтерфейсів 9](#_Toc198562613)

[3.1.1 Інтерфейс користувача 9](#_Toc198562614)

[3.1.2 Апаратний інтерфейс 14](#_Toc198562615)

[3.1.3 Програмний інтерфейс 14](#_Toc198562616)

[3.1.4 Комунікаційний протокол 14](#_Toc198562617)

[3.1.5 Обмеження пам’яті 14](#_Toc198562618)

[3.1.6 Операції 14](#_Toc198562619)

[3.1.7 Функції продукту 15](#_Toc198562620)

[3.2 Властивості програмного продукту 15](#_Toc198562621)

[3.3 Атрибути програмного продукту 15](#_Toc198562622)

[3.3.1 Надійність 15](#_Toc198562623)

[3.3.2 Доступність 16](#_Toc198562624)

[3.3.3 Безпека 16](#_Toc198562625)

[3.3.4 Супровід 16](#_Toc198562626)

[3.3.5 Переносимість 16](#_Toc198562627)

[3.3.6 Продуктивність 16](#_Toc198562628)

[3.4 Вимоги до бази даних 17](#_Toc198562629)

[3.5 Інші вимоги 17](#_Toc198562630)

[4. Додаткові матеріали 18](#_Toc198562631)

[4.1 Діаграми баз даних 18](#_Toc198562632)

[4.2 Діаграма прецедентів 20](#_Toc198562633)

[4.3 BPMN діаграма 21](#_Toc198562634)

# 1. Вступ

## 1.1 Огляд продукту

Платформа **Knowlity**— це веб-програмна система для надання та отримання освітніх і консультаційних послуг у форматі менторства. Вона дозволяє користувачам створювати профілі менторів або учнів, публікувати оголошення про послуги, бронювати сесії, здійснювати оплату через Stripe та вести комунікацію через вбудований чат (Talk.js). Система спрямована на спрощення доступу до менторської підтримки в різних галузях (програмування, дизайн, маркетинг тощо), підвищення якості взаємодії між користувачами та ефективне керування навчальним процесом. Монетизація реалізується через комісію з проведених транзакцій або платні функції для менторів.

## 1.2 Мета

Метою розробки програмної системи є створення безпечної, масштабованої та зручної платформи, яка дозволяє користувачам у ролі менторів та учнів ефективно взаємодіяти між собою. Система повинна забезпечити централізоване управління профілями, можливість фільтрації послуг за категоріями, швидке бронювання, онлайн-оплату, а також інтегрований канал комунікації. Проєкт також спрямований на підвищення якості надання освітніх послуг, розширення доступу до досвідчених фахівців, а також розвиток культури наставництва. Платформа має на меті створити позитивний користувацький досвід завдяки сучасному інтерфейсу, простому процесу реєстрації та продуманій системі ролей.

## 1.3 Межі

Програмна система Knowlity включає веб-додаток, який реалізує повний функціонал платформи: реєстрація, автентифікація (через email та Google), створення і перегляд оголошень, внутрішній чат, перегляд розкладу, бронювання сесій та здійснення платежів. Система не включає окремий мобільний додаток. Підтримуються три основні ролі користувачів: учень, ментор та адміністратор, кожна з яких має доступ лише до відповідного функціоналу.

Серверна частина реалізована за допомогою FastAPI у монолітній структурі, однак з чітким поділом відповідальності між модулями. Клієнтська частина створена на базі Next.js з використанням компонентної архітектури. Контроль за процесом розробки здійснюється через систему управління проектами Trello.

## 1.4 Посилання

Текст документу містить наступні посилання:

1. Посилання на репозиторії проекту:   
   <https://github.com/TeraBasedProgrammer/bachelor-work-backend>

<https://github.com/TeraBasedProgrammer/bachelor-work-backend>

## 1.5 Означення та абревіатури

В документі використовуються наступні означення та абревіатури:

HTTP – HyperText Transfer Protocol

REST – REpresentational State Transfer

SQL – Structured Query Language

AWS – Amazon Web Services

EC2 - Amazon Elastic Compute Cloud

K8S - Kubernetes

# 2. Загальний опис

## 2.1 Перспективи продукту

У зв’язку зі зростанням попиту на індивідуальне навчання, розвитком EdTech-сектору та популяризацією моделі mentor-to-mentee, створення платформи **Knowlity** є актуальним та комерційно доцільним рішенням. Онлайн-менторство стало важливою складовою кар’єрного розвитку, перекваліфікації та професійного зростання, особливо в умовах гнучкого ринку праці. За даними LinkedIn Learning, попит на персоналізоване навчання щороку зростає, що створює сприятливе середовище для подібних цифрових сервісів.

Програмна система **Knowlity** орієнтована на такі сегменти ринку:

1. **Ментори-фахівці** — можуть проводити монетизацію своїх знань, створюючи публічні профілі та приймаючи бронювання.
2. **Учні та фахівці-початківці** — отримують швидкий доступ до консультацій, навчальних сесій та фідбеку.
3. **B2B-партнери (курси, школи, HR-компанії)** — можуть інтегрувати платформу як частину внутрішніх програм навчання або профорієнтації.
4. **EdTech-ринки** — розширення функціоналу платформи дозволяє масштабувати її в сегменти колективного навчання, міні-курсів та сертифікації.
5. **Фриланс та консультаційні ринки** — за допомогою Knowlity ментори можуть не лише навчати, а й супроводжувати фахівців у робочих проектах.
6. **Платформи для професійного зростання** — система може стати інструментом для побудови персоналізованих траєкторій розвитку кар’єри.

## 2.2 Функції продукту

Knowlity — це веб-платформа з повним циклом взаємодії між ментором і учнем. Серед основного функціоналу:

* **Профілі користувачів** — реєстрація, налаштування профілю, вибір ролі (ментор, учень, обидва).
* **Публікація оголошень** — створення пропозицій щодо консультацій або менторських сесій з можливістю вказати категорію, ціну та опис.
* **Фільтрація та пошук** — розширений пошук менторів за спеціалізацією, ціною, рейтингом тощо.
* **Система бронювання** — можливість бронювати сесії, керувати розкладом.
* **Платіжна система** — інтеграція з Stripe для здійснення онлайн оплат.
* **Чат** — вбудований месенджер на основі Talk.js для попереднього спілкування.
* **Панель адміністратора** — верифікація менторів.
* **Система повідомлень** — email та in-app сповіщення про бронювання, нові повідомлення та статуси оплати.
* **Безпека** — автентифікація через JWT/Google, обмеження доступу до чутливих дій згідно ролей.

