

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»

на тему: Інформаційна система підтримки надання психологічної допомоги.
Підсистема клієнта

Здобувача (ки) групи ІТ.м-23 Владислав СІРИК
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

(підпис)

Владислав СІРИК
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник _____ к.т.н., доц. Юлія ПАРФЕНЕНКО _____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«___» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Сірик Владислав Юрійович

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Інформаційна система підтримки надання психологічної допомоги. Підсистема клієнта

Затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом закінченого проекту «___» грудня 2023 р.

3 Вхідні дані до проекту Завдання на розробку підсистеми клієнта для інформаційної системи надання психологічної допомоги.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) Аналіз предметної області, постановка задачі та методів її дослідження, проектування інформаційної системи, реалізація підсистеми клієнта для інформаційної системи підтримки психологічної допомоги.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Актуальність проблеми, мета проекту, постановка задачі, функціональні вимоги до проекту, контекстна діаграма використання інформаційної системи, декомпозиція першого рівня в нотації IDEF0, схема інформаційної технології, діаграма варіантів використання, фізична модель даних, засоби реалізації, практична реалізація, висновок.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	09.09. 2023 –	

		03.10. 2023	
2	Постановка задачі та методи дослідження	28.09.2023 – 04.10.2023	
3	Проектування інформаційної системи	05.10.2023 – 10.10-2023	
4	Реалізація інформаційної системи	10.10.2023 – 10.11.2023	
5	Тестування інформаційної системи	11.11.2023 – 16.11.2023	
6	Здача роботи	08.12.2023 – 15.12.23	

Магістрант _____

Владислав СІРИК

Керівник роботи _____

к.т.н., доц. Юлія ПАРФЕНЕНКО

АНОТАЦІЯ

Тема магістерської кваліфікаційної роботи «Інформаційна система підтримки надання психологічної допомоги. Підсистема клієнта» .

Структура роботи включає в себе вступ, чотири розділи, висновки, список використаних джерел та додатки. Робота має загальний обсяг у 84 сторінок, в якому 48 сторінок припадає на основний текст, 4 сторінки на список використаних джерел та 25 сторінок на додатки.

Перший розділ присвячений огляду та аналізу актуальності проблеми. Проаналізовано та проведено огляд сучасних аналогів для надання психологічної підтримки. Створена порівняльна таблиця систем і виявлені їхні переваги та недоліки.

У другому розділі визначається мета та завдання дослідження. Визначені інструменти для реалізації інформаційної системи.

Третій розділ присвячений створенню структури проекту, де побудовані діаграми роботи інформаційної системи, діаграми декомпозиції та діаграма використання.

У четвертому розділі наведено процес реалізації додатку та приклади роботи веб-додатку, спрямованого на надання психологічної допомоги для клієнту.

Результатом роботи є створений підсистема клієнта для інформаційної системи підтримки надання психологічної допомоги у вигляді веб-додатку та серверу з його контролерами, яка надає можливість перегляду, пошуку та консультації із психотерапевтами.

Ключові слова: веб-додаток, інформаційна система, психологічна допомога, онлайн-консультація, анонімність, сервер, психотерапевт, клієнт-серверна архітектура.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Огляд проблеми та актуальність інформаційної системи.....	9
1.2 Аналіз сучасних аналогів	14
2 ПОСТАНОВКА ЗАДАЧІ.....	22
2.1 Мета та задачі дослідження.....	22
2.2 Технології та інструменти реалізації підсистеми клієнта.....	24
3 ПРОЕКТУВАННЯ ПІДСИСТЕМИ КЛІЄНТА	28
3.1 Структурно-функціональне моделювання процесу.....	28
3.2 Моделювання діаграми варіантів використання.....	30
3.3 Проектування моделі бази даних.....	34
4 РОЗРОБКА ПІДСИСТЕМИ КЛІЄНТА.....	43
4.1 Розробка архітектури інформаційної системи	43
4.2 Розробка веб-додатку для підсистеми клієнта	45
4.3 Розробка серверної частини для підсистеми клієнта	51
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А. ПЛАНУВАННЯ РОБІТ	59
А.1 Ідентифікація мети ІТ-проекту	59
А.2 Планування змісту структури робіт інформаційної системи	60
А.3 Побудова календарного графіку виконання інформаційної системи	62
А.4 Планування ризиків проекту	63
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ	72

ВСТУП

Актуальність теми зумовлена тим, що сучасний світ інформаційних технологій постійно розвивається, вносячи зміни у всі сфери людського життя, включаючи надання психологічної допомоги. Із зростанням глобалізації та поширенням цифрових технологій, важливість інформаційних систем, що підтримують надання психологічної допомоги, стає все більш відчутною. Ця тема набуває актуальності у контексті зростаючої потреби в психологічній підтримці, особливо в умовах, коли традиційні методи консультацій можуть бути обмежені через різноманітні соціальні та географічні фактори.

Проблема полягає у необхідності створення ефективної, надійної та доступної інформаційної системи, яка б забезпечувала якісне надання психологічної допомоги. Важливо враховувати не лише технічні аспекти таких систем, але й психологічні, етичні та соціальні вимоги, що є ключовими для їх ефективності та прийнятності серед користувачів.

Об'єктом дослідження є інформаційне забезпечення підтримки надання психологічної допомоги через Інтернет.

Предметом дослідження є використання сучасних технологій та підходів в розробці інформаційної системи для психологічної допомоги, її архітектури, а також інтерфейсу.

Метою дослідження є розробка інформаційної системи, яка ефективно підтримуватиме процес надання психологічної допомоги, враховуючи сучасні технологічні можливості та специфічні потреби користувачів.

Для досягнення поставленої мети було визначено основні задачі:

- провести аналіз існуючих рішень у сфері інформаційних систем для психологічної допомоги;
- провести огляд найновіших наукових досліджень і опублікованих матеріалів;

- визначити ключові вимоги функціональності для ефективної системи психологічної підтримки;
- розробити підсистему клієнта для інформаційну систему підтримки надання психологічної допомоги.

Практична значущість дослідження виражається у використанні розробленої системи в реальних умовах для поліпшення якості та доступності психологічної допомоги. Така система може бути корисною для психологічних центрів, освітніх закладів та інших організацій, що займаються психологічною підтримкою.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд проблеми та актуальність інформаційної системи

Останнім часом використання інтернет-технологій у психологічній практиці активно обговорюється в наукових публікаціях. Дослідники з України поділяють думку про значний потенціал Інтернету у сфері психологічної допомоги, але водночас вказують на існуючі виклики, які потребують подолання [1].

Дистанційне консультування сьогодні стало відповіддю на мінливі умови життя, коли особистий, віч-на-віч контакт став майже неможливим. У наш час у людей виникає потреба працювати віддалено та отримувати психологічну підтримку онлайн. Цей метод дозволяє надавати консультації у спосіб, який є здоровим як для клієнтів, так і для психологів-консультантів, і при цьому залишається повністю ефективним [2, 3].

Однією з головних переваг такого підходу є зручність для клієнта. Люди можуть отримати психологічну підтримку, не виходячи з дому чи офісу, що особливо важливо в сучасному мінливому середовищі. Це робить процес консультування більш гнучким і пристосованим до індивідуальних потреб кожного клієнта.

Крім того, дистанційне консультування відкриває можливості для роботи з клієнтами з різних географічних регіонів і навіть країн. Це особливо корисно в сучасному світі, де люди можуть переїжджати або перебувати в різних місцях через роботу, навчання чи інші обставини.

Ще однією важливою перевагою дистанційного консультування є можливість анонімності. Клієнти можуть отримати допомогу, не відчуваючи страху, викликаного необхідністю особистого спілкування з психологом. Це особливо цінно в тих сферах, де важлива конфіденційність.

Звичайно, дистанційне консультування підходить не для всіх випадків і не для

всіх людей. Однак його зростаюча популярність свідчить про те, що цей метод є важливим інструментом забезпечення доступу до психологічної підтримки в сучасному світі.

Розуміння особливостей цільової аудиторії та їхніх потреб є фундаментальним для розробки та ефективного впровадження інформаційних систем психологічної допомоги. Цільова аудиторія може варіюватись за багатьма параметрами, включаючи вік, стать, культурний та соціально-економічний статус. Наприклад, молоді користувачі можуть шукати більш інтерактивні рішення, тоді як старше покоління може віддавати перевагу більш традиційним формам комунікації.

Психологічні потреби цільової аудиторії також різняться: від легкого стресу і тривоги до більш серйозних станів, таких як депресія або посттравматичний стресовий розлад. Системи повинні бути достатньо гнучкими, щоб адаптуватися до різних рівнів і особливостей цих потреб.

Важливим аспектом є також технічні навички та доступність. Для деяких користувачів, особливо тих, хто старший за віком або має обмежений доступ до технологій, системи повинні бути простими та інтуїтивно зрозумілими. Культурні та мовні відмінності також можуть впливати на сприйняття та ефективність психологічної допомоги. Системи мають бути культурно чутливими та доступними на різних мовах, щоб забезпечити рівний доступ для всіх користувачів.

Конфіденційність інформації – одна з найважливіших потреб, особливо при обговоренні делікатних психологічних питань. Забезпечення безпеки та конфіденційності персональних даних користувачів має важливе значення для збереження довіри та ефективності системи. Врахування цих різних аспектів вимагає цілісного підходу до розробки інформаційних систем, щоб вони відповідали різноманітним потребам і очікуванням користувачів. Це включає не лише технічні аспекти, але й глибоке розуміння психологічних, культурних та соціальних факторів, які впливають на поведінку та досвід людей.

Більшість інформаційних систем психологічної допомоги розробляються як веб-додатки або мобільні додатки [4]. Це забезпечує широкий доступ до послуг,

дозволяючи користувачам отримувати підтримку в будь-який час і в будь-якому місці. Мобільні додатки особливо корисні для постійного моніторингу настрою користувачів та надання негайної підтримки [5].

Штучний інтелект та машинне навчання використовуються для аналізу даних користувачів, прогнозування шаблонів поведінки та надання персоналізованих рекомендацій [6]. Ці технології дозволяють системам адаптуватися до індивідуальних потреб користувачів та надавати більш цілеспрямовану допомогу.

Хмарні рішення забезпечують гнучкість і масштабованість систем, дозволяючи легко розширювати ресурси відповідно до зростаючого попиту [7]. Хмарні платформи також допомагають знизити витрати на утримання та управління інфраструктурою.

Завдяки цим технологічним рішенням (табл. 1.1), інформаційні системи психологічної допомоги можуть ефективно надавати підтримку великій кількості користувачів, адаптуючись до їх індивідуальних потреб та особливостей. Вони також відіграють важливу роль у розширенні доступу до психологічних послуг, особливо у районах, де такий доступ традиційно обмежений [8, 9].

Таблиця 1.1 – Ключові технології та їх особливості

Технологія	Особливості	Приклади застосування
Безпека даних та конфіденційність	Шифрування, безпечні протоколи	Захист інформації про стан здоров'я, захист конфіденційних даних
Інтерактивний дизайн та UX/UI	Інтуїтивний інтерфейс, зручність	Легкість використання
Веб-технології та мобільні додатки	Доступність, мобільність, зручність	Відео та текстові чати, мобільні додатки для моніторингу настрою

Продовження таблиці 1.1

Штучний інтелект та машинне навчання	Персоналізація, аналіз поведінки	Прогнозування патернів поведінки, персоналізовані рекомендації
Хмарні технології	Масштабованість, гнучкість, зниження витрат	Розширення ресурсів, зберігання даних, хмарні платформи для послуг
Інтеграція з іншими системами	Обмін даними, комплексність	Зв'язок з медичними записами, інтеграція з оздоровчими платформами

Джерело: власна розробка

Ця таблиця підсумовує основні технологічні рішення, їх особливості та застосування в контексті інформаційних систем для психологічної допомоги. Вона відображає, як різні технологічні рішення сприяють поліпшенню доступності, ефективності та безпеки цих систем.

Вплив інформаційних систем на ефективність психологічної допомоги є ключовим аспектом при розробці та впровадженні цих технологій. Це важливо для того, щоб цифрові рішення не лише відповідали технічним вимогам, але й дійсно покращували якість та доступність психологічної допомоги.

На сьогоднішній день існує різноманітні типи дистанційних сервісів, які надають клієнтам можливість отримати психологічну підтримку та консультування в онлайн-форматі.

Загалом ці сервіси можна розділити на дві категорії: розмовні боти, що використовують техніку активного слухання, та онлайн-консультації. Як правило, у таких ботах відсутній персоналізований підхід до потреб клієнта, а їхня функція полягає в тому, щоб надати можливість просто висловити свої думки. Онлайн-консультації з психотерапевтом, з іншого боку, надають кваліфіковану допомогу,

але вартість цих послуг може бути досить високою, що ускладнює використання такої інформаційної системи протягом тривалого періоду часу [10]. Кожна з цих груп має свої переваги та недоліки.

Отже, для проведення онлайн-консультацій існують інформаційні системи, що дозволяють користувачам отримувати психологічну допомогу через Інтернет. Ці системи зазвичай включають відео, аудіо та текстові чати, а також можливість використання електронної пошти [11]. Головна перевага онлайн-консультацій полягає у їхній доступності та зручності, оскільки користувачі можуть отримувати консультації в зручний для них час і місце [12]. Крім того, ці системи часто використовуються для допомоги людям, які живуть у віддалених або малонаселених районах, де доступ до традиційних форм психологічної допомоги може бути обмеженим.

Розмовні боти – це програми, які можуть взаємодіяти з людьми та надавати їм психологічну підтримку. Маючи багато переваг перед традиційними методами психотерапії, розмовні боти доступні цілодобово і можуть використовуватися будь-де. Вони також можуть бути більш прийнятними для людей, які не відчують себе комфортно в спілкуванні з живим психотерапевтом.

Однак важливо зазначити, що, не зважаючи на ці переваги, боти психологічної підтримки, що розмовляють, не є заміною традиційної психотерапії. Вони можуть бути корисним доповненням до неї, але не можуть надати такого ж рівня підтримки та допомоги, як професійний психотерапевт. Враховуючи ці переваги та недоліки, розробники систем психологічної підтримки повинні прагнути до створення балансу між наданням високоякісної допомоги, збереженням конфіденційності та забезпеченням легкого доступу для користувачів [13].

Інформаційні системи можуть ефективно доповнювати традиційні методи психологічної допомоги. Це включає можливість використання онлайн-ресурсів як частини комплексного лікування, що об'єднує цифрові та особистісні підходи [14].

Актуальність цього завдання обумовлена відсутністю доступних у

відкритому режимі інформаційних систем, що ускладнює можливість створення нових систем на основі існуючих. Це обмеження унеможлиблює додавання, заміну або удосконалення існуючих інформаційних систем.

1.2 Аналіз сучасних аналогів

Загалом існуючі системи психологічної допомоги надають цінний огляд поточних тенденцій та викликів у цій сфері. Розуміння цих систем допомагає у формуванні вимог до нових розробок, щоб забезпечити їхню ефективність, доступність та прийнятність для користувачів [15].

Аналізуючи сучасні системи психологічного допомоги, слід відзначити низку переваг, які визначають їх ефективність та актуальність у практичному застосуванні. Доступність є однією із ключових характеристик сучасних систем. Онлайн-формат надає можливість отримувати психологічну допомогу в будь-який час та в будь-якому місці, що стає важливим у віддалених або малонаселених регіонах. Такий підхід вирішує проблему географічних обмежень та поліпшує загальний рівень доступності психологічних послуг [16, 17].

Фактор анонімності також відіграє важливу роль у сучасних системах психологічної підтримки. Багато користувачів відчують себе більш комфортно, обговорюючи особисті проблеми в анонімному режимі, що сприяє їх більш відкритому спілкуванню з фахівцем. На етапі оцінки вартості важливо враховувати, що онлайн-системи часто є більш доступними, ніж традиційні методи психотерапії. Це робить психологічну допомогу фінансово доступною для ширшого кола користувачів.

Сучасні системи психологічної допомоги відзначаються різноманіттям послуг і ресурсів. Вони пропонують широкий спектр можливостей, включаючи текстові чати, відео та аудіо консультації та інші форми взаємодії, що забезпечує більшу гнучкість у виборі та користуванні послугами [18].

Персоналізований підхід визначається здатністю багатьох систем адаптувати свою роботу до конкретних потреб та вподобань кожного користувача. Це створює унікальні умови для надання ефективної допомоги відповідно до індивідуальних особливостей кожного клієнта [19].

Інтеграція з іншими технологіями є ще однією характеристикою, що впливає на функціональність сучасних систем психологічної допомоги. Можливість взаємодії із мобільними додатками, системами штучного інтелекту та іншими передовими технологіями сприяє підвищенню ефективності процесу надання допомоги.

Наведені переваги сучасних систем психологічної допомоги свідчать про їхню велику потенційну користь для різних категорій користувачів. Однак, разом із цим, необхідно враховувати й низку недоліків, які можуть впливати на їхню загальну ефективність та придатність у практиці.

Одним з недоліків є обмежений емоційний зв'язок, який може виникнути в результаті онлайн-спілкування між терапевтом і клієнтом. Важливість емоційного взаєморозуміння може бути недооцінена в деяких формах психотерапії.

Питання конфіденційності та безпеки даних становлять серйозну проблему для сучасних систем. Зберігання та передача персональних даних через Інтернет завжди супроводжується ризиками конфіденційності та безпеки [20].

Технічні проблеми, такі як непередбачуване з'єднання або апаратні проблеми, можуть перешкоджати ефективному спілкуванню в онлайн-середовищі. Важливо враховувати, що для складних випадків психологічних станів може бути важливою або навіть неприйнятною формою допомоги. Онлайн-підтримка може виявитися недостатньою в таких випадках. Відсутність особистого контакту може стати значущим недоліком для деяких методів терапії та для тих клієнтів, які надають перевагу традиційним візитам і особистому спілкуванню [21].

