

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Розробка інформаційного забезпечення автоматизованої системи оцінки  
психологічного стану»

САХНО ОЛЕКСАНДР ГЕННАДІЙОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т. А.

„\_\_\_” \_\_\_\_\_ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Розробка інформаційного забезпечення автоматизованої системи оцінки  
психологічного стану»

Виконав: студент 4-го курсу, групи КНс-21

Спеціальності: 122 «Комп'ютерні науки»

Спеціалізація: «Інформаційні управляючі  
системи і технології»

(шифр і назва напрямку підготовки, спеціальності)

Сахно О. Г.

(прізвище та ініціали)

Керівник к.т.н., доц. Горда О. В.

(прізвище та ініціали)

Рецензент к.т.н., доц. Шабала Є. Є.

(прізвище та ініціали)

Київ, 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «бакалавр» за ОП

Спеціальність: 122 «Комп'ютерні науки»

Спеціалізація: Інформаційні управляючі системи і технології

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІТ

к.т.н. Гончаренко Т.А

„ 12 ” лютого 2024 року

**З А В Д А Н Н Я  
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Сахно Олександр Геннадійович

Тема роботи: Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану

затверджена наказом ректора КНУБА № 2650/2 від « 12 » лютого 2024 р.

2. Керівник роботи: Горда Олена Володимирівна, к.т.н, доцент

кафедри інформаційних технологій проектування та прикладної математики

3. Строк подання студентом роботи до захисту: \_\_\_\_\_

4. Зміст пояснювальної записки за розділами:

Р.1. Аналіз предметної області та постановка задачі

Р.2. Проектування інформаційного забезпечення

Р.3. Практична реалізація

Р.4. Бізнес-план

5. Інформаційні слайди:

С.1. Вступ

С.2. Дерево основних цілей

С.3. Порівняння розробок систем оцінки психичного стану

С.4. Порівняння баз даних

- C.5. Дерево функцій
- C.6. Концептуальна модель представлення
- C.7. Логічна модель представлення
- C.8. Фізична модель представлення
- C.9. Вибір програмного інструментарію
- C.10. Створення бази даних
- C.11. Інструменти взаємодії з базою даних
- C.12. Створення індексів
- C.13. UML-діаграма класів
- C.14. Процес ініціалізації даних
- C.15. Висновки

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Січень 2024 р.
Р. 2. Алгоритмічне та математичне забезпечення	Лютий 2024 р.
Р. 3. Розробка програмного забезпечення	Березень 2024 р.
Тестовий приклад програми	Квітень 2024 р.
Р. 4. Бізнес план	Травень 2024 р.
Остаточне оформлення роботи	Травень 2024 р.
Направлення роботи на рецензування	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

#### 7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Ергономіка інформаційних технологій	к.т.н., доц. Рябчун Ю.В.		
Прийом програмного продукту	к.т.н., доц. Шабала Є.Є.		

8. Дата видачі завдання: 12 лютого 2024 р.

Керівник

(підпис)

Горда О.В.

(прізвище та ініціали)

Бакалавр

(підпис)

Сахно О. Г.

(прізвище та ініціали)

## АНОТАЦІЯ

Сахно О.Г. Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану.

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», спеціалізація: «Інформаційні управляючі системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Бакалаврська робота присвячена дослідженню основних тенденцій, принципів та реалізації інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини, яка покликана полегшити процес діагностики психічних захворювань. Інструментом реалізації інформаційного забезпечення обрана нереляційна документоорієнтована система управління базою даних MongoDB.

Ключові слова: інформаційне забезпечення, база даних, психологічний стан, MongoDB.

## SUMMARY

Sakhno O.G. Development of information support for an automated system for assessing the psychological state.

Bachelor's thesis for a bachelor's degree in speciality: 122 'Computer Science', specialisation: 'Information management systems and technologies.' - Kyiv National University of Construction and Architecture - Kyiv, 2024.