## 2.3 Характеристики користувачів

Система Knowlity взаємодіє з трьома основними типами користувачів:

* **Ментор** — створює оголошення, керує сесіями, спілкується з учнями, переглядає історію оплат.
* **Учень** — переглядає оголошення, фільтрує менторів, бронює сесії, оплачує послуги.
* **Адміністратор** — проводить верифікацію менторів

Усі користувачі можуть зареєструватися за допомогою email або Google, редагувати профіль, переглядати історію своїх взаємодій та отримувати повідомлення про дії в системі.

## 2.4 Загальні обмеження

Knowlity складається з клієнтської та серверної частин. Сервер реалізовано за допомогою **FastAPI** (Python), клієнт — з використанням **Next.js**. Архітектура побудована у вигляді модульного моноліту з чітко поділеними зонами відповідальності, що дозволяє легко масштабувати окремі компоненти.

Для зберігання даних використовується **PostgreSQL**, медіа файли обробляються через **Cloudinary**. Платформа розгортається в хмарі (AWS EC2), з підтримкою CI/CD (GitHub Actions). У майбутньому розглядається можливість переходу до мікросервісної структури з використанням та K8S.

Клієнтська частина розробляється з урахуванням адаптивності, використовуючи Tailwind CSS та компонентну архітектуру. NextAuth забезпечує захищену автентифікацію, а Talk.js — стабільну чат-комунікацію. Stripe використовується для обробки онлайн-платежів.

## 2.5 Припущення та залежності

**Припущення:**

1. Ринок потребує інструментів для менторства та індивідуального навчання.
2. Користувачі мають базову цифрову грамотність для взаємодії з платформою.
3. Інтегровані сторонні сервіси (Stripe, Talk.js) працюватимуть стабільно.

**Залежності:**

1. Зовнішні API: Stripe (оплата), Talk.js (чат), Google OAuth (автентифікація).
2. Серверна частина — FastAPI, PostgreSQL, Cloudinary.
3. Клієнтська частина — Next.js, Tailwind CSS, NextAuth, React Hook Form.
4. CI/CD — GitHub Actions.
5. Хостинг — AWS.
6. Продуктивність платформи залежить від стабільності інтернет-з'єднання користувача та середовища хостинга.
7. Процес розробки залежить від часу команди на вивчення, впровадження й тестування функціоналу.

# 3. Конкретні вимоги

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Після переходу на платформу Knowlity, неавторизованого користувача автоматично перенаправляє на сторінку входу в систему (рисунок 1).

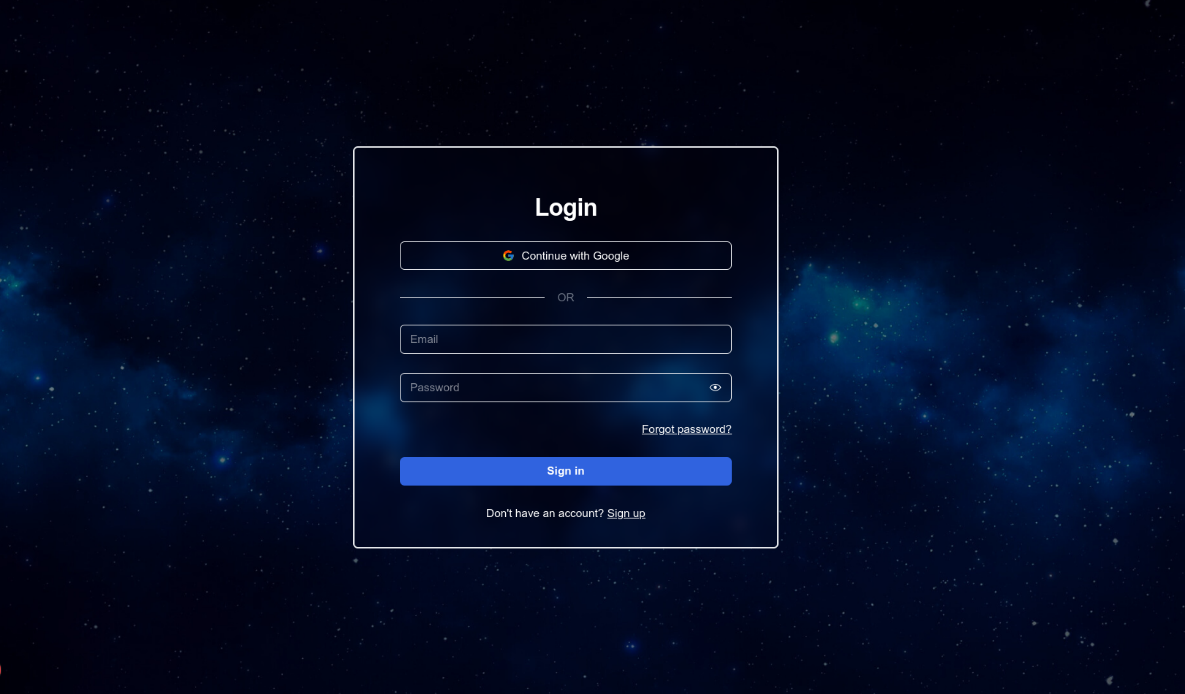


Рисунок 1 — Сторінка входу

Сторінка входу містить форму автентифікації, яка включає поля для введення електронної пошти та пароля, а також кнопку входу через Google. Під формою знаходиться кнопка переходу на сторінку реєстрації. Оформлення має бути простим, інтуїтивно зрозумілим, з фокусом на зручність входу до системи.