В останні роки спостерігається зростання кількості психологічних сервісів, що пропонують психологічну підтримку онлайн, таких як BetterHelp, SimplePractice, Talkspace, TherapyNotes та інші. Проаналізувавши

функціональність, зручність використання, безпеку даних, можливості інтеграції та підтримку різних терапій в існуючих системах, можна зрозуміти сильні та слабкі сторони кожної з них і використати ці знання для вдосконалення нової системи. На фоні різних інших інформаційних систем можна виділити ряд характеристик. Наприклад, TherapyNotes визначається високою функціональністю, зокрема у сфері управління документацією та записами пацієнтів, і володіє зручним інтерфейсом, а також високим рівнем безпеки даних. З іншого боку, система має свої обмеження в інтеграції з іншими платформами. На рисунку 1.1 представлено зображення інформаційної системи TherapyNotes.

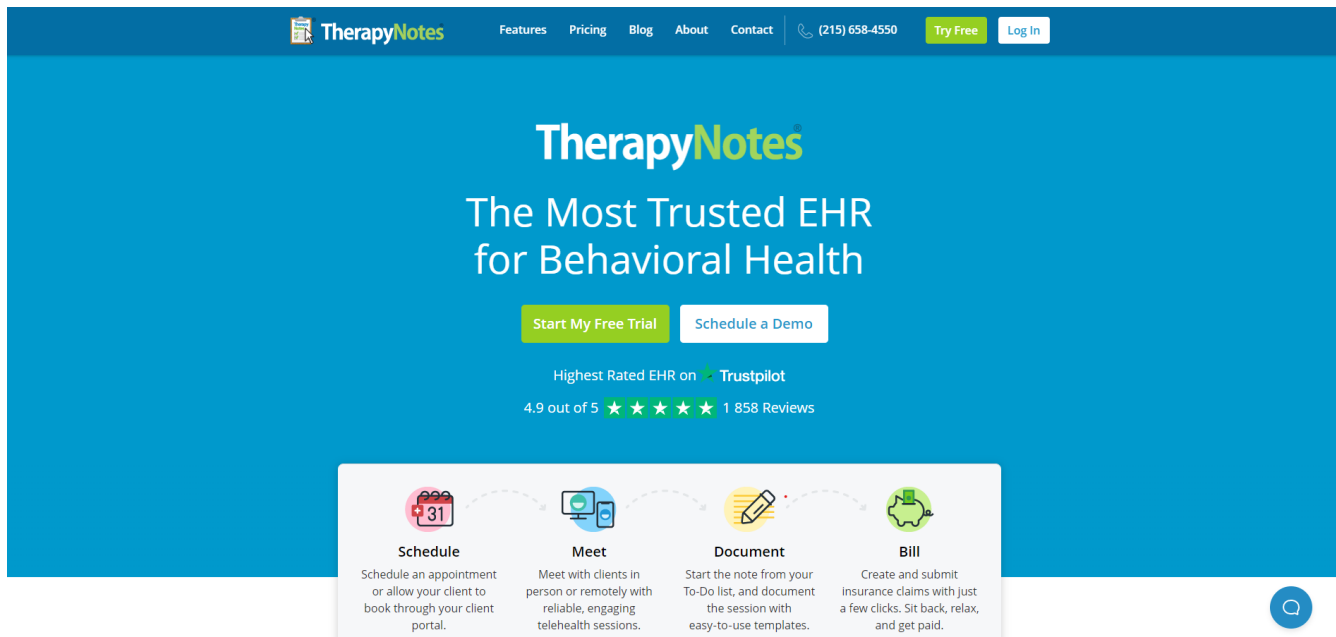


Рисунок 1.1 – Головна сторінка сайту TherapyNotes

Джерело: зображення екрану

У SimplePractice (рис.1.2), зручність користування є ключовою перевагою, оскільки інтерфейс її програми інтуїтивно зрозумілий. Однак, порівняно з іншими системами, вона має меншу функціональність ніж інші інформаційні системи.

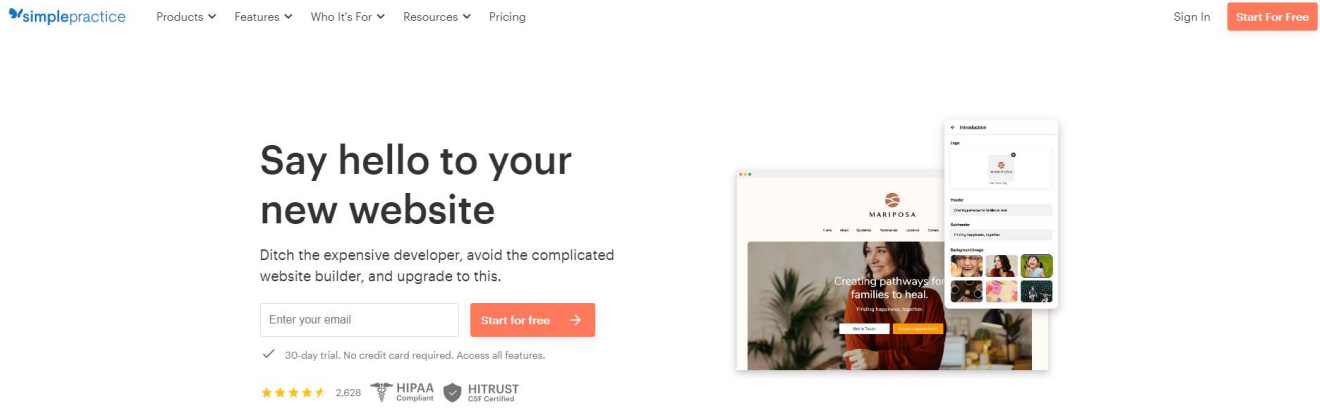


Рисунок 1.2 – Головна сторінка сайту SimplePractice

Джерело: зображення екрану

Talkspace спеціалізується на наданні онлайн-терапії та відзначається високим рівнем функціональності, особливо щодо підтримки різних форм онлайн-консультацій. Водночас інтерфейс може бути не дуже зручним для деяких користувачів і має вікові обмеження. Одним з основних недоліків цієї інформаційної системи є складність пошуку терапевта, з яким ви почуваетесь комфортно. Приклад інформаційної системи Talkspace зображений на рисунку 3.

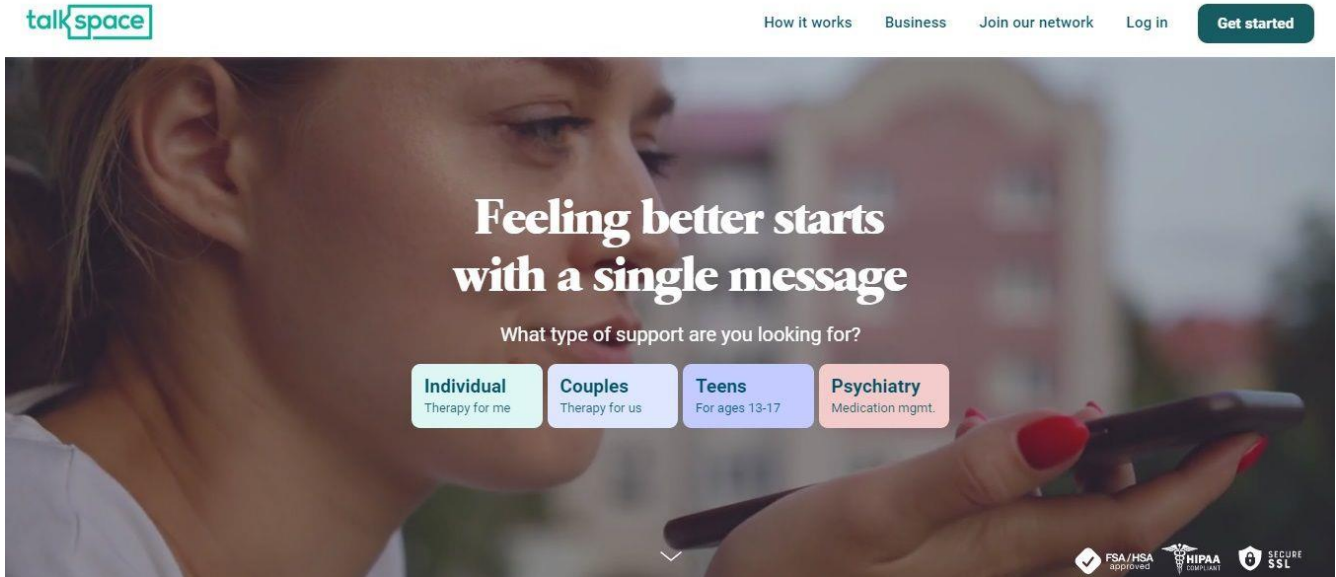


Рисунок 1.3– Головна сторінка сайту Talkspace

Джерело: зображення екрану

BetterHelp, яка фокусується на доступності психологічної допомоги через Інтернет, пропонує широкий спектр функцій. Однак можливі недоліки з точки зору зручності використання та інтеграції з іншими системами можуть вплинути на її привабливість для користувачів. До переваг можна віднести той факт, що BetterHelp має велику мережу ліцензованих терапевтів. Приклад інформаційної системи BetterHelp показано на рисунку 1.4.

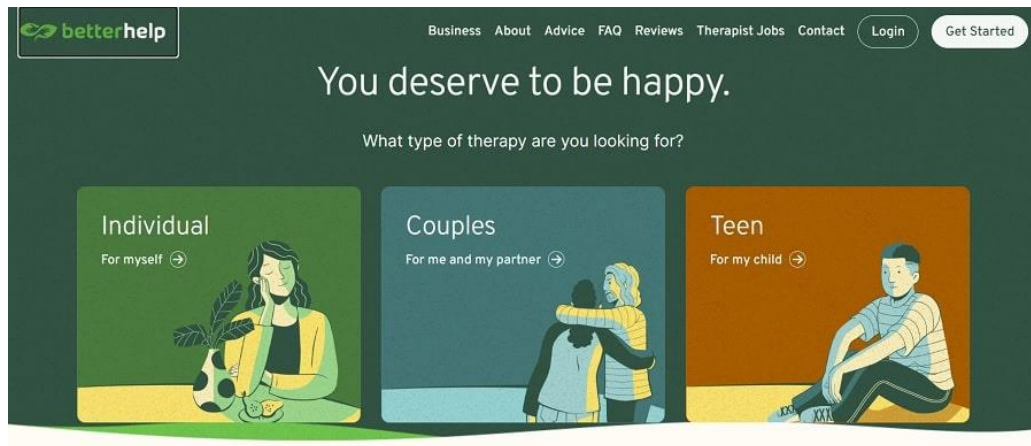


Рисунок 1.4 – Головна сторінка сайту BetterHelp

Джерело: зображення екрану

Одна із останніх випущених інформаційних систем, яка широко набирає популярності на території України є Mindly (рис.1.5). Це інформаційна платформа, яка фокусується на зміцненні здоров'я та благополуччя шляхом надання послуг фахівцям з психічного здоров'я та їхнім клієнтам у формі онлайн-маркетплейсу. Використовуючи сучасні технології, Mindly прагне зробити психологічні послуги доступними для всіх верств населення. Інформаційна система сприяє зручній взаємодії між психотерапевтами та їхніми клієнтами, надаючи можливість планувати консультації, обмінюватися повідомленнями та забезпечувати конфіденційну взаємодію.

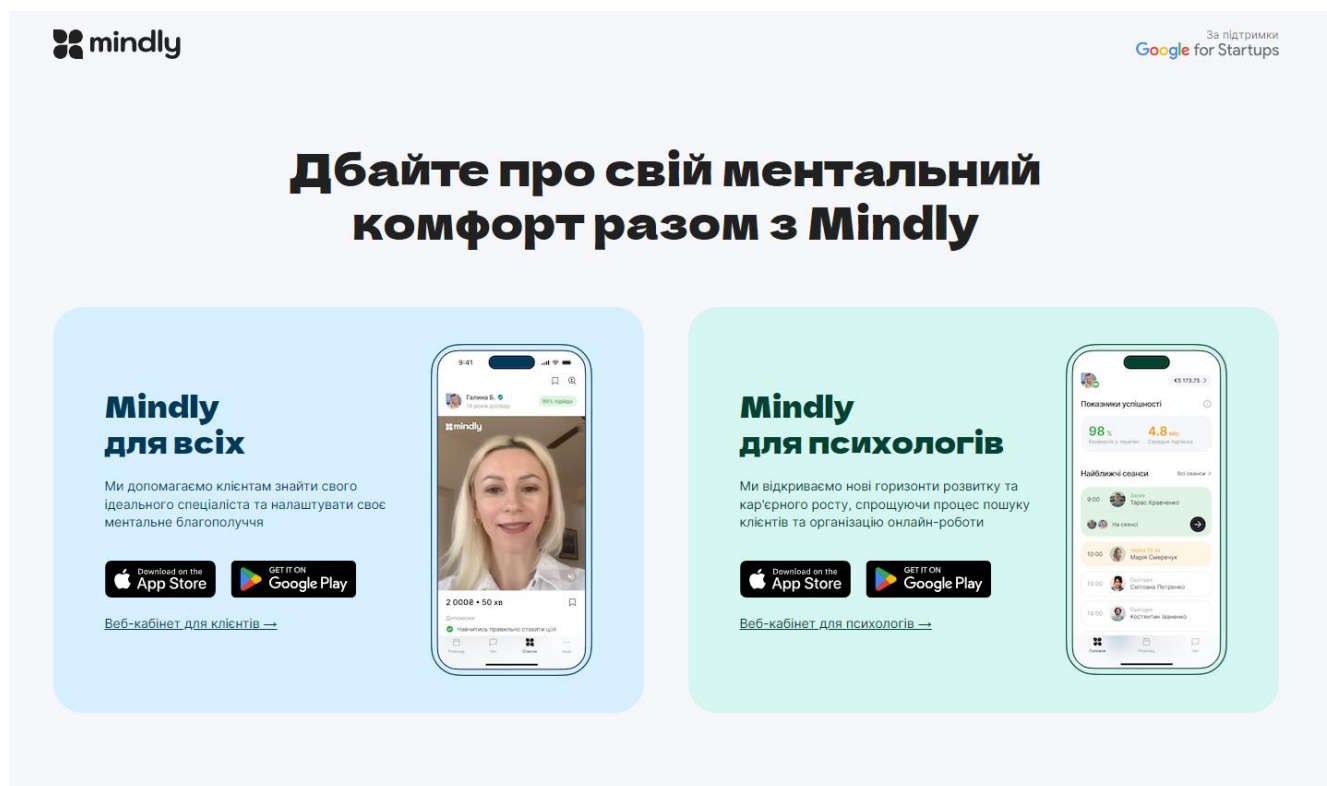


Рисунок 1.5 – Головна сторінка сайту Mindly

Джерело: зображення екрану

Функціональність і зручність використання – два ключові параметри для оцінки ІТ-систем. Хороша система повинна бути не лише багатofункціональною, але й простою у використанні. Наприклад, TherapyNotes і Talkspace є дуже функціональними, але можуть мати різний рівень зручності використання. SimplePractice, з іншого боку, робить сильний акцент на інтуїтивно зрозумілому дизайні, що допомагає зробити систему більш широко використовуваною для професіоналів з різним технічним досвідом.

Безпека даних має вирішальне значення для систем, які обробляють конфіденційну інформацію. Усі проаналізовані системи надають великого значення захисту персональних даних користувачів. Однак рівень інтеграції цих систем з іншими медичними або адміністративними додатками різниться. Інтеграція важлива для створення єдиного інформаційного простору, який полегшує обмін даними між різними учасниками медичного процесу.

Сучасні інформаційні системи повинні підтримувати різні терапевтичні підходи. Наприклад, Talkspace і BetterHelp зосереджені на онлайн-терапії, що особливо актуально з огляду на зростаючу популярність дистанційної допомоги. TherapyNotes і SimplePractice, з іншого боку, пропонують більш традиційний підхід, включаючи ведення нотаток, планування сеансів тощо.

Загальна оцінка кожної системи базується на збалансуванні всіх вищезгаданих факторів. Важливо, щоб нова інформаційна система взяла до уваги як сильні, так і слабкі сторони існуючих рішень, а також запропонувала інновації, які б задовольнили вимоги сучасного ринку психологічних послуг. Під час проведення аналізу наявних інформаційних систем були піддані оцінці основні фактори за шкалою від 1 до 10, як вказано в таблиці 1.2.

Таблиця 1.2 – Оцінка інформаційних систем за основними факторами

Категорія	TherapyNotes	SimplePractice	Talkspace	BetterHelp	Mindly
Функціональність	8	7	9	6	5
Зручність користування	7	8	6	7	6
Безпека даних	9	7	8	8	7
Інтеграція	6	7	8	5	4
Підтримка різних терапій	8	6	9	7	5
Загальна оцінка	7.6	7.0	8.0	6.6	5.4

Джерело: власна розробка

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Мета та задачі дослідження

Розробка інформаційної системи для підтримки надання психологічної допомоги спрямована на вирішення важливих завдань, які є критичними для ефективної роботи психологів і задоволення потреб пацієнтів. Система повинна забезпечувати реєстрацію пацієнтів та зберігання їхніх персональних та медичних даних, включаючи інформацію про стан здоров'я, історію психологічних консультацій та результати терапевтичних сесій, з урахуванням конфіденційності та безпеки цих даних. Також система має включати функціонал для оцінки та аналізу психологічного стану пацієнтів, з можливістю інтеграції інструментів для проведення психологічних тестів та опитувань, а також аналітичних алгоритмів для інтерпретації отриманих даних.

Система також має бути гнучкою в підтримці різноманітних терапевтичних підходів, включаючи традиційні та сучасні методи, такі як онлайн-консультації та групові сесії, а також інтеграцію з мобільними додатками для самостійної роботи пацієнтів. Додаткові функції системи включають створення звітів для психологів, надання рекомендацій і вказівок щодо підходів до лікування та забезпечення взаємодії між психологами і пацієнтами за допомогою інтегрованих інструментів комунікації.