The bachelor's thesis is devoted to the study of the main trends, principles and implementation of information support for an automated system for assessing the psychological state of a person, which is designed to facilitate the process of diagnosing mental illness. The tool for implementing information support is the non-relational document-oriented database management system MongoDB.

Keywords: information support, database, psychological state, MongoDB.

## ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ .....	9
1.1 Постановка та аналіз проблеми .....	9
1.2 Дерево цілей.....	12
1.3 Вимоги та особливості проектування системи .....	13
1.4 Аналіз існуючих розробок.....	15
1.4.1 Аналіз існуючих розробок автоматизованих систем оцінки психологічного стану.....	15
1.4.2 Аналіз існуючих баз даних.....	18
1.5 Постановка задачі.....	22
Вхідні дані:.....	23
Вихідні дані:.....	23
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ .....	24
2.1 Основні питання проектування інформаційного забезпечення .....	24
2.2 Дерево функцій.....	25
2.3 Вибір типу бази даних .....	28
2.3.1 Аналіз реляційних баз даних.....	28
2.3.2 Аналіз нереляційних баз даних.....	30
2.4 Вибір розміщення бази даних .....	33
2.5 Проектування моделі бази даних.....	34
2.5.1 Створення концептуальної моделі бази даних.....	35
2.5.2 Створення логічної моделі бази даних .....	38
2.5.3 Створення фізичної моделі бази даних.....	39
2.6 Загальний вигляд сутностей бази даних .....	41
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	45
3.1 Вибір програмного інструментарію .....	45
3.1.1 Система управління базами даних .....	45
3.1.2 Вибір мови програмування .....	48
3.1.3 Вибір програмного середовища.....	50

3.2 Розробка бази даних.....	51
3.3 Інструменти взаємодії з базою даних.....	55
3.4 Створення індексів для пришвидшення запитів.....	58
3.5 Налаштування безпеки при взаємодії з базою даних.....	61
3.6 Створення класів документів.....	62
3.6 Розробка репозиторіїв для взаємодії з даними.....	65
3.8 Ініціалізація бази даних.....	67
4. БІЗНЕС ПЛАН.....	69
4.1 Впровадження продукту.....	69
4.1.1 Визначні риси по відношенню до продукту конкурентів.....	70
4.1.3. Очікувані основні фінансові результати.....	73
4.2 Проектований продукт, або вид послуг.....	75
4.2.1 Потреби продукту по відношенню до проєкту.....	75
4.2.2. Особливості і відмінні риси розроблюваного проєкту.....	75
4.2.3. Якісні показники продукту.....	76
4.3 Конкуренція.....	77
4.3.1. Найбільші виробники аналогічного продукту.....	77
4.3.2. Сутність предмету конкуренції.....	77
4.4 Організаційний план.....	78
4.4.1. Формування команди фахівців.....	78
4.4.2. Організаційна структура підприємства.....	80
4.5 Стратегія маркетингу.....	82
4.5.1 Схема розповсюдження товарів.....	82
4.5.2 Ціноутворення та реклама.....	82
4.6 Стратегія фінансування.....	84
4.6.1. Фінансове планування.....	84
4.6.2. Джерела фінансових ресурсів.....	84
ВИСНОВОК.....	86
Список використаних джерел.....	88
Додатки.....	90

## ВСТУП

Актуальність дослідження. У сучасному світі психічні захворювання стають все поширенішими. Після повномасштабного вторгнення Російської Федерації в Україну кількість людей, що страждають від таких захворювань, стрімко зростає через травматичні події, які вони пережили. Вчасна діагностика і виявлення цих захворювань є важливим етапом для успішного лікування психологічних проблем. Щоб полегшити процес діагностики і зробити його більш доступним, доцільно розробити автоматизовану систему оцінки психологічного стану людини. В основі автоматизованої діагностичної системи лежить база даних, що становить її інформаційне забезпечення. Тому розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану є актуальною темою дослідження, оскільки дана система дає змогу полегшити процес діагностики і зробити його більш доступним, що призведе до вчасного виявлення проблеми з психологічним станом людини і в результаті цього збільшить вірогідність видуження пацієнта.