На сторінці реєстрації (рисунок 2) користувач має змогу створити обліковий запис як учень. Поля реєстрації включають: електронну пошту, номер телефону, ім’я та пароль. Також присутня кнопка входу через Google. Роль користувача не вибирається вручну — за замовчуванням усі зареєстровані користувачі отримують роль учня. Перед завершенням реєстрації користувач має погодитися з політикою конфіденційності. Додатково присутня кнопка переходу на форму входу.

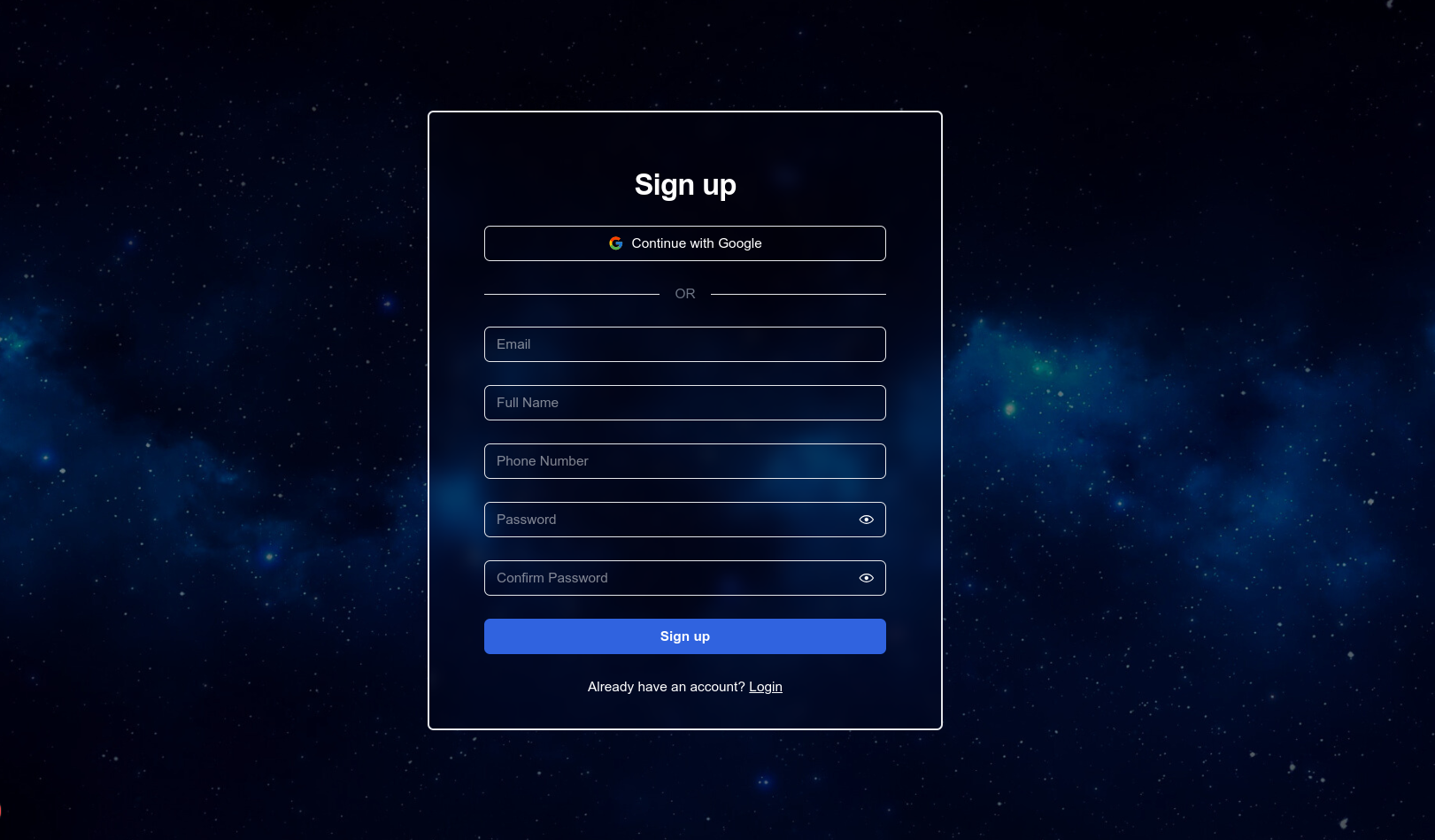
****

Рисунок 2 — Форма реєстрації користувача

Після входу до системи користувач потрапляє у профіль. Спочатку йому пропонується заповнити базові особисті дані (рисунок 3): завантажити фотографію, додати номер телефону, обрати категорії для пошуку оголошень.