До завдань дослідження входить розробка концептуальної моделі системи з детальним описом функцій та можливостей, включаючи збір та аналіз даних про пацієнтів та підтримку різних форм терапії. Важливо також проаналізувати потреби користувачів, щоб розробити вимоги до системи та оцінити сучасні інформаційні технології для її реалізації. Завдання дослідження включають деталізацію функціональних вимог, розробку архітектури системи, аналіз методів захисту даних, вивчення інтеграції з іншими системами, створення прототипу та тестування

системи, а також аналіз правових та етичних аспектів. Все це складає основу дисертації та надає напрямок для розробки ефективної інформаційної системи, яка відповідає потребам психологічної допомоги.

Інформаційна система, що надає психологічну підтримку, обслуговує різні категорії користувачів, яких можна розділити на клієнтів і психотерапевтів. Кожна з цих категорій має чітко визначені ролі та цілі в межах системи. Користувачі – це особи, які безпосередньо взаємодіють з інформаційною системою для отримання або надання психологічної підтримки. Ця категорія включає в себе:

- клієнти: особи, які потребують психологічного оцінювання, терапії, кризового втручання або самостійного опрацювання матеріалів;
- психотерапевти: кваліфіковані особи, що надають професійні послуги в рамках системи.

В рамках кваліфікаційної роботи буде розглянуто розробку підсистеми, орієнтованої на обслуговування клієнтів, в контексті інформаційної системи, що підтримує надання психологічної підтримки.

Реєстрація та профілювання користувачів є важливими функціональними аспектами. Система повинна дозволяти реєструвати нових користувачів, забезпечуючи високий рівень конфіденційності та безпеки персональних даних. Крім того, важливо створювати та ефективно управляти особистими профілями клієнтів для визначення їхніх потреб та надання персоналізованої підтримки. Психологічне оцінювання представляє інший значущий аспект функціоналу. Система повинна надавати інструменти для проведення психологічного оцінювання, щоб визначити особливості та потреби клієнтів. Аналіз результатів оцінювання дасть змогу розробити індивідуальну стратегію психологічної підтримки.

Ефективна терапевтична взаємодія є ключовим елементом системи. Це включає реалізацію засобів для забезпечення онлайн взаємодії між клієнтами та фахівцями з психічного здоров'я, зокрема чату та відеоконференцій. Зберігання та обмін конфіденційною інформацією між сторонами грає важливу роль у цьому

контексті.

Електронний журнал терапевтичних сесій та його аналіз є ще однією важливою функціональністю. Записи терапевтичних сесій надають можливість ведення історії та визначення прогресу клієнтів.

Всі вищезазначені аспекти функціональності працюють разом для створення інтегрованої системи надання психологічної допомоги з урахуванням індивідуальних потреб та вимог різних користувачів. Тому в ході кваліфікаційної роботи буде проведено комплексний аналіз та спроектовано клієнтські підсистеми для інформаційної системи підтримки надання психологічної допомоги, з акцентом на її розвиток, оптимізацію та відповідність сучасним стандартам.

2.2 Технології та інструменти реалізації підсистеми клієнта

Для інформаційної системи підтримки надання психологічної допомоги вибір технологій програмування передбачає визначення мов програмування, фреймворків, баз даних, інструментів розробки та інфраструктури, які найкраще відповідають нашим цілям і вимогам. У сучасному світі програмування вибір правильних технологій та інструментів для проекту є важливим аспектом. Для створення інформаційної системи використовується одні із найпопулярніших та найуспішніших технологій у цій сфері є C#, React, Azure, MSSQL та технології .NET Core та Entity Framework.

C# – це об'єктно-орієнтована мова програмування, розроблена Microsoft, яка широко використовується для створення масштабних та надійних додатків. Вона забезпечує потужні можливості для розробки, зокрема, строгу типізацію, автоматичне управління пам'яттю та широку підтримку бібліотек [22]. Це робить C# ідеальним вибором для бекенд-розробки у комплексних проектах.

.NET Core є відкритою та кросплатформенною платформою розробки, розробленою корпорацією Microsoft, яка надає широкий інструментарій для

створення різноманітних програмних додатків. Вибір .NET Core обґрунтовується його кросплатформенністю та можливістю розгортання на різних платформах, включаючи Microsoft Azure. До переваг також відноситься його легка інтеграція із об'єктно-реляційною технологією доступу до даних Entity Framework [23].

Цей фреймворк надає розробникам зручний і високорівневий інтерфейс для взаємодії з базами даних, де структура об'єктів в програмі відображається на структурі таблиць у базі даних. Завдяки Entity Framework розробники можуть працювати з даними, використовуючи об'єктно-орієнтований підхід, що полегшує розробку та забезпечує абстракцію від деталей роботи з базою даних. Також перевагою Entity Framework є використання запитів LINQ для роботи із даними [24].

React з іншого боку, є популярною JavaScript-бібліотекою для побудови інтерфейсів користувача. Його головна перевага – це можливість створювати великі веб-додатки, які можуть оновлювати дані без перезавантаження сторінки. React дозволяє розробникам ефективно управляти станами веб-додатків, забезпечуючи гнучкість та високу продуктивність [25, 26].

Azure від Microsoft – це хмарна платформа, яка надає широкий спектр обчислювальних, зберігаючих, мережевих та інших сервісів. Вона дозволяє розробникам швидко розгортати та управляти додатками в хмарі, забезпечуючи високу доступність та масштабованість [27]. Azure інтегрується з багатьма іншими технологіями, включаючи C# та MSSQL, що робить його відмінним вибором для комплексних рішень.

MSSQL – це система управління базами даних від Microsoft, яка використовується для забезпечення надійного зберігання, обробки та управління даними. Основною характеристикою MSSQL є її масштабованість та висока продуктивність. Система здатна обробляти великі обсяги транзакцій та запитів, забезпечуючи при цьому високу швидкість обробки та стабільність [28]. Це робить MSSQL ідеальним рішенням для застосунків, де необхідна надійність та ефективність обробки даних.

Використання цих технологій та інструментів в проекті дає можливість реалізувати складні та високопродуктивні рішення. Наприклад, використання C# для бекенду забезпечує міцну та надійну основу, в той час як React на фронтенді дозволяє створити інтуїтивно зрозумілий та зручний інтерфейс користувача. Така комбінація технологій дозволяє розробникам оптимізувати процес розробки та забезпечити високу якість кінцевого продукту.

Microsoft Azure, як хмарна платформа, відіграє вирішальну роль у сучасній архітектурі програмного забезпечення. Ця платформа надає широкий спектр хмарних сервісів, включаючи обчислення, аналітику, зберігання та багато іншого, що дозволяє розробникам створювати, тестувати, запускати та управляти додатками у хмарному середовищі.

Однією з ключових переваг Azure є його масштабованість. Хмарні сервіси Azure можуть легко нарощувати або зменшувати обчислювальні ресурси відповідно до потреб користувача, що забезпечує високу гнучкість та економічну ефективність. Також Azure пропонує різноманітність сервісів з безпеки, що є важливим аспектом при роботі з конфіденційними даними. MSSQL, у свою чергу, є потужним інструментом для управління даними. Його використання забезпечує високу швидкість обробки запитів та здатність ефективно працювати з великими обсягами даних. Це особливо важливо для додатків, що збирають та обробляють велику кількість інформації.

Далі необхідно обрати зручне середовище розробки. Наразі одними з найпопулярніших та найзручніших є Visual Studio та Visual Studio Code.

Visual Studio, розроблена компанією Microsoft, є інтегрованим середовищем розробки (IDE), яке підтримує різні мови програмування та інструменти розробки програмного забезпечення [29]. Це ключовий компонент у розробці програмного забезпечення, що надає розробникам єдину платформу для написання, тестування та налагодження коду. Однією з переваг Visual Studio є її гнучкість і масштабованість, що дозволяє розробникам працювати з різними мовами програмування, такими як C#, C++, Python, JavaScript та багатьма іншими. Це

робить її універсальним інструментом для розробників, що сприяє технічній гнучкості та інноваціям.

Окрім Visual Studio варто зазначити також використання Visual Studio Code, яке є легковаговим та потужним редактором коду від Microsoft. Visual Studio Code відрізняється від Visual Studio, пропонуючи більшу гнучкість та ширший спектр можливостей для розробників.

Для проектування графічної інтерфейсу інформаційної системи, таких як сторінки, форми, діалогові вікна та елементи керування, використовується інструмент Figma

Комбінування цих технологій та інструментів у проекті дозволяє створити сильну, безпечну та гнучку систему, яка може ефективно вирішувати поставлені бізнес-завдання. Такий інтегрований підхід в розробці не лише сприяє створенню якісного продукту, але й значно підвищує швидкість розробки та знижує витрати на підтримку та розвиток проекту.

3 ПРОЕКТУВАННЯ ПІДСИСТЕМИ КЛІЄНТА

3.1 Структурно-функціональне моделювання процесу

Для даної кваліфікаційної роботи чітко визначено ціль – розробити підсистему клієнта для Інформаційної системи підтримки надання психологічної допомоги. У реалізації цього завдання передбачено використання вхідних та вихідних даних. Вхідними даними є інформація про емоційний стан користувача. Результатом роботи є скорегований емоційний стан та завдання щодо підвищення емоційної стабільності. Механізми включають клієнта, розробника, психотерапевта, програмне та апаратне забезпечення у вигляді веб-серверу та бази даних, а також стороні системи. Відповідна контекстна діаграма в нотації IDEF0 з точки зору клієнта представлена на рисунку 3.1.

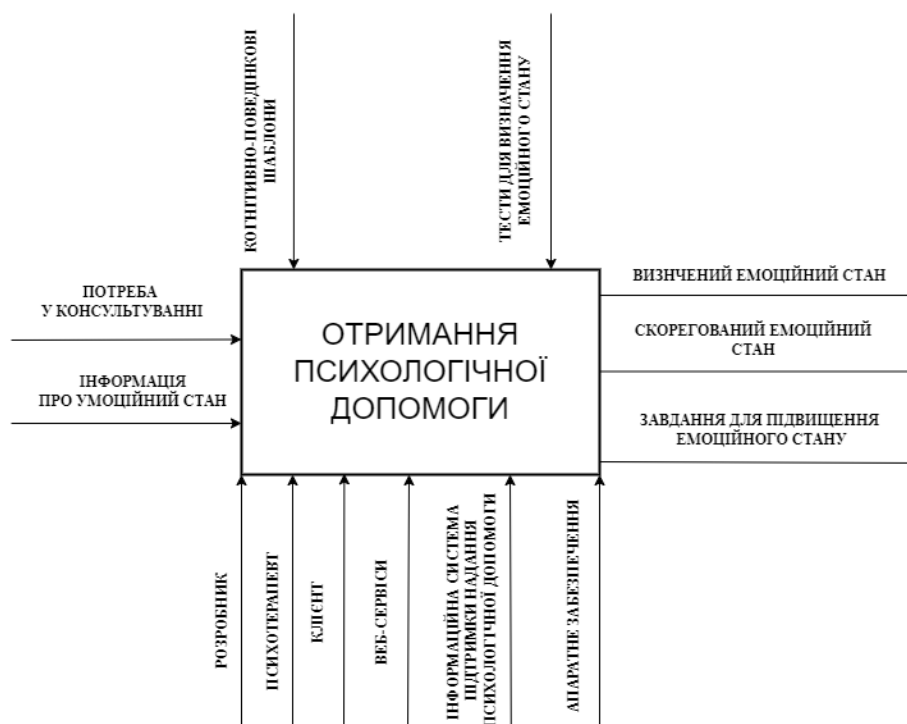


Рисунок 3.1 – Контекстна діаграма інформаційної системи процесу отримання психологічної допомоги

Джерело: власна розробка

Далі розглянемо діаграму декомпозиції IDEF0 для нашої інформаційної системи (рис. 3.2).

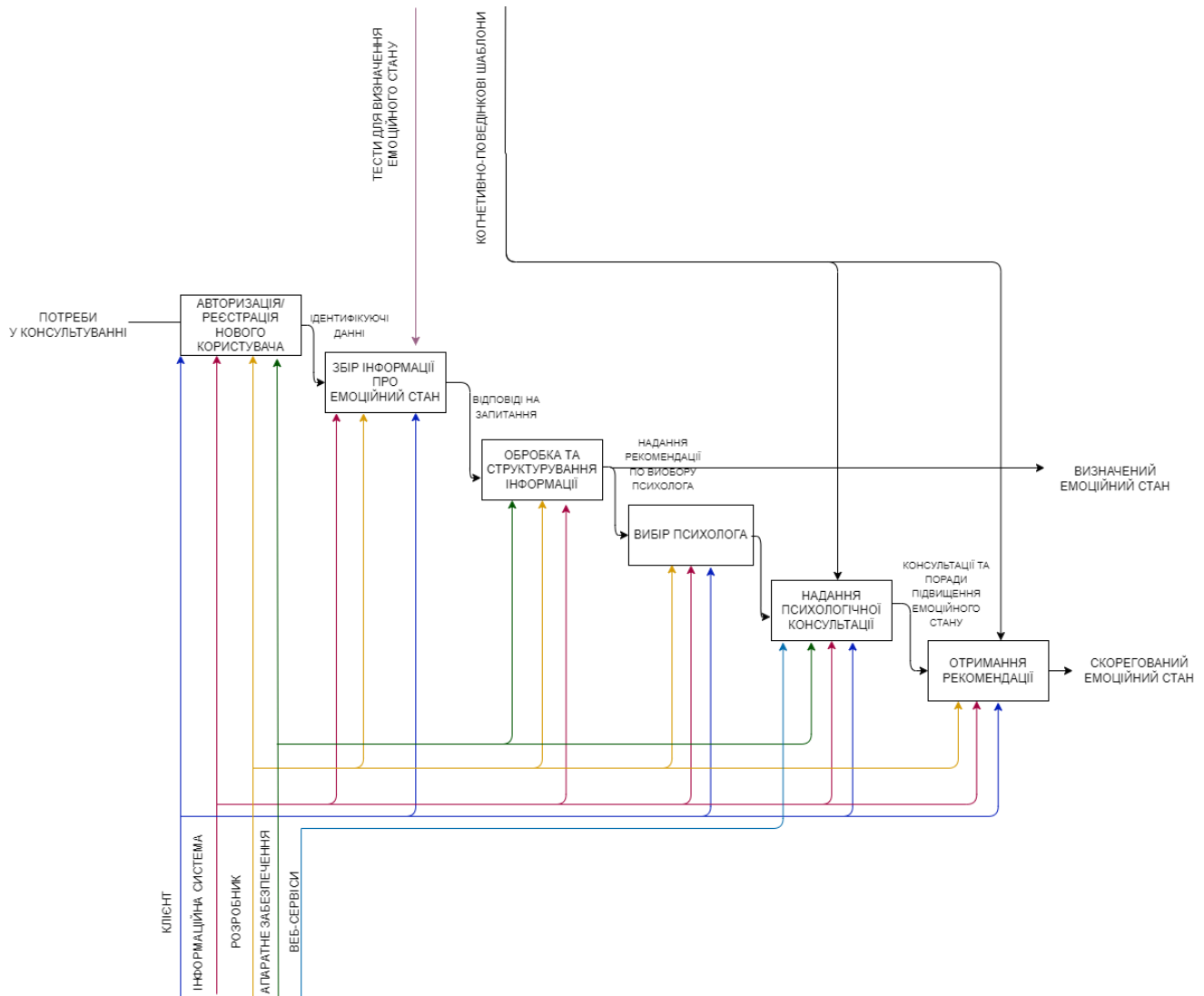


Рисунок 3.2 – Декомпозиція контекстної діаграми процесу отримання психологічної допомоги

Джерело: власна розробка

Ця декомпозиція контекстної діаграми відображає головні функції системи та їх взаємозв'язки. Усі функції розташовані у вигляді блоків, а стрілки вказують на напрямок потоку інформації. Вона може слугувати основою для подальшого

детального моделювання та проектування інформаційної системи [30, 31].

3.2 Моделювання діаграми варіантів використання

Після створення контекстних діаграм, необхідно розробити діаграму варіантів використання. Ця діаграма визначає, який функціонал програмної системи, що розробляється, доступний для кожної групи користувачів.

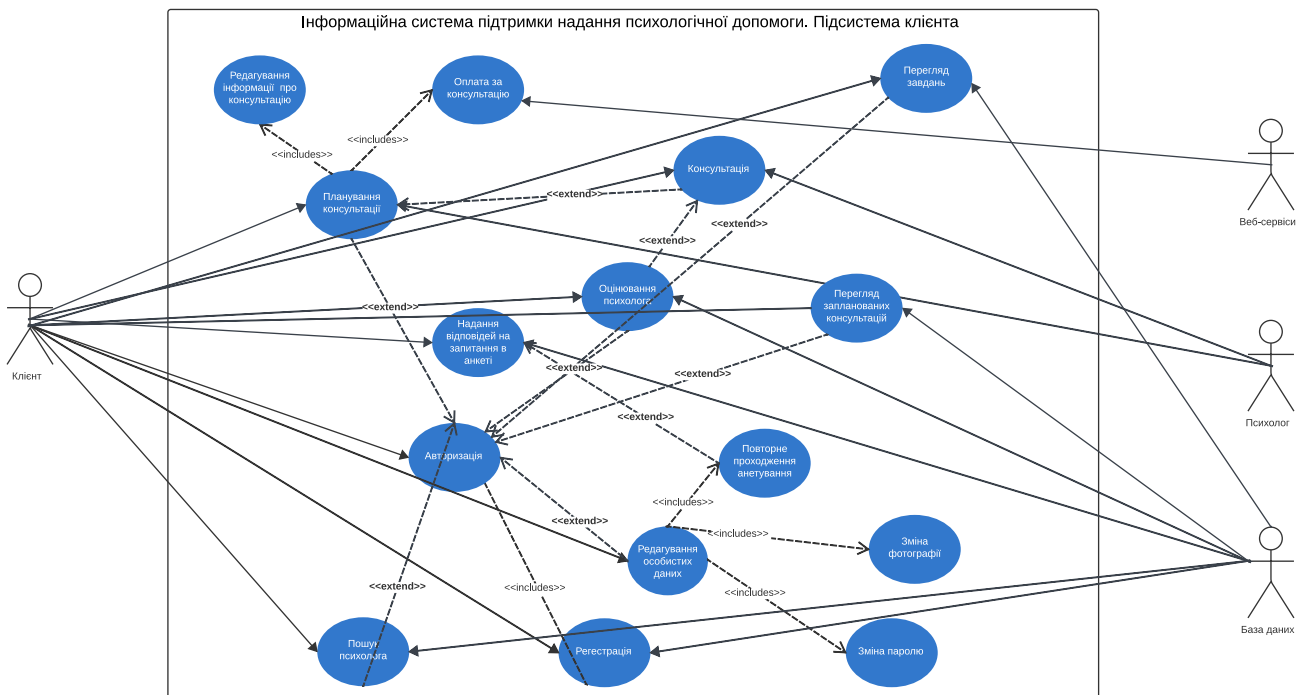


Рисунок 3.3 – Діаграма варіантів використання інформаційної системи

Джерело: зображення екрану

На цій діаграмі зображено взаємодію між користувачами та інформаційною системою. Вона охоплює різноманітні варіанти взаємодії з системою користувачів (рис. 3.3). Подробиці про користувачів та варіанти використання можна знайти в таблицях 3.1 та 3.2, відображених нижче.