Мета дослідження. Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини. Система повинна складатися з тестів, які будуть включати питання про симптоми і пережиті події. На основі отриманих даних буде формуватись оцінка поточного стану особи та надаватись відповідні рекомендації. Якщо система виявить, що людина має високий ризик розвитку психологічного захворювання, їй дадуть рекомендації подальшого обстеження у психіатра або психолога.

Об'єкт дослідження. База даних автоматизованої системи оцінки психологічного стану. Ця база даних є збором інформації про психологічний стан людини, яка може бути використана для оцінки психологічного здоров'я людини та виявлення можливих порушень психологічного стану.

Предмет дослідження. Методи моделі розробки бази даних автоматизованої системи оцінки психологічного стану. Цей предмет дослідження визначає, які саме

дані повинні бути включені до бази даних, як вони повинні бути структуровані та як вони можуть бути використані для оцінки психологічного стану людини.

Методика та завдання дослідження. Методика дослідження розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану базується на наступних кроках:

1. Аналіз літературних джерел та наукових публікацій, що стосуються аналізу існуючих методів та технологій оцінки психологічного стану людини, а також розроблених баз даних для зберігання даних про психічний стан.
2. Визначення вимог до функціональності системи на основі проведеного аналізу
3. Реалізація бази даних на основі визначених вимог.
4. Тестування бази даних для перевірки її відповідності вимогам.
5. Впровадження розробленої системи
6. Зібрання та аналіз результатів використання бази даних та визначення можливих напрямків її подальшого вдосконалення.

Практична значимість. Результати дослідження можуть бути використані для розробки та реалізації ефективних систем оцінки психологічного стану людини. Розроблена база даних буде використовуватися для зберігання даних про психічний стан, а також для обробки цих даних для виявлення проблем з психологічним здоров'ям людини.

Результати дослідження. Результатом дослідження є розробка бази даних для зберігання даних про психологічний стан людини, а також обробки цих даних для визначення психологічного стану. Результатом є також розробка оптимальної структури баз даних, що забезпечують ефективну обробку, зберігання та взаємодію різних даних в системі.

Короткий зміст розділів:

- У першому розділі частково описана загальна та теоретична інформація щодо відомостей, концепцій та проблематики розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану. Описується постановка та аналіз проблеми, дерево цілей, встановлюються

вимоги та особливості проектування системи, аналізуються існуючі розробки автоматизованих систем оцінки психологічного стану, проводиться аналіз існуючих рішень баз даних, встановлюється задача дипломного проекту.

- Другий розділ охоплює проектування інформаційного забезпечення, зокрема розгляд основних питань постановки завдання, опис дерева функцій, вибір між реляційною та нереляційною базою даних, а також вибір розміщення бази даних на хмарному сервері або локально. Також надається інформація про створення концептуальної, фізичної та логічної моделей бази даних та загальний опис сутностей.

- Третій розділ присвячений практичній реалізації завдання. Він надає теоретичні відомості про вибір системи управління базами даних, мови програмування та середовища програмування, що створює базу програмного інструментарію для подальшої практичної реалізації бази даних. Також розглядається розробка самої бази даних, методи роботи з нею, встановлення індексів для пришвидшення запитів, розробка репозиторіїв взаємодії з даними та ініціалізація бази даних.

- Четвертий розділ розглядає бізнес-план проекту. Він охоплює впровадження проекту, описується проєктований продукт або вид послуг, аналізується конкуренція в даній сфері розробки, формується організаційний план, а також розглядаються можливі стратегії фінансування.

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Постановка та аналіз проблеми

Для визначення напрямку даної роботи, необхідно спочатку визначити об'єкт дослідження і предмет дослідження, а також надати опис, за допомогою яких методів планується проводити дослідження і вирішення проблеми.