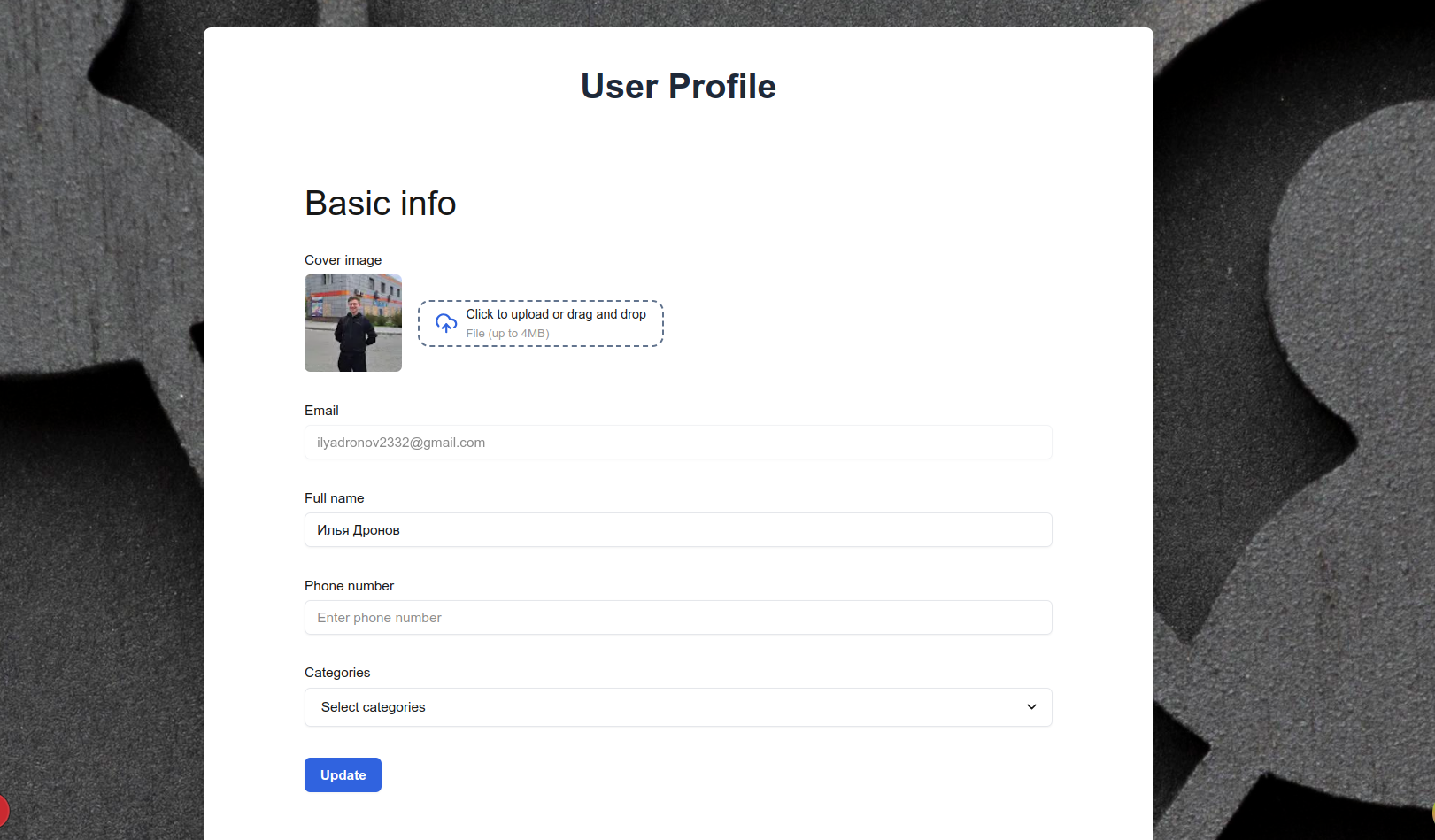


Рисунок 3 — Заповнення особистого профілю

Після цього користувач може подати заявку на створення менторського профілю (рисунок 4). Ця форма включає: завантаження фото паспорта, текст про себе, відео про себе, резюме (файл) та вибір категорій, у яких користувач бажає надавати менторські послуги.

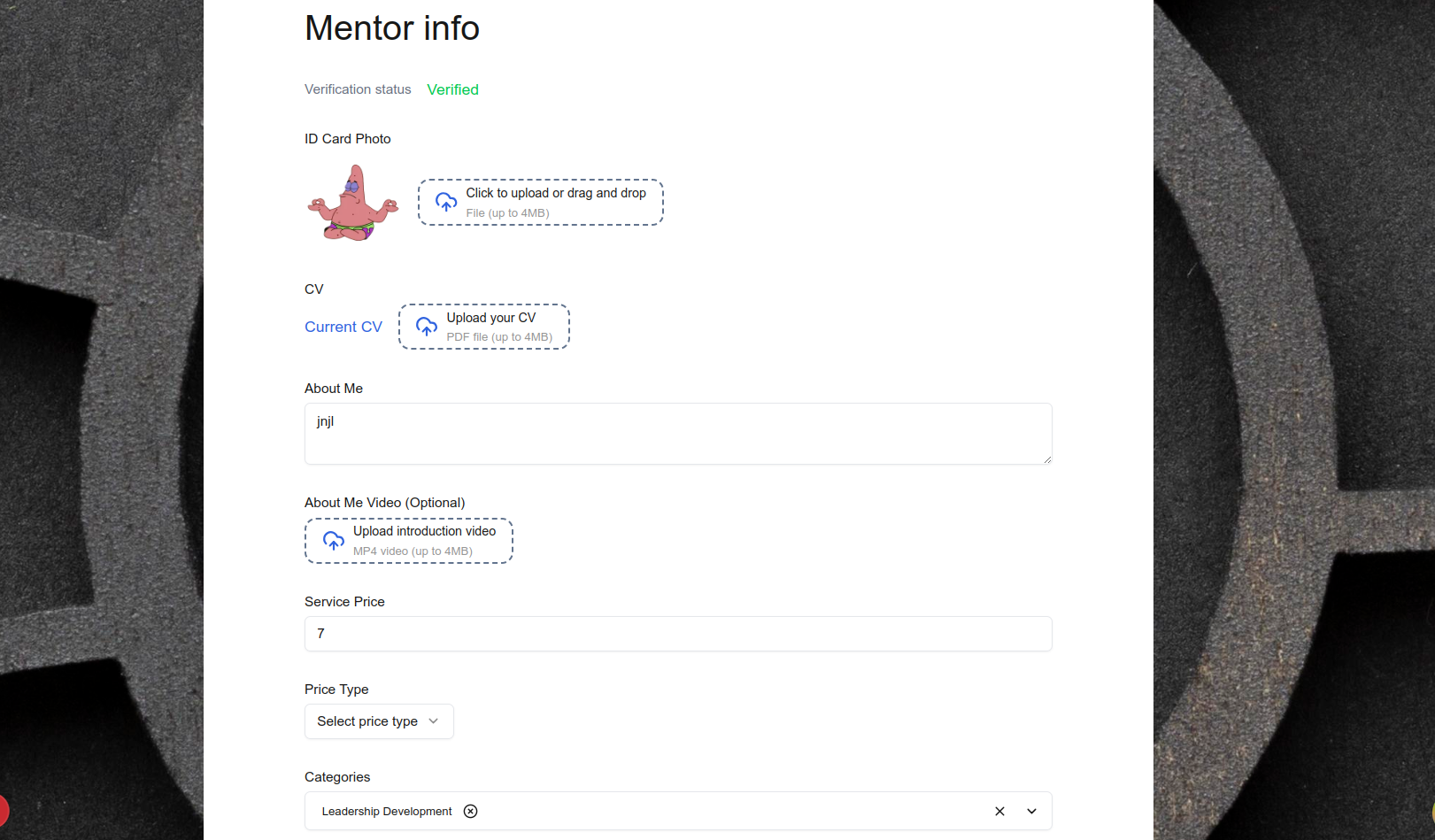


Рисунок 4 — Форма створення менторського профілю

Комунікація між учнем і ментором реалізована у вигляді вбудованого чату (рисунок 5). У чаті відображаються повідомлення, інформація про співрозмовника та зручний інтерфейс для відповіді. Платформа підтримує як текстові повідомлення, так і push-сповіщення про нові вхідні.