На цій діаграмі зображено взаємодію між різними користувачами та інформаційною системою. Вона охоплює різноманітні сценарії для клієнтів (рис. 3.3). Подробиці про акторів та сценарії використання можна знайти в таблицях 3.1 та 3.2, відображених нижче.

Таблиця 3.1. – Опис користувачів

№	Назва користувача	Детальний опис
1	Клієнт (К)	Основний користувач системи, який входить, виходить і взаємодіє з іншими користувачами з метою отримання необхідної інформації або виконання певних дій в інформаційній системі.
2	Психотерапевт (П)	Взаємодіє з системою з метою надання психологічної підтримки або отримання інформації щодо користувача. Може аналізувати введену користувачами інформацію.
3	База даних (БД)	Зберігає та управляє інформацією, необхідною для роботи системи. Це можуть бути дані про користувачів та інша інформація.
4	Стороні системи (СС)	Забезпечує взаємодію між різними частинами системи через мережу Інтернет.

Далі наведено короткий огляд основних функцій, що допоможе краще розуміти їхню сутність та можливості (табл. 3.2).

Таблиця 3.2. – Опис варіантів використання

№	Варіанти використання	Детальний опис	Користувач
1	Авторизація	Механізм входу для користувачів, який забезпечує безпеку та ідентифікацію осіб.	К
2	Реєстрація	Можливість створення нового облікового запису користувача з введенням необхідних особистих даних.	К/БД
	Консультація	Модуль, що надає можливість взаємодії користувачів із психологами, включаючи текстовий, аудіо або відео-комунікацію.	П/К/БД/СС
4	Перегляд завдань	Функціонал, який дозволяє користувачам переглядати свої завдання або дії, пов'язані з консультаціями. Перегляд запланованих консультацій:	К/БД
5	Перегляд запланованих консультацій	Інформація про надходження майбутніх консультацій, їх дати та час.	К/БД
6	Оплата за консультацію	Вбудована система оплати для забезпечення фінансових транзакцій між користувачами і психологами.	К/СС

Продовження таблиці 3.2

7	Планування консультації	Можливість визначення дат та часу для майбутніх консультацій.	К/П
8	Редагування інформації про консультацію	Можливість зміни деталей консультацій, таких як час, дата або тип взаємодії.	К/БД
9	Надання відповідей на запитання в анкеті	Модуль для відповіді на питання анкети для отримання більш детальної інформації від користувачів.	К
10	Оцінювання психолога	Функція для користувачів виражати своє задоволення або незадоволення від консультацій та взаємодії з психологом.	К/БД
11	Пошук психолога	Можливість знаходження психологів за різними критеріями, такими як спеціалізація чи рейтинг.	К
12	Зміна паролю	Функціонал для зміни паролю користувача для забезпечення безпеки облікового запису.	К/БД
13	Зміна фотографії	Можливість користувачів змінювати своє фото в профілі.	К/БД
14	Повторне проходження анкетування	Можливість користувачів переходити або оновлювати свою анкету.	К/БД

Джерело: власна розробка

3.3 Проектування моделі бази даних

Розробка ефективної інформаційної системи психологічної допомоги вимагає ретельного моделювання бази даних для забезпечення надійності та ефективності обробки інформації. У цьому розділі представлено концептуальний огляд та модель відносин сутність-зв'язок (ERD) бази даних, що використовується для забезпечення функціональності інформаційної системи.

Розробка ефективної інформаційної системи підтримки надання психологічної допомоги вимагає детального моделювання бази даних для забезпечення надійності та ефективності обробки інформації. Ця концептуальна модель виступає основою для подальшого створення фізичної бази даних. Одним із ключових інструментів для візуалізації цього відображення є модель сутність-зв'язок [32].

Це графічне відображення структури даних, де сутності представлені таблицями, а взаємозв'язки - стрілками, які вказують на зв'язки між сутностями. Використання ER-діаграми дозволяє розробникам, аналітикам та іншим учасникам проекту легко спілкуватися і розуміти основні аспекти бази даних. На рисунку 3.4 представлена ER-діаграма бази даних для інформаційної системи поставленої в завданні.

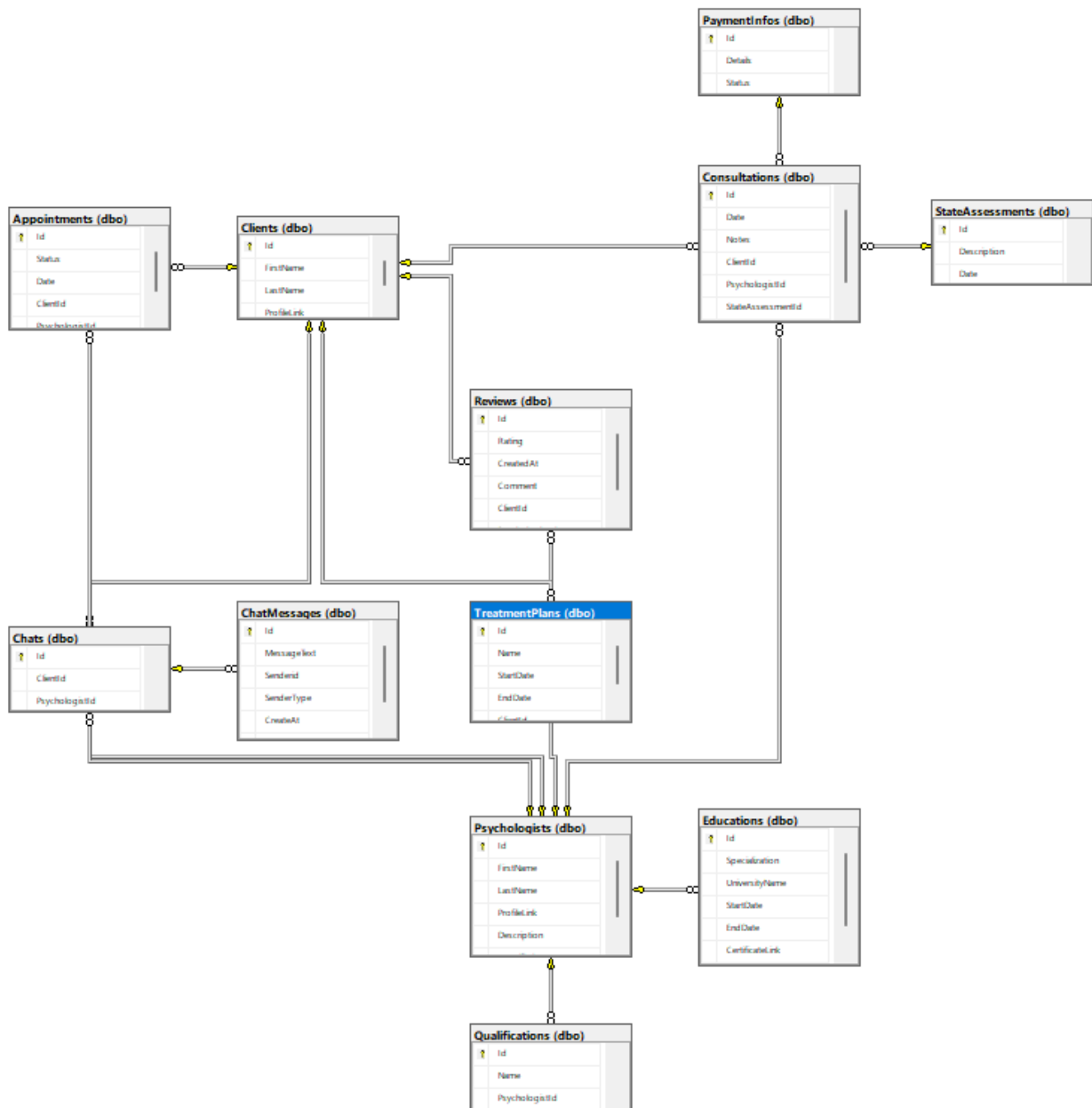


Рисунок 3.4 – ER-діаграма бази даних інформаційної системи

Джерело: зображення екрану

У таблиці 3.3 наводимо детальний опис ER-діаграми, де ми описуємо структуру таблиці, як типи полів і первинні і вторинні ключі, які відображають зв'язок між таблицями.

Таблиця 3.3 – Опис таблиць для ER-діаграми

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
1	Appointments	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Status	Поточний стан	int		Не порожній
		Date	Дата створення	datetime2(7)		Не порожній
		ClientID	Ідентифікатор клієнта	uniqueidentifier	FK	Не порожній
		PsychologistId	Ідентифікатор психолога	uniqueidentifier	FK	Не порожній
2	Clients	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		FirstName	Ім'я особи	nvarchar(MAX)		Не порожній
		LastName	Прізвище особи	nvarchar(MAX)		Не порожній
		ProfileLink	Посилання на профіль чи сторінку	nvarchar(MAX)		
3	Reviews	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Rating	Оцінка рейтингу	int		Не порожній

Продовження таблиці 3.3

		CreatedAt	Дата створення	datetime2(7)		Не порожній
		ClientId	Ідентифікатор клієнта	uniqueidentifier	FK	Не порожній
		PsychologistId	Ідентифікатор психолога	uniqueidentifier	FK	Не порожній
4	Consultation	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Date	Дата проведення	datetime2(7)		Не порожній
		Notes	Додаткові деталі	nvarchar(MAX)		
		ClientId	Ідентифікатор клієнта	uniqueidentifier	FK	Не порожній
		PsychologistId	Ідентифікатор психолога	uniqueidentifier	FK	Не порожній
		StateAssessmentId	Ідентифікатор на рекомендації	uniqueidentifier	FK	Не порожній

Продовження таблиці 3.3

		PaymentInfoId	Ідентифікатор транзакції на оплату	Uniqueidentifier	FK	
5	StateAssesments	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Description	Назва категорії	nvarchar(MAX)		Не порожній
		Date	Дата створення	datetime2(7)		
6	PaymentInfos	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Details	Опис оплати	nvarchar(MAX)		
		Status	Статус оплати	int		Не порожній
7	TreatmentPlans	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Name	Назва плану	nvarchar(MAX)		Не порожній
		StartDate	Дата початку	datetime2(7)		Не порожній

Продовження таблиці 3.3

		EndDate	Дата завершення	datetime2(7)		Не порожній
		ClientId	Ідентифікатор клієнта	uniqueidentifier	FK	Не порожній
8	Psychologists	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		FirstName	Ім'я особи	nvarchar(MAX)		Не порожній
		LastName	Прізвище особи	nvarchar(MAX)		Не порожній
		ProfileLink	Посилання на профіль чи сторінку	nvarchar(MAX)		
		Description	Текстовий опис	nvarchar(MAX)		Не порожній
		IsVerified	Позначка, що підтверджує або відхиляє перевірки статус	bit		Не порожній

Продовження таблиці 3.3

9	Qualifications	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Name	Назва кваліфікації	nvarchar(MAX)		Не порожній
		PsychologistId	Ідентифікатор психолога	uniqueidentifier	FK	
10	Educations	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		Specialization	Опис спеціалізації	nvarchar(MAX)		Не порожній
		UniversityName	Назва	nvarchar(MAX)		Не порожній
		StartDate	Дата початку	datetime2(7)		Не порожній
		EndDate	Дата завершення	datetime2(7)		
		CertificateLink	Посилання на сертифікат	nvarchar(MAX)		

Продовження таблиці 3.3

		PsychologistId	Ідентифікатор психолога	uniqueidentifier	FK	Не порожній
		Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		ClientID	Ідентифікатор клієнта	uniqueidentifier	FK	Не порожній
		PsychologistId	Ідентифікатор психолога	uniqueidentifier	FK	Не порожній
12	ChatMessages	Id	Унікальний ідентифікатор	uniqueidentifier	PK	Не порожній
		MessageText	Текст повідомлення	nvarchar(MAX)		
		SenderId	Ідентифікатор клієнта чи психолога	uniqueidentifier		Не порожній
		SenderType	Тип відправника	int		Не порожній
		CreatedAt	Дата створення	datetime2(7)		Не порожній

Продовження таблиці 3.3

		ChatId	Ідентифікатор чату	uniqueidentifier	FK	Не порожній
--	--	--------	--------------------	------------------	----	----------------

Джерело: власна розробка

4 РОЗРОБКА ПІДСИСТЕМИ КЛІЄНТА

4.1 Розробка архітектури інформаційної системи

Архітектура інформаційної системи підтримки надання психологічної допомоги наведена на рисунку 4.1. На цьому рисунку показана взаємодія різних частин інформаційної системи між собою, а також із іншими сторонніми системами.

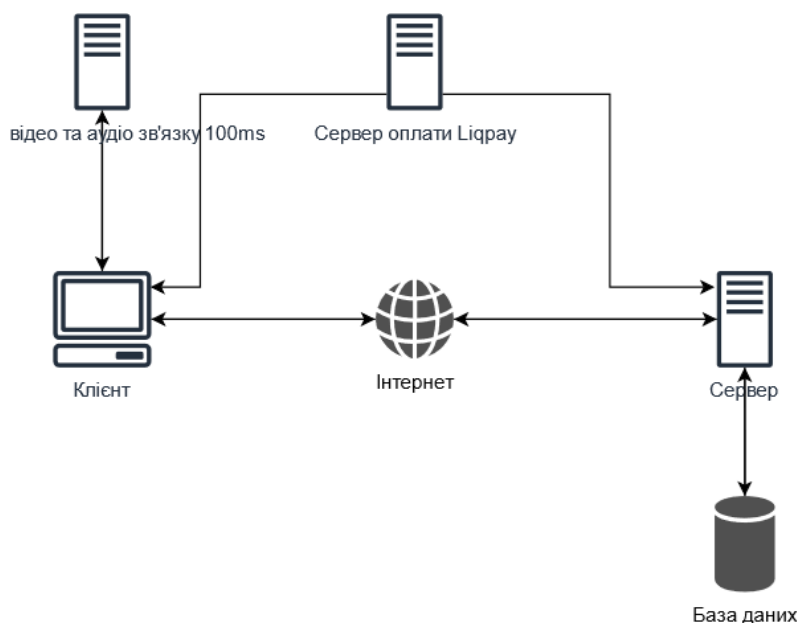


Рисунок 4.1 – Архітектура інформаційної системи

Джерело: зображення екрану

Структурно підсистема клієнта, складається із чотирьох основних компонентів: сервера, клієнта, бази даних та сторонні сервіси, а саме для фінансових транзакцій та для встановлення відео-аудіо зв'язку. В основу нашої інформаційної системи взято клієнт-серверну архітектуру. Архітектура клієнт-сервер є типом архітектури, яка формує основу комп'ютерної мережі [33, 34]. У цій системі може існувати кілька клієнтів, які є віддаленими процесорами і звертаються до централізованого сервера для отримання послуги. Клієнтські комп'ютери мають

інтерфейс, який дозволяє користувачам запитувати послуги від сервера та отримувати відповіді у зручному форматі. Сервери чекають на нові запити від клієнтів і надають їм відповіді. Загалом, сервер створює стандартизований та прозорий інтерфейс для клієнтів, щоб вони не мали необхідності розуміти будову системи та її особливості. Це полегшує взаємодію користувачів з сервером, оскільки вони можуть просто відправляти запити і отримувати відповідь, не задумуючись про технічні деталі [35].

В системі сервер виступає як центральний вузол взаємодії, призначений для обробки запитів, що надходять від клієнту, та взаємодії з базою даних та зовнішніми сервісами. До функцій сервера входить, реєстрація користувачів, керування консультаціями та відео-аудіо зв'язку під час них, а також обробку транзакцій від платіжного сервісу. Клієнт виступає інтерфейсом взаємодії для кінцевого користувача, забезпечуючи ефективну та зручну взаємодію з інформаційною системою. Цей компонент може існувати як у формі веб-додатка так і у формі мобільного додатка, який використовує сервер для обміну та зберігання необхідною інформацією.

Лідром відповідає за обробку фінансових транзакцій, тоді як 100ms забезпечує низьку затримку та високу якість відео-аудіо обміну. Використання цих сторонніх систем сприяє функціональній комплексності та спрощення процесу розробки.

База даних є фундаментальним компонентом для ефективного функціонування будь-якої інформаційної системи. Вона виступає в ролі центрального елемента архітектури, яка дає можливість системі ефективно обробляти, зберігати та витягати дані. Для ефективного зберігання, вибірку та оновлення даних, база даних визначається як реляційна, що свідчить про використання моделі даних, основаної на зв'язках між таблицями. Цей підхід до структуризації інформації надає інформаційній системі гнучкість, сприяє ефективному управлінню даними та масштабованості [36, 37].

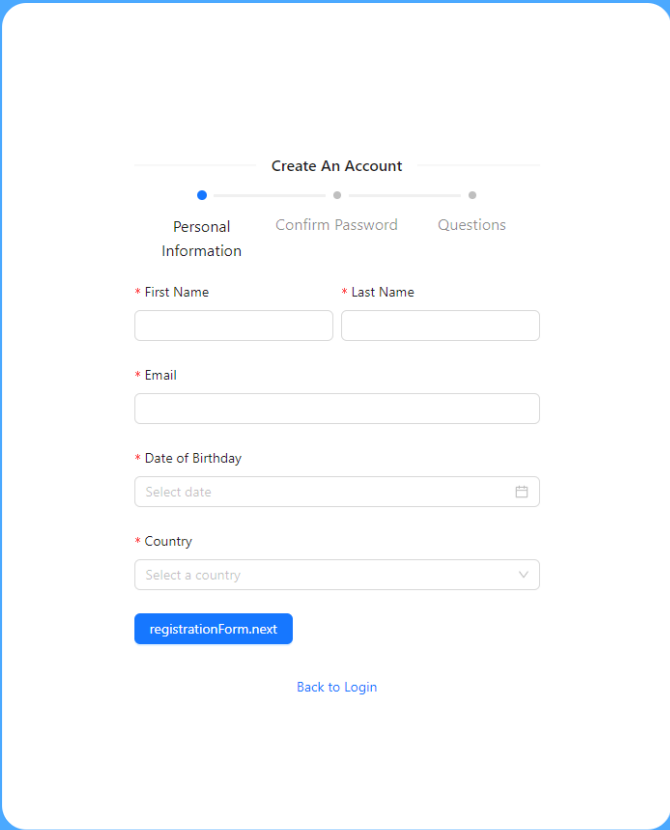
Описана вище архітектура інформаційної системи, володіє фундаментальною збалансованістю, яка ґрунтується на принципах клієнт-серверної архітектури,

надаючи ефективну взаємодію, гнучкість та легкість.

4.2 Розробка веб-додатку для підсистеми клієнта

Для розробки дизайну веб-додатку визначено використання фреймворку Ant Design як основного інструменту, який надає високоякісні та легко управляючі компоненти, сприяючи стабільності та однорідності користувацького інтерфейсу. Застосування цього інструменту дозволяє досягти не тільки ефективності розробки, але й забезпечити привабливий зовнішній вигляд додатку для кінцевого користувача. Відповідно до технологічного стеку було створено веб-додаток, і всі функціональні частини цього додатку, які сприяють його повноцінному та зручному функціонуванню, розглядаються нижче [38].

Процес взаємодії клієнту з додатком в першу чергу розпочинається із етапу аутентифікації або реєстрації, якщо такого не існує. Забезпечення аутентифікації та створення нового облікового запису на стороні веб-додатку виконується в кілька послідовних етапів. На першому етапі клієнт взаємодіє з інтерфейсом додатку ініціюючи процес реєстрації чи авторизації. У разі створення нового облікового запису користувач подає відповідні дані, такі як електронна пошта та пароль (рис. 4.2).



The image shows a registration form titled "Create An Account" with three steps: "Personal Information", "Confirm Password", and "Questions". The "Personal Information" step is currently active. The form includes the following fields:

- * First Name
- * Last Name
- * Email
- * Date of Birthday (with a "Select date" dropdown and a calendar icon)
- * Country (with a "Select a country" dropdown)

At the bottom of the form, there is a blue button labeled "registrationForm.next" and a link "Back to Login".

Рисунок 4.2 – Форма реєстрації клієнта у веб-додатку

Джерело: зображення екрану

Система перевіряє ці дані на відповідність вимогам безпеки та їх валідність та відправляє їх до серверу. Якщо введені дані відповідають критеріям, сервер створює новий обліковий запис. Щодо аутентифікації, при повторному вході в систему клієнту не потрібно вже реєструватися, він вводить свої дані, які перевіряються на валідність (рис. 4.3). У разі вірності введених даних він отримує доступ до особистого облікового запису.

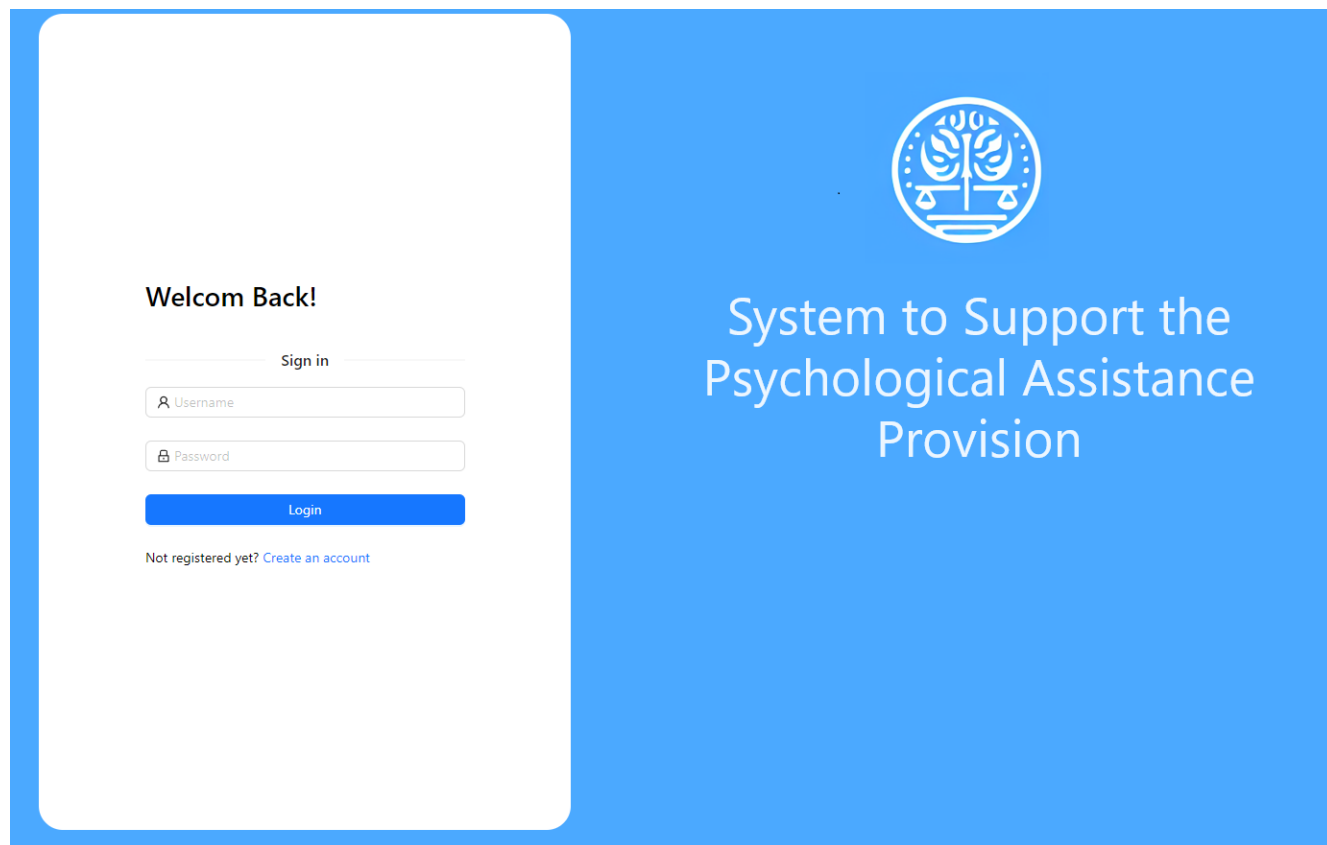


Рисунок 4.3 – Форма авторизації клієнта у веб-додатку

Джерело: зображення екрану

Після успішної аутентифікації або реєстрації клієнт взаємодіє із головним екраном, де реалізовано календар з відображенням усіх запланованих консультацій та можливістю записуватись на нові консультації., дозволяє користувачеві вибирати час і дату нового заходу (рис. 4.4).

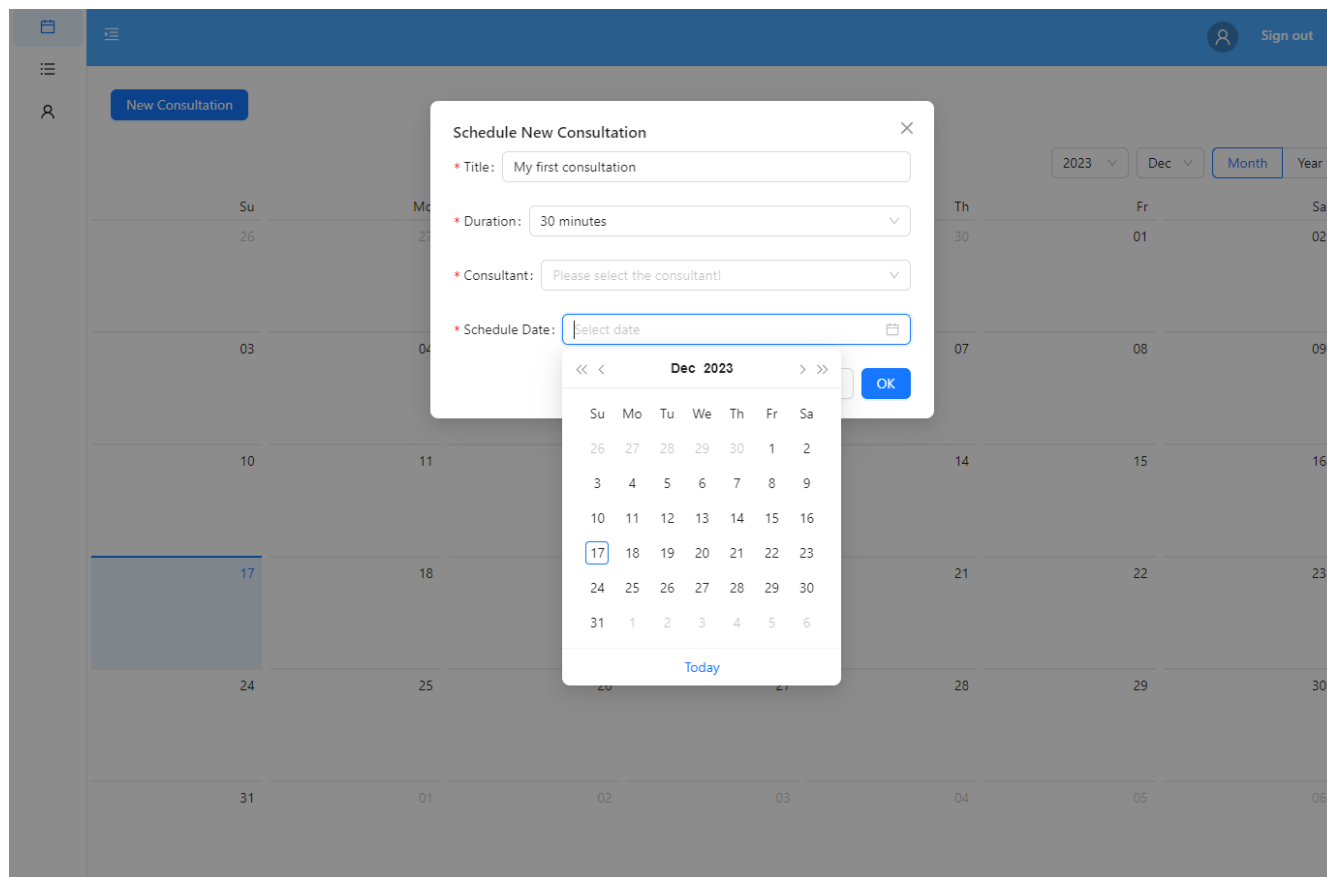


Рисунок 4.4 – Екран планування консультацій клієнта у веб-додатку
Джерело: зображення екрану

Пошук психотерапевтів для клієнта з метою уточнення та фільтрації доступних фахівців за різними критеріями є важливим функціоналом в інформаційній системі. Клієнт може знайти власного психотерапевта з урахуванням таких аспектів, як його спеціалізація, рейтинг та часова доступність (рис. 4.5). Функціонал підсистеми дозволяє здійснювати відбір за конкретними спеціалізаціями, що включають різні напрями психотерапії, а також враховувати рейтинг психотерапевта як міру якості його послуг. Крім того, користувач може оптимально вибирати психотерапевта з урахуванням його часової доступності, що надає можливість адаптувати консультації до власного графіку та потреб.

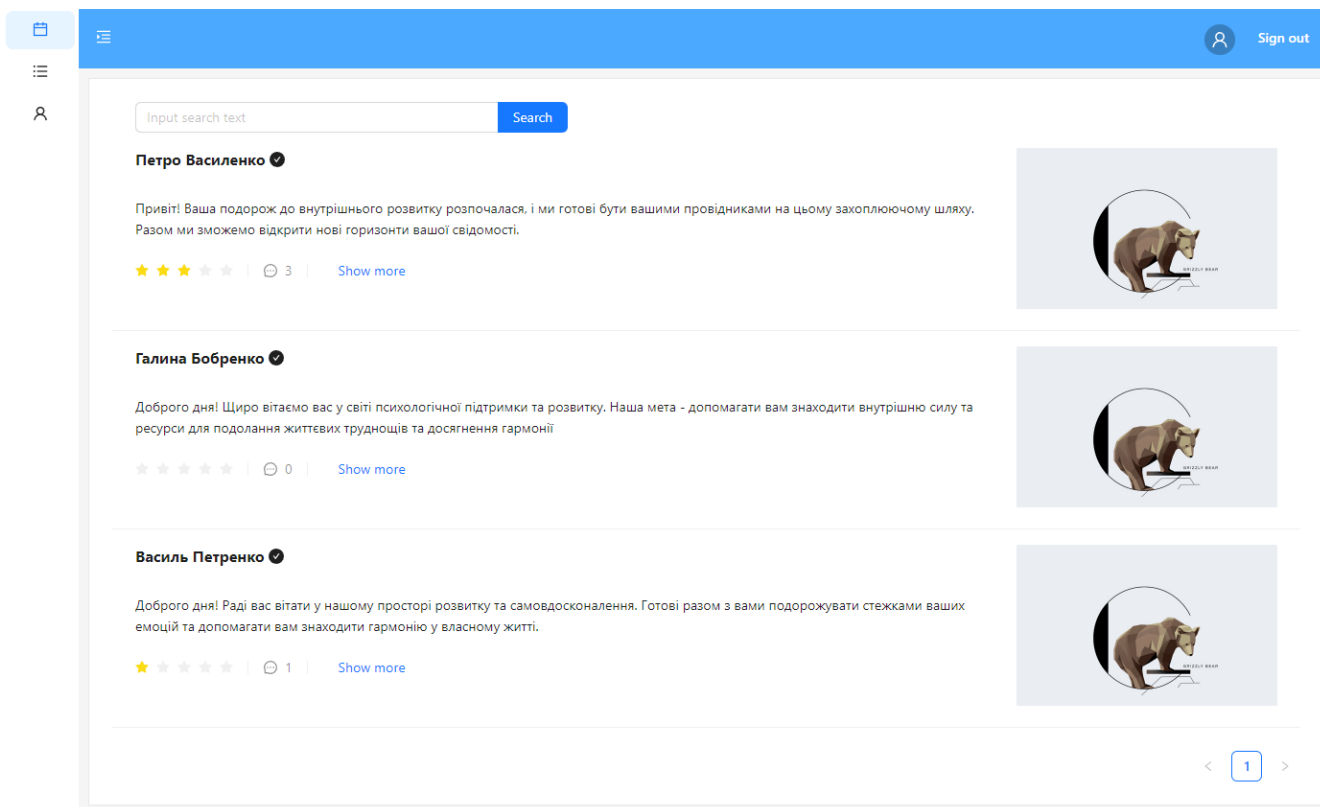
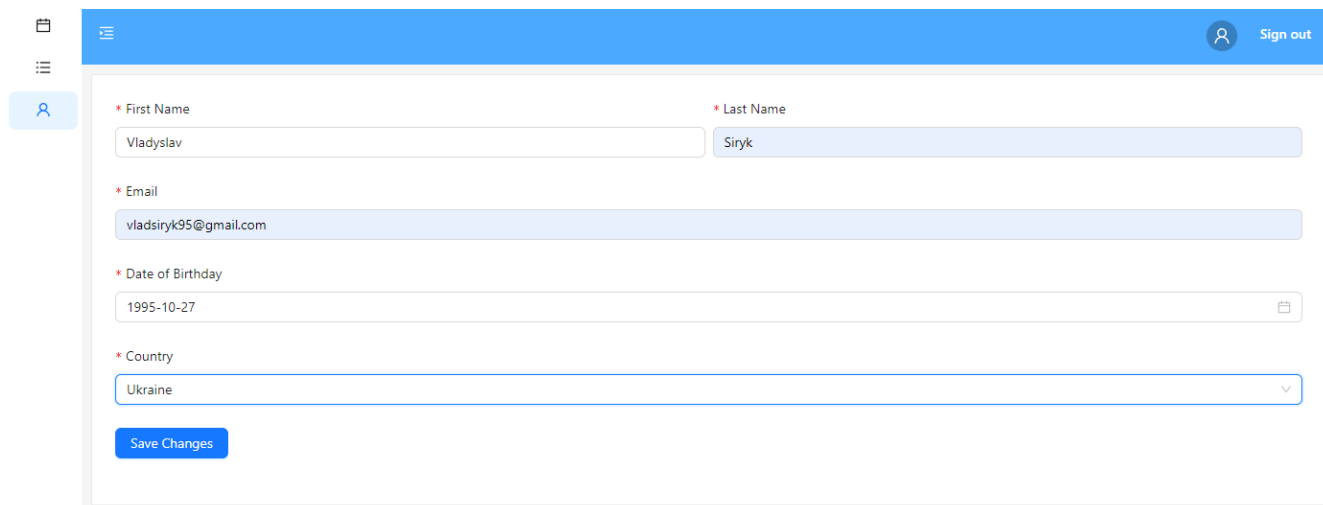


Рисунок 4.5 – Пошук психотерапевтів для клієнт у веб-додатку

Джерело: зображення екрану

Клієнт має можливість редагувати особистий профіль для оптимального відображення своїх потреб та побажань. Це включає в себе зміну особистих даних, оновлення контактної інформації, а також додавання додаткових деталей про власні психологічні або емоційні потреби (рис. 4.6). Редагування профілю сприяє персоналізації досвіду користувача та полегшує взаємодію із платформою.