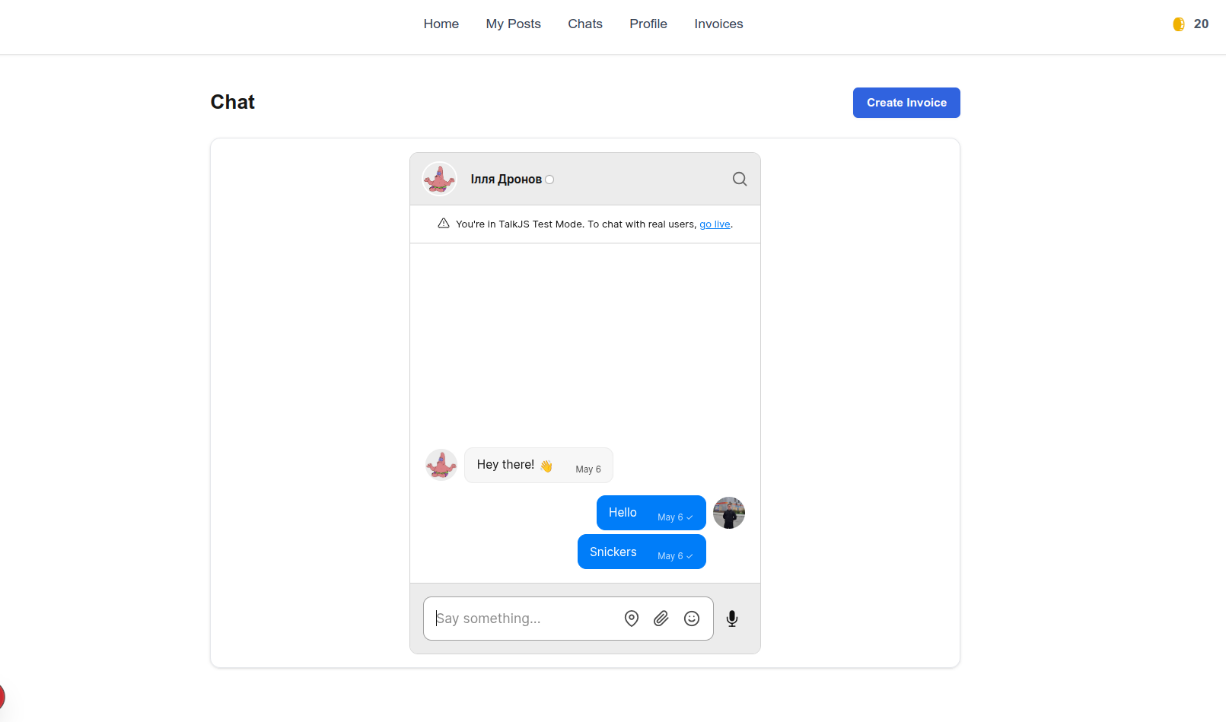


Рисунок 5 — Чат між користувачами

Усі фінансові операції (оплати сесій, комісії) фіксуються у вигляді інвойсів, які доступні користувачам у відповідному розділі профілю (рисунок 6). Кожен інвойс містить інформацію про дату, суму та сторони оплати