The screenshot shows a web application interface for editing a client profile. At the top, there is a blue header bar with a 'Sign out' button on the right. On the left side, there is a vertical navigation menu with icons for home, list, and search. The main content area contains a form with the following fields:

- * First Name: Input field containing 'Vladyslav'
- * Last Name: Input field containing 'Siryk'
- * Email: Input field containing 'vladsiryk95@gmail.com'
- * Date of Birthday: Input field containing '1995-10-27' with a calendar icon on the right.
- * Country: Dropdown menu showing 'Ukraine'.

At the bottom of the form is a blue button labeled 'Save Changes'.

Рисунок 4.6 – Екран редагування профіля клієнта у веб-додатку

Джерело: зображення екрану

Основною частиною інформаційної системи є консультація в рамках даного веб-додатку, яка полягає у взаємодії між психотерапевтом та клієнтом у віртуальному середовищі. На рис 4.7 екран відео та аудіо консультації, використовуючи сторонній сервіс 100ms.

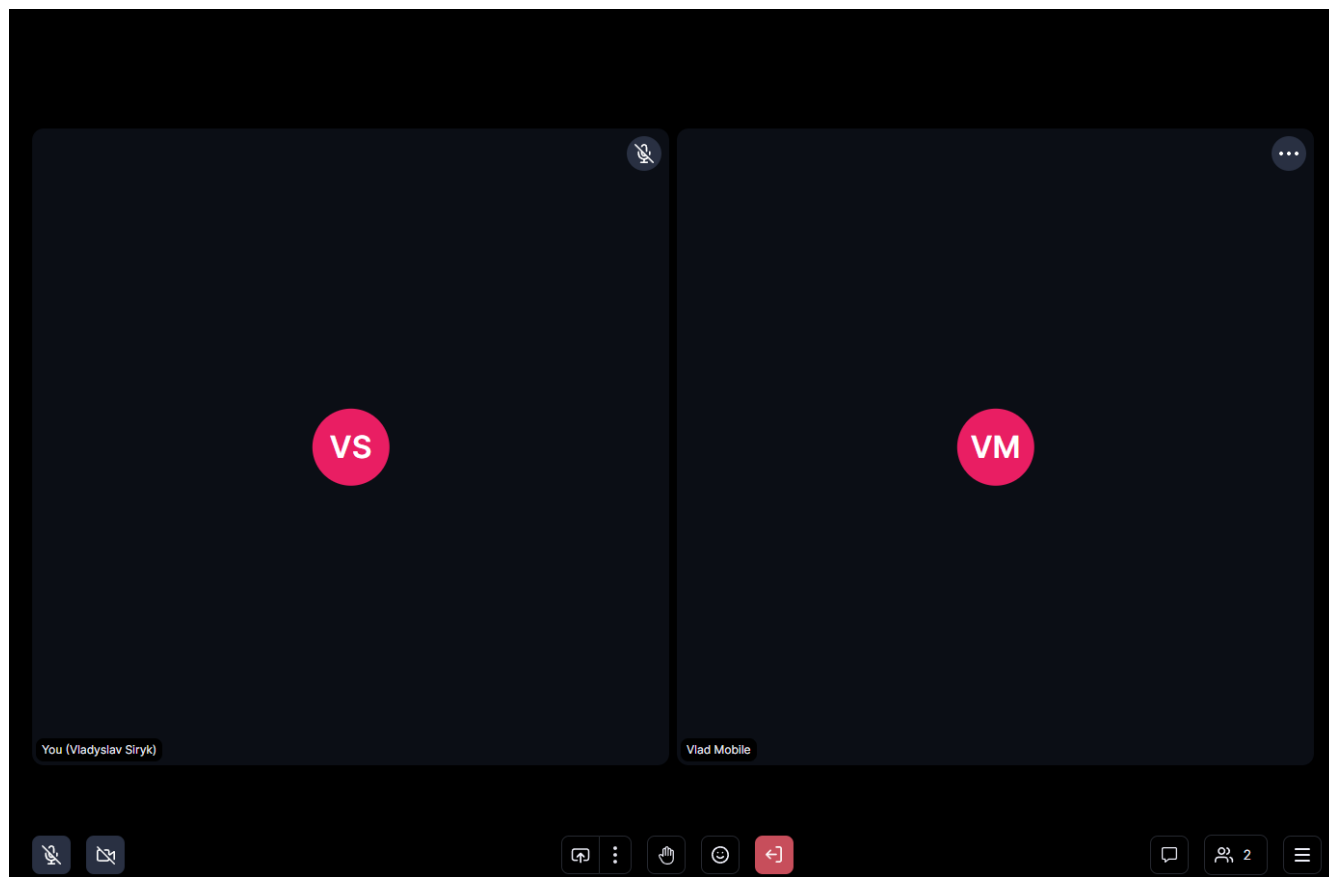


Рисунок 4.7 – Екран відео та аудіо консультації у веб-додатку

Джерело: зображення екрану

Наведені зображення та опис функціональних можливостей веб-додатку показують, як взаємодіють усі компоненти для створення злагоженого інтерфейсу клієнта. Наприклад, обираючи психотерапевта через пошук, користувач легко може записатися на консультацію, а система автоматично підключить до консультації, згідно запису в календарі. Такий підхід сприяє оптимізованому використанню додатка, що робить його більш зручним та ефективним для усіх клієнтів.

4.3 Розробка серверної частини для підсистеми клієнта

Розробка сучасних веб-додатків невіддільно пов'язана із використанням технологій REST API, що дозволяють забезпечити ефективну комунікацію між клієнтським та серверним програмним забезпеченням [39, 40]. У рамках даного

проекту була вирішена задача створення REST API за допомогою п'яти контролерів: ScheduleController, UserController, AuthController, ConsultationController, ReviewController та PsychologyController (рис. 4.8).

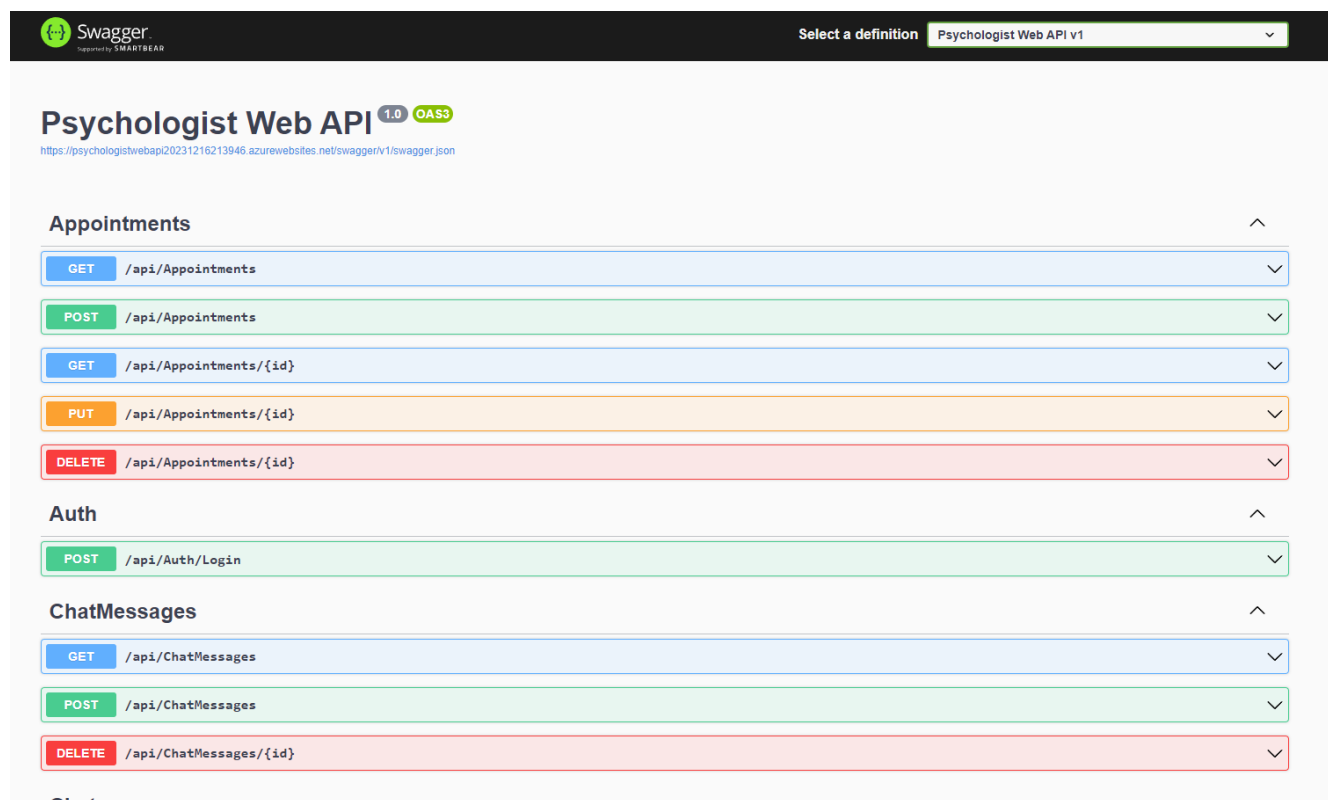


Рисунок 4.8 – Схема кінцевих точок для серверної частини

Джерело: зображення екрану

Розглянемо контролери які використовуються:

- ScheduleController: надає можливість отримання, створення, оновлення та видалення розкладу консультацій;
- UserController: за допомогою цього контролера отримуємо інформацію про клієнтів, створювати нових клієнтів, а також змінювати та видаляти їхні дані;
- AuthController: надає можливість отримання токенів доступу, перевірку валідності сесій та інші функції, пов'язані з безпекою;
- ConsultationController: функціональність включає отримання списку

консультацій, створення нових записів та оновлення інформації про консультації;

- ReviewController: відповідає за обробку запитів, пов'язаних із відгуками. Він містить методи для отримання, збереження та оновлення відгуків про психотерапевтів у системі;
- PsychologyController: у контролері реалізована функціональність знаходження психологів за встановленими фільтрами а також отримання детальної інформації про психолога.

Такий підхід до розробки REST API сприяє прозорості коду, забезпечує простоту управління та розширення функціоналу додатку. За допомогою цієї структури розробники можуть легко ідентифікувати та модифікувати різні компоненти системи, що робить розробку, тестування та супровід додатку більш зручним та ефективним.

ВИСНОВКИ

У даній кваліфікаційній роботі магістра було розглянуто розробку інформаційної системи підтримки надання психологічної допомоги. Ця система розроблена з метою покращення процесу комунікації між пацієнтами та психологами, забезпечення ефективного управління психологічними сесіями та підвищення загальної доступності якісних психологічних послуг.

У процесі розробки було враховано потреби цільової аудиторії, включаючи легкість використання системи для пацієнтів і гнучкість функціоналу для психологів. Інтерфейс користувача був спроектований з акцентом на інтуїтивно зрозумілість та зручність, що виявилось ключовим для забезпечення високого рівня задоволеності користувачів.

Для реалізації проекту були використані сучасні технології, включаючи C# та React. Вибір цих технологій забезпечив гнучкість та масштабованість системи. Система пройшла комплексне тестування, що дозволило виявити та усунути потенційні проблеми.

Для подальшого розвитку системи можуть бути розглянуті такі напрямки, як впровадження штучного інтелекту для покращення аналітичних можливостей, розширення функціоналу з урахуванням нових психотерапевтичних технік та методів, а також інтеграція з іншими медичними системами для створення єдиної ефективної екосистеми психологічної допомоги.

У підсумку, розроблена система демонструє значний потенціал у поліпшенні доступності та якості психологічної допомоги. Вона сприяє оптимізації робочих процесів, підвищує ефективність взаємодії між психологами та пацієнтами та вносить значний вклад у сучасну психотерапевтичну практику.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гончарова, Г. М., Петрова, Н. С. Використання інформаційних систем у психотерапії. Психотерапія і консультування. 2019. № 1. С. 67-79.
2. Сидоренко, М. С., Шевченко, В. О. Використання інформаційних систем у роботі психолога. Психологічна допомога і консультування. 2021. № 4. С. 23-36.
3. Шевченко, О. С., Петров, І. В. Використання інформаційних технологій у психологічних центрах. Інновації в психології. 2019. № 1. С. 56-68.
4. Кравченко, О. М., Гончарова, М. Г. Використання інформаційних технологій у психологічній підтримці населення. Психологічні дослідження і практика. 2019. № 6. С. 23-36. Використання мережевих технологій у психологічній роботі / І. М. Петров, Н. М. Ткаченко // Психологія і комп'ютери. - 2018. - № 5. - С. 23-36.
5. Гончаров, М. С., Петренко, Г. О. Ефективність використання інформаційних технологій у роботі психолога. Психологічні науки. 2021. № 3. С. 34-47.
6. Полякова, М. О., Гончарова, Т. О. Ефективність інформаційних систем у психологічній діяльності. Психологічна інформатика. 2022. № 3. С. 34-47.
7. Гончаров, В. С., Ткаченко, О. В. Ефективність інформаційних систем у психологічній роботі. Психологічний прогрес. 2021. № 2. С. 23-36.
8. Поляков, Г. О., Гончарова, О. Г. Інноваційні підходи до використання інформаційних систем у психологічній діяльності. Психологічні дослідження і практика. 2018. № 6. С. 12-24.
9. Гончарова, М. М., Ткаченко, О. О. Інноваційні підходи до створення інформаційних систем для психологічної допомоги. Інформаційні технології і психологія. 2020. № 1. С. 45-58.
10. Поляков, Г. М., Гончарова, Т. М. Інтерактивні інформаційні системи для психологічної підтримки. Психологічні дослідження і практика. 2020. № 1. С. 45-58.
11. Ткаченко, В. М., Лисенко, Г. М. Інтерактивність інформаційних систем для

- психологічної допомоги. Інновації в психології. 2020. № 4. С. 23-36.
12. Поляков, Г. С., Ткаченко, Т. М. Інтерактивність інформаційних систем у психологічній інформатиці. Психологічна інформатика. 2018. № 5. С. 23-36.
 13. Сидоренко, І. М., Шевченко, Н. С. Інтерактивність інформаційних систем у роботі психолога. Інноваційні підходи в психологічній допомозі. 2021. № 4. С. 12-24.
 14. Кравченко, М. М., Полякова, О. В. Інформаційні системи в психологічній діяльності: досягнення і виклики. Психологічні аспекти розвитку. 2017. № 4. С. 45-58.
 15. Гончаров, Г. М., Ткаченко, О. Г. Інформаційні системи для надання психологічної допомоги: досягнення і перспективи. Психологічні аспекти розвитку. 2020. № 6. С. 45-58.
 16. Іванова, В. С., Гончаров, О. М. Інформаційні технології в психологічному консультуванні. Психологічні дослідження. 2021. № 2. С. 45-58.
 17. Полякова, О. Г., Гончарова, Т. М. Інформаційні системи психологічної підтримки: досвід впровадження. Інформаційні технології у соціальній сфері. 2018. № 5. С. 23-36.
 18. Гончарова, М. О., Соколова, Н. М. Інформаційні системи психологічної підтримки: проблеми і перспективи. Психологічні аспекти розвитку. 2020. № 3. С. 34-47.
 19. Ткаченко, І. О., Гончаров, В. Г. Інформаційні технології в психологічному консультуванні. Психологічний прогрес. 2020. № 5. С. 23-36.
 20. Лисенко, Г. О., Гончарова, Т. М. Інформаційні технології в психологічному консультуванні. Психологічні науки. 2018. № 1. С.
 21. Price M. J. C# 7.1 and .NET Core 2.0 – Modern Cross-Platform Development. Third Edition. / M. J. Price: Packt Publishing, 2017 – 800 p.
 22. Офіційна документація Microsoft з ASP.Net Core: Introduction to ASP.NET Core [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/core/>. (дата звернення: 15.11.2023).

23. Albahari, J., Albahari, B. LINQ Pocket Reference: Learn and Implement LINQ for .NET Applications. O'Reilly Media, 2008. 174 p.
24. Методи та засоби розвитку інформаційних систем у психологічній практиці / Т. В. Гончарова, М. О. Соколова // Психологічний інструментарій. - 2018. - № 2. - С. 45-58.
25. Schwarzmüller, Maximilian, and Abramov, Dan. React: The Complete Guide. Packt Publishing, 2023. 256p.
26. Чиннатамби К. Изучаем React. [Електронний ресурс] – Режим доступу до ресурсу: <https://grenka.ua/326415/izuchaemreact>. Дата звернення: 15.11.2023.
27. Nikam, P. P., & Suryawanshi, R. S. (2015). Microsoft windows azure: developing applications for highly available storage of cloud service. International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Volume 4 Issue 12, December 2015
28. K. Lachewicz, Performance analysis of selected database systems: MySQL, MS SQL, PostgreSQL in the context of web applications. Journal of Computer Sciences Institute. 14, (Mar. 2020), 94-100.
29. Офіційна документація Microsoft для Visual Studio. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/visualstudio/> (дата звернення: 27.10.2023).
30. The complete guide to understand IDEF diagram. edrawmax.com [Електронний ресурс] – Режим доступу до ресурсу: <https://www.edrawmax.com/article/the-complete-guide-to-understand-idef-diagram.html>. (дата звернення: 05.12.2023).
31. Саварин П. Електронний посібник із дисципліни "Комп'ютерні системи та мережі". Лекція 6. Нотація IDEF0 // Луцький національний технічний університет, Факультет цифрових, освітніх та соціальних технологій [Електронний ресурс] – Режим доступу до ресурсу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%BE%D0%BD%D0%B4%D1%96%D1%83%D1%81%202%20%D0%B3%D0%BE%D1%2%D0%BE%D0%B2%D0%B0/page9.html. (дата звернення:
32. https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%BE%D0%BD%D0%B4%D1%96%D1%83%D1%81%202%20%D0%B3%D0%BE%D1%2%D0%BE%D0%B2%D0%B0/page9.html. (дата звернення:

06.12.2023).

33. Основи UML [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html>. (дата звернення: 06.12.2023).
34. Dragoni N. et al. Microservices: yesterday, today, and tomorrow [Електронний ресурс] – Режим доступу до ресурсу: <http://arxiv.org/abs/1606.04036>. (дата звернення: 06.12.2023).
35. Петренко О.О. Порівняння типів архітектури систем сервісів. *System Research & Information Technologies*. 2015. № 4. С. 48–62.
36. R. Kelley, A. D. Antu, A. Kumar, and B. Xie, "Choosing the Right Compute Resources in the Cloud: An analysis of the compute services offered by Amazon, Microsoft, and Google," 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Chongqing, China, 2020, pp. 214-223
37. Doglio F. Pro REST API Development with Node.js. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.goodreads.com/book/show/25635366-pro-rest-api-development-with-node-js> (дата звернення: 03.11.2023).
38. David Li. *Building Web Applications with Go and Beego: An Introduction to Backend Development*. Birmingham : On Kindle Scribe, 2023. 350 p.
39. Сучасні підходи до створення інформаційних систем для психологічної підтримки / В. М. Гончаров, Г. В. Лисенко // *Психологічна інформатика*. - 2020. - № 3. - С. 34-47.
40. Шевченко, І. І. Розвиток інформаційних систем у психологічній допомозі / І. І. Шевченко // *Психологія та педагогіка*. - 2019. - № 3. - С. 45-58.

ДОДАТОК А. ПЛАНУВАННЯ РОБІТ

А.1 Ідентифікація мети ІТ-проекту

Система SMART є найпоширенішим методом створення та уточнення цілей, названим за аббревіатурою англійських слів: specific (конкретна), measurable (вимірювана), achievable (досяжна), relevant (сумісна) і time-bound (визначена в часі). Ці принципи допомагають не тільки чітко визначити завдання, але й спрямовують ефективні зусилля команди.

Метою виконання кваліфікаційної роботи магістра є підсистема клієнта інформаційної системи підтримки надання психологічної допомоги, яка повинна базуватися на SMART-принципах.

Нижче представлена таблиця, що деталізує основні аспекти проекту, використовуючи SMART-критеріїв, забезпечуючи систематичний та вимірюваний підхід до досягнення успіху у визначених сферах. Результати деталізації методом SMART розміщені у табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

Назва	Опис
Specific (конкретна)	Розробити підсистему клієнта в інформаційній системі підтримки надання психологічної допомоги з необхідними функціями, яка спрямована на поліпшення психічного здоров'я користувачів.
Measurable (вимірювана)	Забезпечити можливість зручного та ефективного доступу користувачів до ресурсів системи та стабільну роботу системи.
Achievable (досяжна)	Переконатись, що у наявності є достатньо технічних та програмних ресурсів для виконання проекту.

Продовження таблиці А.1

Relevant (реалістична)	Розробити зручний інтерфейс для введення особистої інформації та спілкування з психологом, а також забезпечити безпеку та конфіденційність даних користувачів.
Time-framed (обмежена у часі)	Термін розробки клієнтської частини обмежений по часу виконанням кваліфікаційної роботи.

Джерело: власна розробка

Проект інформаційної системи психологічної допомоги відповідає принципам SMART, визначаючи чіткі цілі, вимірювані показники успіху та реалістичні завдання. Проект фокусується на створенні ефективного інтерфейсу для покращення психічного здоров'я користувачів, враховуючи обмеження ресурсів та дотримання стандартів безпеки. Чіткі терміни виконання завдань забезпечують своєчасну реалізацію проекту. Загалом, проект виявляє систематичний та вимірюваний підхід до досягнення успіху.

А.2 Планування змісту структури робіт інформаційної системи

WBS (рис. А.1) представляє собою ієрархічну декомпозицію проекту на фази, кінцеві результати та пакети робіт. Це ієрархічна структура, що демонструє розподіл необхідних зусиль для досягнення мети, таких як програма чи проект. Розробка WBS зазвичай відбувається на початковому етапі проекту перед детальним плануванням.

Етапи створення WBS:

1. Визначення рівня деталізації проектних робіт так, щоб їх можна було оцінити.

2. Визначення кількості рівнів, зазвичай три-чотири, при цьому для сучасних компаній оптимально чотири рівні.
3. Розробка структури для кожного рівня, що включає формування горизонтальних рівнів.
4. Створення коротких описів для елементів WBS, що складається з їхніх стислих назв.

WBS організована навколо ключових продуктів проекту (чи запланованих результатів), а не необхідних робіт для їхнього випуску. Це дозволяє легко визначити кожен діяльність проекту та звести її до однієї термінальної події в WBS.

Для створення WBS нашої підсистеми ми використовуємо ієрархічний підхід. На першому рівні розглядається весь проект, на другому – його етапи, на третьому – блоки, і, нарешті, на четвертому рівні визначаються конкретні робочі завдання.

На рисунку А.1 відображено WBS з розробки підсистеми клієнта в інформаційній системі підтримки надання психологічної допомоги.

Після побудови WBS, далі йде розробка організаційної структури виконавців (OBS), яка є графічним відображенням учасників або відповідальних осіб, що беруть участь у реалізації проекту. Рисунок А.2 відображає організаційну структуру проекту (OBS). Таблиця А.2 містить перелік учасників, які беруть участь у виконанні проекту.

Таблиця А.2 – Виконавці проекту

Роль	ПІБ	Опис
Менеджер проекту	Сірик В.Ю.	Визначає стратегію та забезпечує комунікацію між учасниками проекту.
Розробник	Сірик В.Ю.	Відповідає за розробку функціоналу.

Продовження таблиці А.2

Тестувальник	Сірик В.Ю.	Виявляє помилки в програмному забезпеченні.
Документатор	Сірик В.Ю.	Створює та підтримує технічну документацію проекту, забезпечуючи доступність і актуальність.
Архітектор	Сірик В.Ю., Шапка С.О., Парфененко Ю.В.	Визначає архітектурні принципи та контролює дотримання стандартів.
Аналітик	Сірик В.Ю., Шапка С.О.	Визначає вимоги до програмного забезпечення.
Дизайнер	Сірик В.Ю.	Розробляє інтерфейс користувача, забезпечує відповідність дизайну.
Серверний інженер	Сірик В.Ю., Шапка С.О.	Відповідає за установку, конфігурацію та обслуговування серверного обладнання та програмного забезпечення.

Джерело: власна розробка

А.3 Побудова календарного графіку виконання інформаційної системи

У цьому розділі роботи представлено діаграму Гранта для нашого проекту. Діаграма Гранта виступає важливим інструментом для візуалізації стратегії впровадження та структури проекту. Вона не лише надає можливість ілюструвати послідовність та тривалість завдань, але й дозволяє ефективно визначати критичні

шляхи, розподіляти ресурси, та попереджати можливі затримки у виконанні проекту.

Кожен сегмент відповідає конкретному завданню або підзадачі, і їх розташування у плані відображається по вертикалі. Початок, кінець і тривалість сегмента на шкалі часу відповідають відповідно початку, кінцю і тривалості відповідного завдання. Деякі діаграми також відображають взаємозв'язок між різними завданнями. Нижче наведено рисунок А.3 з діаграмою Гранта.

Початок проекту відзначений 2 жовтня 2023 року. Протягом виконання проекту відводилось по 3 години на день, включаючи вихідні. З діаграми видно, що робота над різними етапами відбувалася послідовно, за винятком етапу контролю та моніторингу, де процеси відбувалися паралельно. На виконання цього етапу у загальному обсязі було витрачено більше 60 днів.

А.4 Планування ризиків проекту

Проведення якісної оцінки ризиків дозволяє визначити ті, які є найбільш небезпечними та вимагають негайного усунення. Важливість ризику визначає, які конкретні заходи реагування слід прийняти. Далі в процесі є кількісне оцінювання ризиків, яке може проводитися паралельно з якісною оцінкою чи окремо, залежно від вимог проекту.

У таблиці А.3 міститься перелік ризиків, ідентифікованих для даного проекту. Результати оцінки ризиків представлені в таблиці А.4. Таблиця А.5 включає шкалу для класифікації ризиків за ступенем впливу на проект та ймовірністю та впливу.

Таблиця А.3 – Ризики проекту

№ ризику	Назва (опис) ризику
1	Ризик порушення конфіденційності. Психологічна інформація про клієнтів є дуже чутливою. Якщо не буде забезпечено достатньо високий рівень безпеки, інформація може бути доступна третім особам, що може призвести до порушення конфіденційності.
2	Недостатньо досвідчений персонал. Якщо персонал не має достатнього досвіду, це може призвести до затримок у розробці, помилок у програмуванні або неякісного продукту.
3	Недостатньо чітке розуміння вимог. Може призвести до того, що підсистема не буде відповідати потребам користувачів або буде надмірно складною або дорогою у розробці та підтримці.
4	Непередбачені обставини. Під час розробки будь-якого проекту можуть виникнути непередбачені обставини, такі як зміни вимог, затримки у наданні матеріалів або хвороба персоналу.
5	Питання етики. Збір та обробка психологічної інформації може породжувати етичні питання. Недостатня увага до етичних аспектів може призвести до негативних наслідків для користувачів.
6	Проблема із доступом до сервісів. В системі планується використання окремих сервісів, за які ми не зможемо відповідати.

Джерело: власна розробка

Таблиця А.4 – Результати визначення ймовірності, впливу та рангу ризиків проекту

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Ризик порушення конфіденційності	0,1	0,8	0,08
2	Недостатньо досвідчений персонал	0,5	0,4	0,2
3	Недостатньо чітке розуміння вимог	0,9	0,6	0,54
4	Непередбачені обставини	0,1	0,6	0,06
5	Питання етики	0,3	0,4	0,12
6	Проблема із доступом до сервісів	0,3	0,4	0,12

Джерело: власна розробка

Отже, на основі отриманих ймовірностей, будемо матрицю ймовірності та впливу (табл. А.5), де розташуємо кожен ризик. Введемо пороги для визначення рангу ризику:

- дуже низький ризик – 0,05;
- низький ризик – 0,1;
- середній ризик – 0,4;
- високий ризик – 0,6;
- дуже високий ризик – 0,8.

Для оцінки ймовірності використовувалася шкала з трьох рівнів: висока ймовірність виникнення ризику, середня ймовірність можливого виникнення, але

не обов'язково, і низька ймовірність малоїмовірного виникнення ризику.

Таблиця А.5 – Матриця ймовірності та впливу

Ймовірність виникнення ризика	Вплив ризику				
	0,05	0,1	0,4	0,6	0,8
0,9	0,045	0,09	0,36	0,54 (R1)	0,72
0,7	0,035	0,07	0,28	0,42	0,56
0,5	0,025	0,05	0,20 (R2)	0,3	0,40
0,3	0,015	0,03	0,12(R3, R4)	0,18	0,24
0,1	0,005	0,01	0,04	0,06 (R6)	0,08 (R5)

Джерело: власна розробка

Результати оцінки ризиків використовуються для розробки планів реагування на ризики. Плани реагування на ризики для нашої інформаційної системи, які допоможуть зменшити ймовірність виникнення ризику або його вплив наведенні нижче (табл. А.6).

Таблиця А.6 – Ризики та стратегії реагування на них

Назва ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План реагування
Ризик порушення конфіденційності	Низька	Дуже висока	0,08	Перенесення	<ol style="list-style-type: none"> 1 Розробити політику конфіденційності, яка буде визначати правила доступу до бази даних. 2 Провести навчання персоналу з питань безпеки інформації.
Недостатньо досвідчений персонал	Середня	Середня	0,2	Прийняття	<ol style="list-style-type: none"> 1 Залучити до проекту досвідченого персоналу. 2 Розробити систему наставництва для молодих спеціалістів.
Недостатньо чітке розуміння вимог	Висока	Висока	0,54	Зменшення	<ol style="list-style-type: none"> 1 Створити документ з уточненнями та погодженнями.
Непередбачені обставини	Низька	Висока	0,06	Прийняття	<ol style="list-style-type: none"> 1 Розробити універсальний план дій для кожного можливого сценарію. 2 Забезпечити резервні рішення для критичних аспектів.

Продовження таблиці А.6

Питання етики	Середня	Середня	0,12	Прийняття	1 Звернутися до етичних норм та стандартів для визначення правильного вирішення.
Проблема із доступом до сервісів	Середня	Середня	0,12	Перенесення	2 Звернутися до технічної підтримки для вирішення проблеми. 3 Розглянути можливість використання альтернативних сервісів або методів.

Джерело: власна розробка

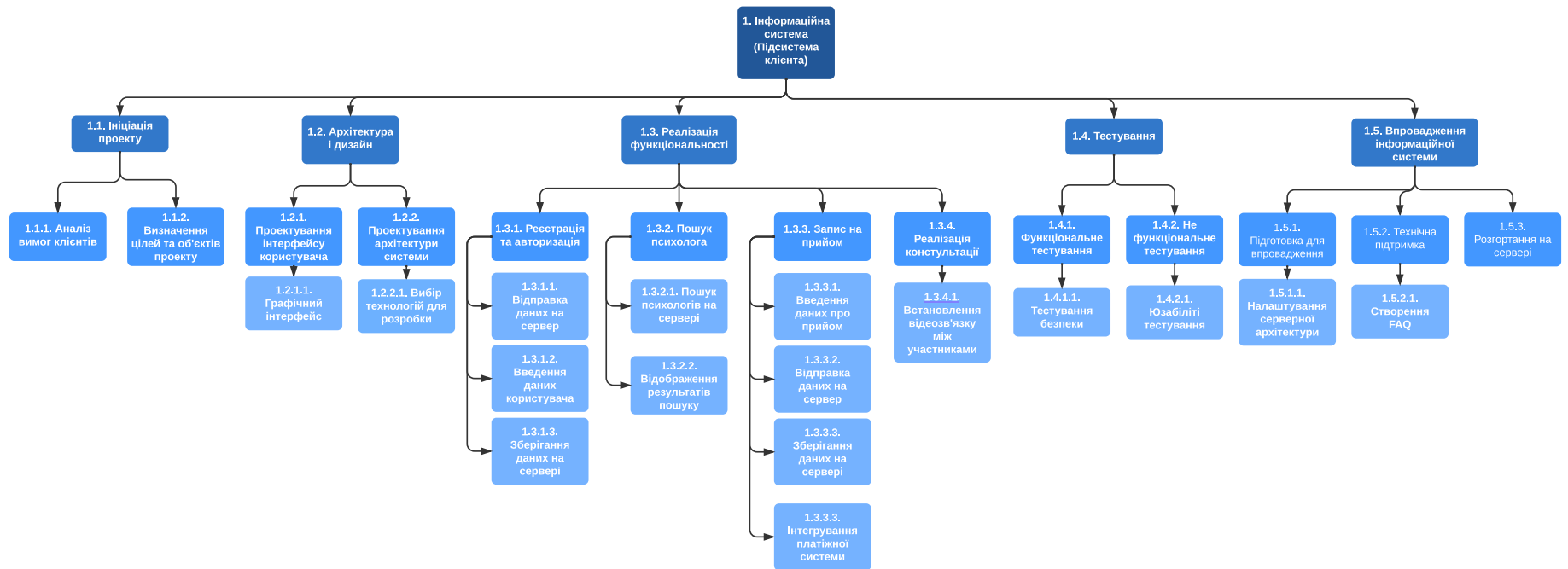


Рисунок А.1 – Діаграма декомпозиції робіт проекту (WBS)