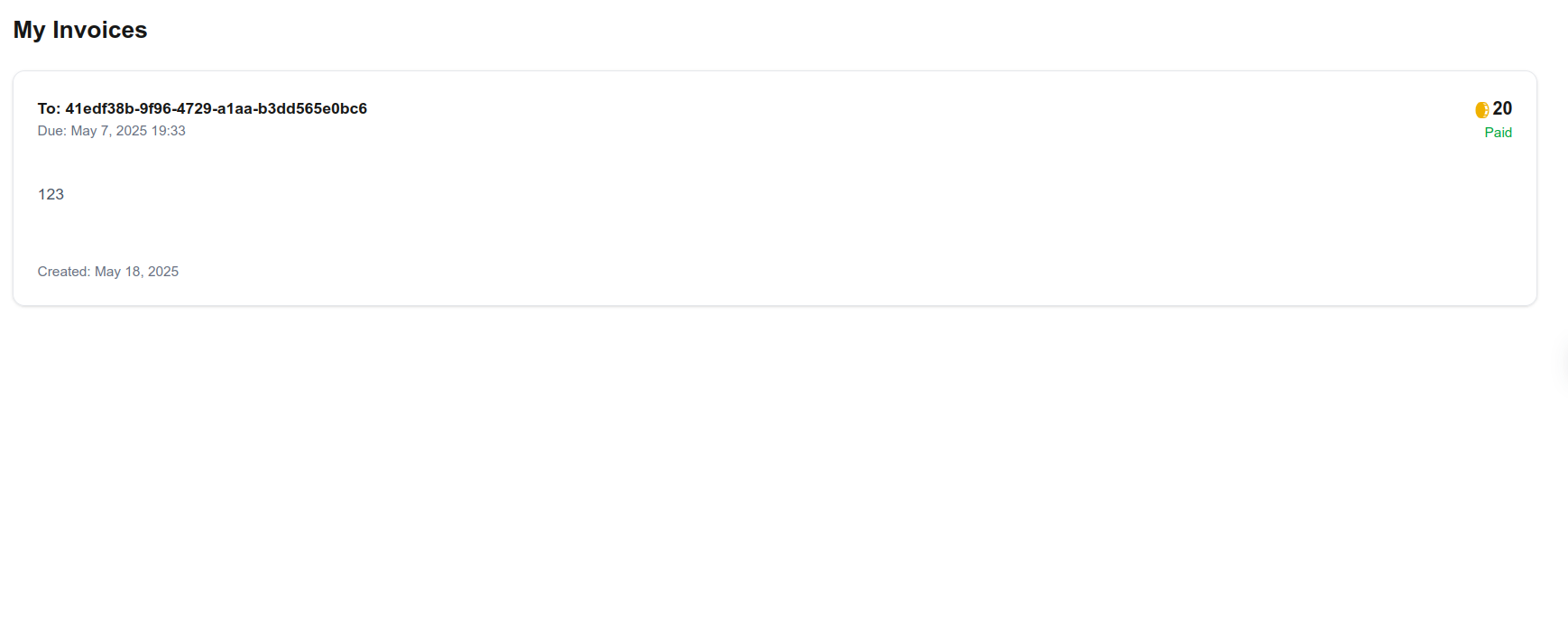


Рисунок 6 — Інвойс користувача

### 3.1.2 Апаратний інтерфейс

Система Knowlity не взаємодіє з апаратними пристроями напряму, тому специфічні апаратні інтерфейси відсутні.

### 3.1.3 Програмний інтерфейс

Система сумісна з усіма сучасними браузерами (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge). Платформа розроблена з використанням HTML5, CSS3, JavaScript ES6, React та Next.js. Мобільна версія доступна через адаптивний веб-інтерфейс, окремі застосунки для iOS/Android поки не розробляються.

### 3.1.4 Комунікаційний протокол

Взаємодія між клієнтською та серверною частинами реалізується через HTTPS та REST API. Комунікація в реальному часі (чат, оновлення повідомлень) забезпечується через WebSocket (Talk.js). Внутрішня взаємодія між модулями може доповнюватися асинхронною обробкою через черги повідомлень (у майбутньому).

### 3.1.5 Обмеження пам’яті

Жодних специфічних обмежень пам’яті не встановлюється. Платформа працює у веб-браузері, і пам’ять визначається ресурсами пристрою користувача та налаштуваннями браузера.

### 3.1.6 Операції

Платформа реалізує REST-архітектуру з такими HTTP-методами:

* GET — отримання даних (профіль, оголошення, повідомлення)
* POST — створення (реєстрація, нове бронювання)
* PUT / PATCH — редагування (профілю, послуги)
* DELETE — видалення (бронювання, оголошення)

### 3.1.7 Функції продукту

ГФ-1: Реєстрація та автентифікація користувачів  
ГФ-2: Створення, перегляд, редагування та видалення оголошень  
ГФ-3: Бронювання сесій між учнями та менторами  
ГФ-4: Інтеграція з Stripe для обробки платежів  
ГФ-5: Вбудований чат між користувачами (Talk.js)  
ГФ-6: Рейтингова система, можливість залишати відгуки  
ГФ-7: Панель адміністратора для модерації контенту та верифікації менторів  
ГФ-8: Система повідомлень (in-app та email)  
ГФ-9: Багатомовна підтримка (українська / англійська)

## 3.2 Властивості програмного продукту

Платформа Knowlity орієнтована на широку аудиторію з різних регіонів. Стабільне інтернет-з’єднання є обов’язковим. Час відповіді системи не повинен перевищувати 1 хвилини, навіть при високому навантаженні. Розмежування функціоналу здійснюється за ролями (ментор, учень, адміністратор). Дані зберігаються централізовано у захищеній базі даних PostgreSQL. Інтерфейс адаптивний і підтримує мобільні пристрої.

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

* Система повинна повертати лише перевірені дані.
* У випадку помилки — показує зрозуміле повідомлення без внутрішніх технічних деталей.

### 3.3.2 Доступність

* Доступна українською та англійською мовами.
* Доступ до функціоналу мають лише авторизовані користувачі.
* Адміністративні функції доступні лише адміністраторам.

### 3.3.3 Безпека

* Обліковий запис захищений паролем (мінімум 6 символів і хоча б одна цифра).
* Доступ лише після вірного введення логіна і пароля / авторизації через Google.
* Дані компаній зберігаються окремо від даних користувачів.
* Функціонал доступний згідно з ролями.

### 3.3.4 Супровід

* Система має бути легко розширювана новими функціями.
* Компоненти системи (сервер, клієнт, чат, оплати) мають бути відокремлені.

### 3.3.5 Переносимість

* Веб-додаток підтримується у сучасних браузерах.
* Адаптивний дизайн забезпечує підтримку планшетів і смартфонів.

### 3.3.6 Продуктивність

* Перехід між вкладками — до 1 секунди.
* Лист підтвердження дій (наприклад, запрошення) надсилається протягом 5 хвилин.

## 3.4 Вимоги до бази даних

У проєкті Knowlity використовується PostgreSQL як основна СУБД. Переваги:

* висока масштабованість;
* транзакційна надійність (ACID);
* потужні засоби контролю доступу;
* підтримка складних зв’язків та індексації;
* інтеграція з FastAPI та SQLAlchemy.

База даних зберігає користувачів, оголошення, категорії, сесії, транзакції, повідомлення, відгуки та адміністративні записи.

### 3.5 Інші вимоги

* Незареєстровані користувачі мають доступ лише до загальнодоступної головної сторінки.
* Уся інша функціональність вимагає авторизації.
* Всі використовувані зображення, ілюстрації, та медіа мають бути вільні від порушення авторських прав.
* Інтерфейс має бути повністю адаптивним — автоматично підлаштовуватися під ширину екрана.
* Усі змінні шрифти, кнопки, елементи інтерфейсу повинні бути придатні до взаємодії на мобільних пристроях.

# 4. Додаткові матеріали

## 4.1 Діаграми баз даних

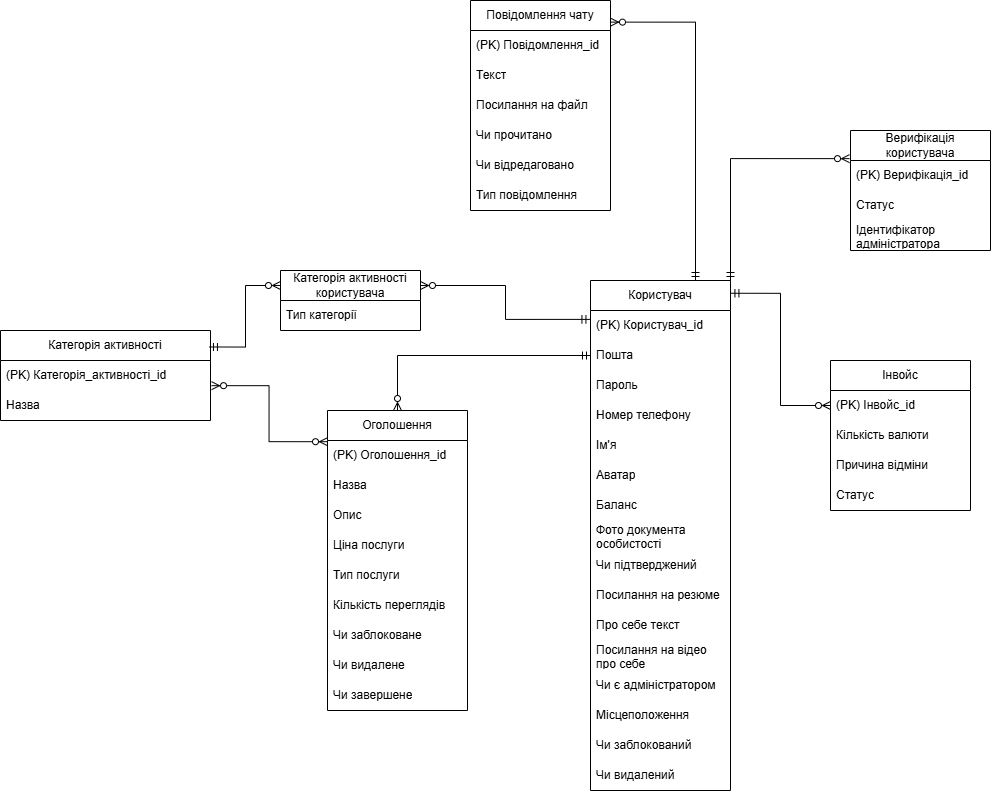


Рисунок 7 – ER-діаграма бази даних

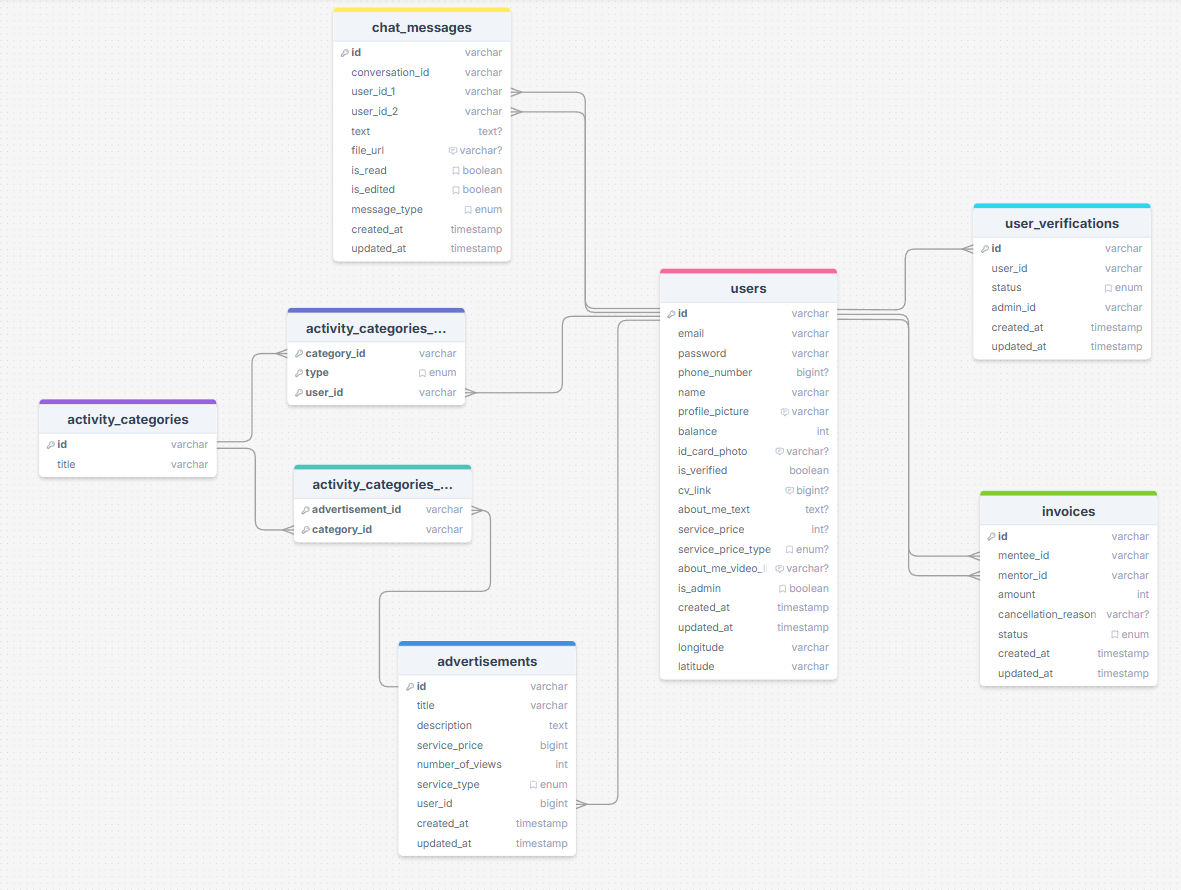


Рисунок 8 – Логічна модель бази даних