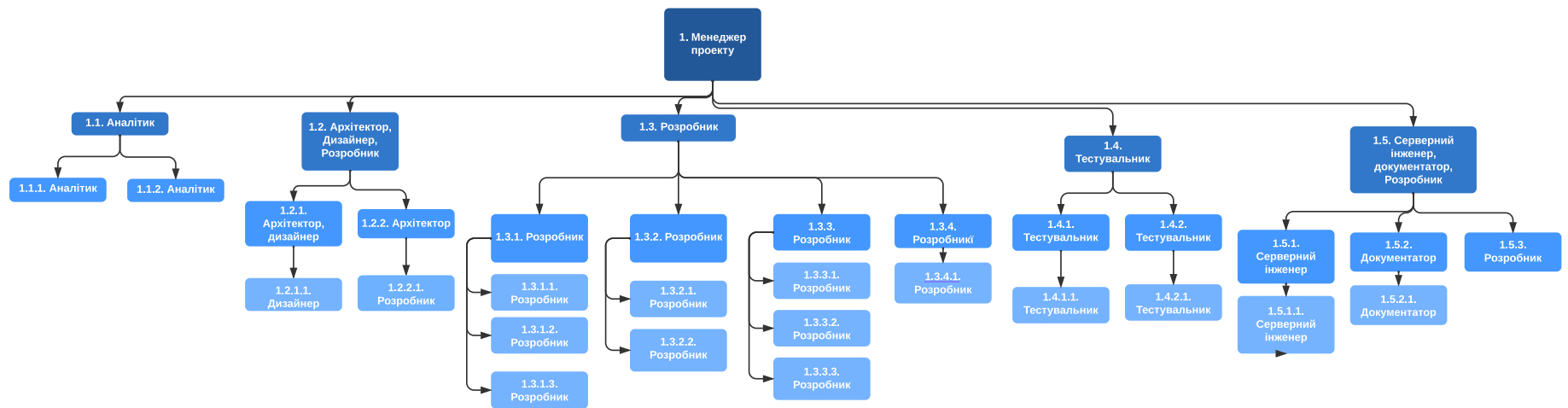


Рисунок А.2 – OBS-структура робіт проекту

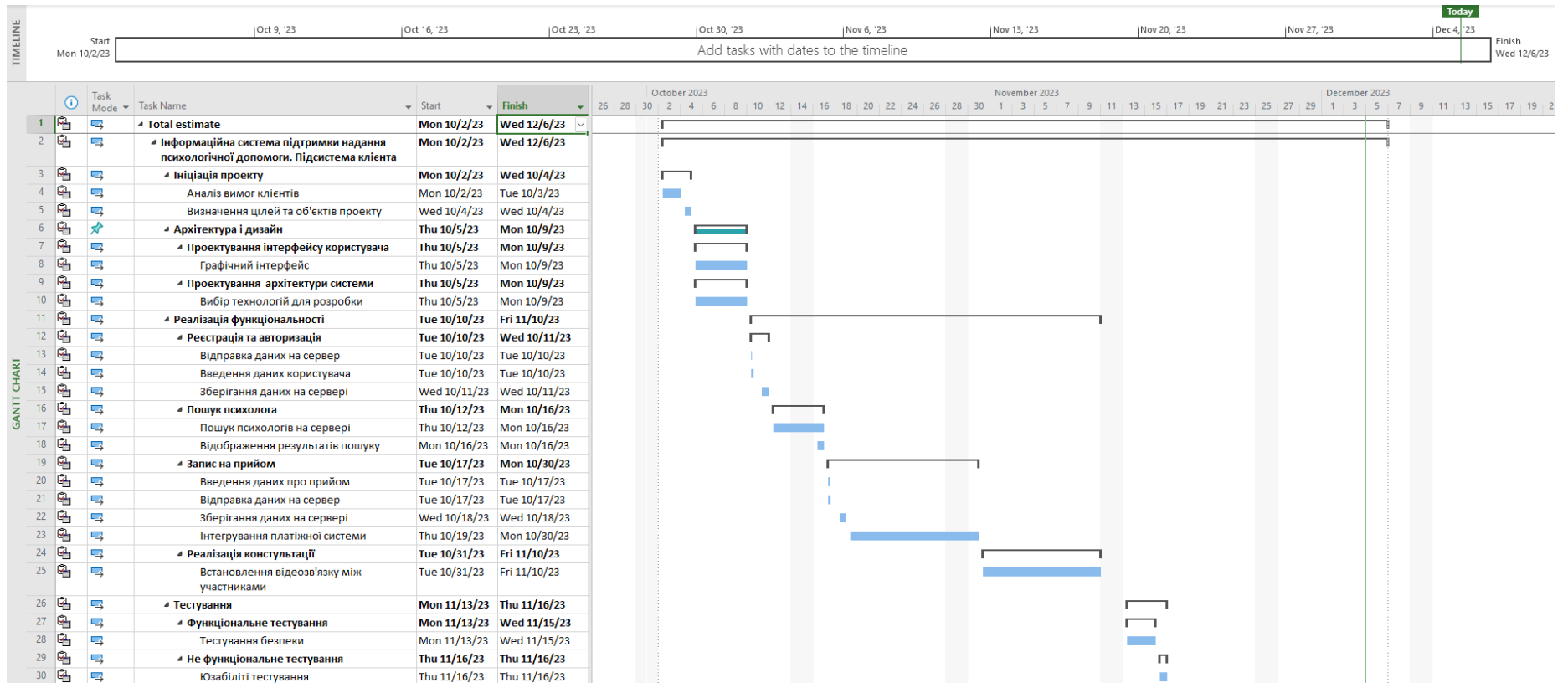


Рисунок А.3 – Діаграма Ганта

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ

Лістинг файлу Login.jsx:

```
import React from "react";
import LoginForm from "./LoginForm";
import { createStyles } from "antd-style";
import { Row, Col, Divider } from "antd";
import { useTranslation } from "react-i18next";
import { Link } from "react-router-dom/dist";

const useStyles = createStyles(({ token, css, cx }) => {
  return {
    container: css`
      display: flex;
      flex-direction: column;
      height: 100vh;
      background-color: #4ba9ff;
      background-size: 100% 100%;
    `,
    containerForm: css`
      height: 100%;
      margin: 30px;
      background-color: #ffffff;
      border-radius: 25px;
      display: flex;
      justify-content: center;
      align-items: center;
      width: 40%;
    `
  };
});

const Login = () => {
  const { styles } = useStyles();
  const { t } = useTranslation();

  return (
    <div className={styles.container}>
      <div className={styles.containerForm}>
        <div style={{ width: "100%", padding: "20%" }}>
          <Row style={{ paddingBottom: 20 }}>
            <Col span={24} style={{ fontSize: 28, fontWeight: 500 }}>
              {t("loginForm.welcom")}
            </Col>
          </Row>
          <Row>
            <Divider>{t("loginForm.signIn")}</Divider>
          </Row>
        </div>
      </div>
    </div>
  );
};
```



```

        </Row>
        <Row>
          <Col span={24}>
            <LoginForm />
          </Col>
        </Row>
        <Row>
          <Col span={24}>
            {t("loginForm.notRegistered")}{ " "}
            <Link
to="/registration">{t("loginForm.createAccount")}</Link>
          </Col>
        </Row>
      </div>
    </div>
  </div>
);
};

export default Login;

```

Лістинг файлу LoginForm.jsx:

```

import { Form, Input, Button, message } from "antd";
import { UserOutlined, LockOutlined } from "@ant-design/icons";
import { useTranslation } from "react-i18next";
import { useNavigate } from "react-router-dom/dist";
import { useSignIn } from "react-auth-kit";
import axios, { AxiosError } from "axios";

const LoginForm = () => {
  const { t } = useTranslation();
  const navigate = useNavigate();
  const signIn = useSignIn();
  const [messageApi, contextHolder] = message.useMessage();

  const handleError = (errorMessage) => {
    messageApi.open({
      type: "error",
      content: errorMessage,
    });
  };

  const handleSubmit = async (values) => {
    try {
      const response = await axios.post(
"https://psychologistwebapi20231216213946.azurewebsites.net/api/Auth/
Login",
      {
        login: values.username,

```

```

        password: values.password,
    }
);

if (
    signIn({
        token: response.data.token,
        expiresIn: 3600,
        tokenType: "Bearer",
    })
) {
    navigate("/");
} else {
    throw new Error(t("loginForm.requestError"));
}
} catch (err) {
    if (err && err instanceof AxiosError) {
        handleError(err.response?.data.message || err.message);
    } else if (err && err instanceof Error) {
        handleError(err.message);
    }
}
};

return (
    <>
    {contextHolder}
    <Form
        initialValues={{
            remember: true,
        }}
        onFinish={handleSubmit}
    >
    <Form.Item
        name="username"
        rules={[
            {
                required: true,
                message: t("loginForm.usernameErrorMessage"),
            },
        ]}
    >
    <Input
        prefix={<UserOutlined />}
        placeholder={t("loginForm.username")}
    />
    </Form.Item>
    <Form.Item
        name="password"
        rules={[
            {

```

```

        required: true,
        message: t("loginForm.passwordErrorMessage"),
      },
    ]}
  >
  <Input
    prefix={<LockOutlined />}
    type="password"
    placeholder={t("loginForm.password")}
  />
</Form.Item>
<Form.Item>
  <Button          type="primary"          htmlType="submit"
style={{ width: "100%" }}>
  {t("loginForm.buttonText")}
</Button>
</Form.Item>
</Form>
</>
);
};

export default LoginForm;

```

Лістинг файлу Registration.jsx:

```

import React from "react";
import RegistrationForm from "../RegistrationForm";
import { createStyles } from "antd-style";
import { Row, Col, Divider } from "antd";
import { useTranslation } from "react-i18next";
import { Link } from "react-router-dom/dist";

const useStyles = createStyles(({ token, css, cx }) => {
  return {
    container: css`
      display: flex;
      flex-direction: column;
      height: 100vh;
      background-color: #4ba9ff;
      background-size: 100% 100%;
    `,
    containerForm: css`
      height: 100%;
      margin: 30px;
      background-color: #ffffff;
      border-radius: 25px;
      display: flex;
      justify-content: center;
      align-items: center;
      width: 50%;
    `
  };
});

```

```

    },
  };
});

const Registration = () => {
  const { styles } = useStyles();
  const { t } = useTranslation();

  return (
    <div className={styles.container}>
      <div className={styles.containerForm}>
        <div style={{ width: "100%", padding: "20%" }}>
          <Row>

<Divider>{t("registrationForm.createAccount")}</Divider>
          </Row>
          <Row>
            <Col span={24}>
              <RegistrationForm />
            </Col>
          </Row>
          <Row style={{ paddingTop: 10, textAlign: 'center' }}>
            <Col span={24}>
              <Link
to="/login">{t("registrationForm.back")}</Link>
              </Col>
            </Row>
          </div>
        </div>
      </div>
    );
  };

export default Registration;

```

Лістинг файлу RegistrationForm.jsx:

```

import React, { useState } from "react";
import { Form, Steps, message } from "antd";
import { useTranslation } from "react-i18next";
import { useNavigate } from "react-router-dom/dist";
import { useSignIn } from "react-auth-kit";
import
  RegistrationUserFormStep from
"./RegistrationUserFormStep";
import
  RegistrationPasswordFormStep from
"./RegistrationPasswordFormStep";
import
  RegistrationQuestionFormStep from
"./RegistrationQuestionFormStep";
import axios, { AxiosError } from "axios";

const { Step } = Steps;

```

```

const RegistrationForm = () => {
  const [currentStep, setCurrentStep] = useState(0);
  const [form] = Form.useForm();
  const { t } = useTranslation();
  const navigate = useNavigate();
  const signIn = useSignIn();
  const [messageApi, contextHolder] = message.useMessage();

  const handleError = (errorMessage) => {
    messageApi.open({
      type: "error",
      content: errorMessage,
    });
  };

  const handleSubmit = async (values) => {
    try {
      const response = await axios.post(
"https://psychologistwebapi20231216213946.azurewebsites.net/api/Clients/Register",
        values
      );

      if (
        signIn({
          token: response.data.token,
          expiresIn: 3600,
          tokenType: "Bearer",
        })
      ) {
        navigate("/");
      } else {
        throw new Error(t("registrationForm.requestError"));
      }
    } catch (err) {
      if (err && err instanceof AxiosError) {
        handleError(err.response?.data.message || err.message);
      } else if (err && err instanceof Error) {
        handleError(err.message);
      }
    }
  };

  const handleNextStep = async () => {
    try {
      await form.validateFields();
      setCurrentStep(currentStep + 1);
    } catch {
      return;
    }
  };

```

```

    }
  };

  const handlePreviousStep = () => {
    setCurrentStep(currentStep - 1);
  };

  const steps = [
    {
      title: t("registrationForm.personalInformation"),
      content: (
        <RegistrationUserFormStep
handleNextStep={handleNextStep} form={form} />
      ),
    },
    {
      title: t("registrationForm.confirmPassword"),
      content: (
        <RegistrationPasswordFormStep
          handleNextStep={handleNextStep}
          handlePreviousStep={handlePreviousStep}
          form={form}
        />
      ),
    },
    {
      title: t("registrationForm.questions"),
      content: (
        <RegistrationQuestionFormStep
          handlePreviousStep={handlePreviousStep}
          handleSubmit={handleSubmit}
          form={form}
        />
      ),
    },
  ];

  return (
    <div>
      {contextHolder}
      <Steps
        progressDot
        current={currentStep}
style={{ paddingBottom: 20 }}>
        {steps.map((step) => (
          <Step key={step.title} title={step.title} />
        ))}
      </Steps>
      <div>{steps[currentStep].content}</div>
    </div>
  );
};

```

```
export default RegistrationForm;
```

Лістинг файлу PsychologistDetailsModal.jsx:

```
import React from "react";
import {
  Modal,
  List,
  Tabs,
  Timeline,
  Rate,
  Avatar,
  Divider,
  Image,
  Row,
  Col,
  Space,
  Tag,
} from "antd";
import { CheckCircleFilled } from "@ant-design/icons";
import { useTranslation } from "react-i18next";

const { TabPane } = Tabs;

const PsychologistDetailsModal = ({ visible, onCancel, item }) =>
{
  const { t } = useTranslation();
  return (
    <Modal open={visible} onCancel={onCancel} footer={null}>
      <Row gutter={8}>
        <Col span={8}>
          <Image src={item.client?.profileLink || ""} />
        </Col>
        <Col span={16}>
          <Row gutter={8}>
            <Col span={24}>
              {item.firstName} {item.lastName}{" "}
              {item.isVerified && <CheckCircleFilled />}
            </Col>
            <Col span={24}>{item.description}</Col>
            <Col span={24}>
              <Space size={[0, 8]} wrap>
                {item.qualifications.map((qualification) => (
                  <Tag
key={qualification.name}>{qualification.name}</Tag>
                ))}
              </Space>
            </Col>
          </Row>
        </Col>
      </Row>
    </Modal>
  );
}
```

```

    <Divider />
    <Tabs defaultActiveKey="1">
      <TabPane      tab={t("psychologistDetails.educationTab")}
key="1">
        <Timeline>
          {item.educations.map((education, index) => (
            <Timeline.Item key={index}>
              <p>
<strong>{t("psychologistDetails.education")}:</strong>{ " "}
                {t("psychologistDetails.educationDetails", {
                  specialization: education.specialization,
                  universityName: education.universityName,
                  startDate: education.startDate,
                  endDate: education.endDate,
                })}
              </p>
            </Timeline.Item>
          ))}
        </Timeline>
      </TabPane>
      <TabPane      tab={t("psychologistDetails.reviewsTab")}
key="2">
        <List
          itemLayout="horizontal"
          dataSource={item.reviews}
          renderItem={(item, index) => (
            <List.Item
              actions={[
                <Rate
                  allowHalf
                  style={{ fontSize: 14 }}
                  disabled
                  defaultValue={item.rating}
                />,
              ]}
            >
              <List.Item.Meta
                avatar={
                  <Avatar
src={`https://xsgames.co/randomusers/avatar.php?g=pixel&key=${index}`}
                  />
                }
                title={
                  <div>
                    {t("psychologistDetails.clientName", {
                      firstName: item.client.firstName,
                      lastName: item.client.lastName,
                    })}
                  </div>
                }
              </List.Item.Meta>
            </List.Item>
          )}
        </List>
      </TabPane>
    </Tabs>
  </div>

```



```

                </div>
            }
            description={item.comment}
        />
    </List.Item>
    )}
  />
  {item.reviews.length === 0 && (
    <p>{t("psychologistDetails.noReviews")}</p>
  )}
</TabPane>
</Tabs>
</Modal>
);
};

export default PsychologistDetailsModal;

```

Лістинг файлу PsychologistList.jsx:

```

import React, { useState } from "react";
import { List } from "antd";
import PsychologistsListItem from "../PsychologistsListItem";
import PsychologistDetailsModal from "../PsychologistDetailsModal";

const PsychologistsList = ({ data, fetchData }) => {
  const [modalVisible, setModalVisible] = useState(false);
  const [item, selectItem] = useState(null);

  const handleShowModal = (item) => {
    selectItem(item);
    setModalVisible(true);
  };

  const handleCancel = () => {
    setModalVisible(false);
  };

  return (
    <div>
      <List
        itemLayout="vertical"
        size="large"
        pagination={{
          onChange: (page) => {
            fetchData(page);
          },
          pageSize: 5,
        }}
        dataSource={data}

```

```

        renderItem={(item) => (
          <PsychologistsListItem
            item={item}
            handleShowModal={handleShowModal}
          />
        )}
      />
    {item && (
      <PsychologistDetailsModal
        visible={modalVisible}
        onCancel={handleCancel}
        item={item}
      />
    )}
  </div>
);
};

export default PsychologistsList;

import React from "react";
import { Button, List, Rate, Space, Tag } from "antd";
import { CheckCircleFilled, MessageOutlined } from "@ant-
design/icons";

const IconText = ({ icon, text }) => (
  <Space>
    {React.createElement(icon)}
    {text}
  </Space>
);

```

Лістинг файлу PsychologistsListItem.jsx:

```

const PsychologistsListItem = ({ item, handleShowModal }) => (
  <List.Item
    key={item.id}
    actions={[
      <Rate
        allowHalf
        style={{ fontSize: 14 }}
        disabled
        defaultValue={
          item.reviews
            .map((review) => review.rating)
            .reduce((sum, rating) => sum + rating, 0) /
          item.reviews.length
        }
      />,
      <IconText
        icon={MessageOutlined}
        text={item.reviews.length} />,
    ]}
  />
);

```

```

    <Button type="link" onClick={() => handleShowModal(item)}>
      Show more
    </Button>,
  ]}
  extra={
    
  }
}
>
<List.Item.Meta
  title={
    <div>
      {item.firstName} {item.lastName}{" "}
      {item.isVerified && <CheckCircleFilled />}
    </div>
  }
  description={
    <Space size={[0, 8]} wrap>
      {item.qualifications.map((qualification) => (
        <Tag
          key={qualification.name}>{qualification.name}</Tag>
        ))}
    </Space>
  }
/>
  {item.description}
</List.Item>
);

export default PsychologistsListItem;

```

Лістинг файлу Psychologists.jsx:

```

import React, { useState, useEffect } from "react";
import { Input } from "antd";
import axios from "axios";
import PsychologistsList from "./PsychologistsList";

const Psychologists = () => {
  const [searchText, setSearchText] = useState("");
  const [psychologistsData, setPsychologistsData] = useState([]);
  const [loading, setLoading] = useState(false);

  const fetchData = async (searchText) => {
    setLoading(true);
    try {

```

```

        const response = await
axios.get("https://psychologistwebapi20231216213946.azurewebsites.net
/api/psychologists", {
    params: { searchText },
});
setPsychologistsData(response.data);
} catch (error) {
    console.error("Error fetching data:", error);
} finally {
    setLoading(false);
}
};

useEffect(() => {
    fetchData(searchText);
}, [searchText]);

const handleSearch = () => fetchData(searchText);

return (
    <div>
        <Input.Search
            placeholder="Input search text"
            allowClear
            enterButton="Search"
            style={{ padding: "0px 24px", width: 500 }}
            value={searchText}
            onChange={(e) => setSearchText(e.target.value)}
            onSearch={handleSearch}
        />
        <PsychologistsList data={psychologistsData}
loading={loading} />
    </div>
);
};

export default Psychologists;

```