## 4.2 Діаграма прецедентів

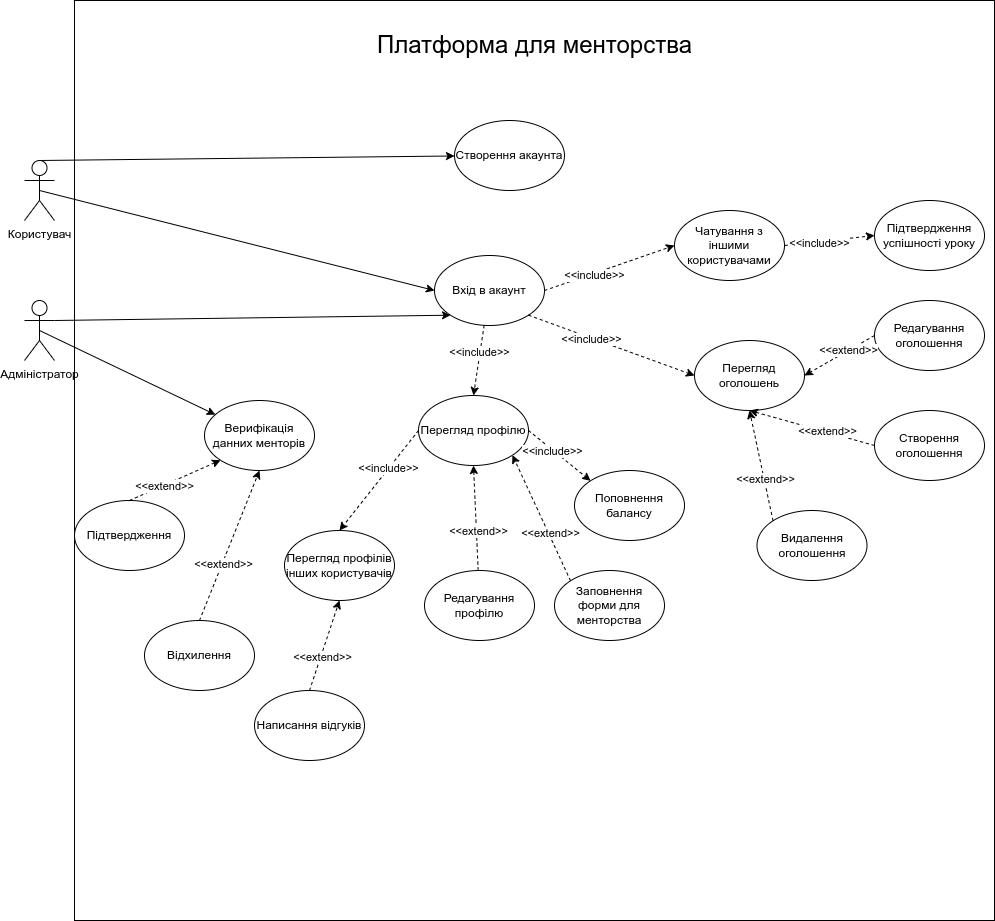


Рисунок 9 – Діаграма прецедентів

## 4.3 BPMN діаграма

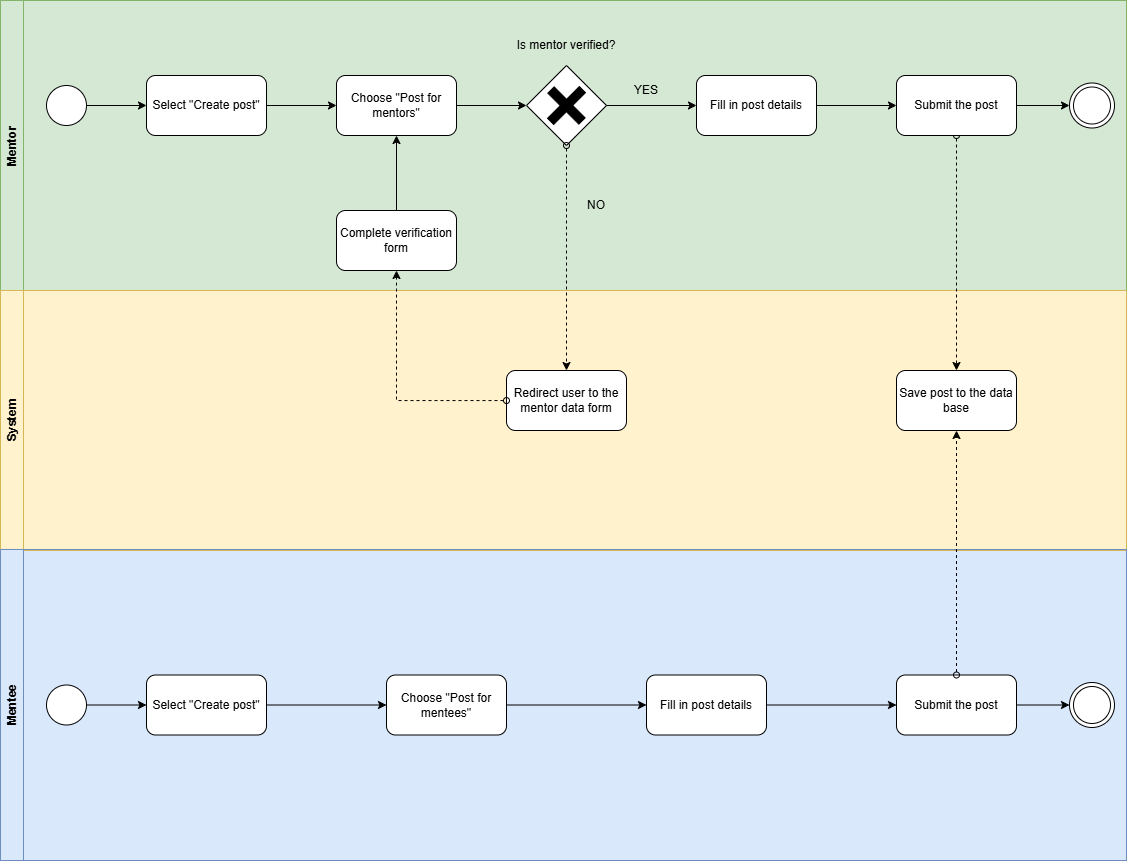


Рисунок 10 – BPMN діаграма