Розробка автоматизованої системи оцінки психологічного стану людини є новою та перспективною сферою людської діяльності, оскільки має потенціал для суттєвого покращення охорони здоров'я людини. Автоматизовані системи оцінки психологічного стану мають ряд переваг порівняно з традиційними методами. Вони можуть бути більш об'єктивними, точними та швидкими. Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану є важливим завданням, яке має потенціал для трансформації охорони здоров'я людини.

Актуальність розробки автоматизованої системи оцінки психологічного стану людини обумовлена кількома важливими чинниками, які впливають на сучасне суспільство. Перш за все, зростає кількість психічних захворювань та розладів, що стає серйозною проблемою для системи охорони здоров'я у багатьох країнах, включаючи Україну. Зокрема, після повномасштабного вторгнення Російської Федерації в Україну значно збільшилася кількість людей, які страждають від посттравматичного стресового розладу (ПТСР), депресій, тривожних розладів та інших психологічних проблем. Це підвищує навантаження на медичні установи та вимагає більш ефективних методів діагностики та лікування. Сучасні методи діагностики часто є дорогими та трудомісткими, що обмежує їх доступність для широких верств населення. Поліпшення психічного здоров'я населення має важливе соціальне значення, оскільки воно впливає на загальний рівень благополуччя, працездатність та якість життя людей. Створення доступної та ефективної системи діагностики психічних розладів сприятиме ранньому виявленню та лікуванню цих станів, що, у свою чергу, покращить соціально-економічні показники.

Поетапний процес розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану можливо представити у вигляді життєвого циклу, серед яких виділяють наступні основні етапи:

1. етап планування;
2. етап збору вимог;
3. етап проектування;
4. етап розробки програмного забезпечення;
5. етап тестування програмного забезпечення;
6. етап експлуатації та технічного обслуговування.

На етапі планування команда фахівців та зацікавлених сторін узгоджує всі деталі майбутнього продукту, відповідаючи на важливі питання. Яка мета майбутнього продукту? Які проблеми він має вирішувати? Навіщо створюється продукт? Після відповіді ці питання, в усіх учасників формується однакове і правильне бачення майбутнього продукту. Що полегшує розробку та мінімізує ризики при створенні ПЗ. [1]

Етап аналізу та збору вимог є одним із найважливіших етапів життєвого циклу розробки автоматизованої системи оцінки психологічного стану людини. На цьому етапі збираються всі необхідні дані для успішної розробки та впровадження системи.

Після того, як стали зрозумілими цілі та завдання системи, а також технології, які будуть використовуватися для її розробки, можна переходити до проектування. На цьому етапі проектується майбутня архітектура проекту у вибраній технології. Створюється адаптивний та зручний дизайн, продумується зв'язок front частини програми з сервером, розробляються модулі та продумується система безпеки ресурсу. У випадку проектування бази даних, важливо врахувати структуру даних, забезпечити цілісність та консистентність даних, а також передбачити можливість масштабування бази даних для обробки великих обсягів інформації.

Стадія розробки - це етап, на якому відбувається створення програмного продукту. На цьому етапі береться проектна документація, прототипи, дизайн та архітектура, і на основі їх створюють код, який реалізує всі функціональні

можливості системи. Завдання на цьому етапі розділені між членами команди відповідно до їхньої галузі спеціалізації. У рамках інформаційного забезпечення розробляються всі необхідні компоненти для збереження, обробки та забезпечення доступу до даних. Це включає створення бази даних, розробку програмного забезпечення для роботи з даними, налаштування засобів безпеки тощо. Результатом етапу розробки є готовий програмний продукт, який відповідає вимогам, визначеним на етапі аналізу та збору вимог.

На етапі тестування команда перевіряє, чи відповідає розроблена система вимогам, визначеним на етапі аналізу та збору вимог. Тестування включає в себе такі основні завдання:

- Перевірка функціональності системи;
- Перевірка ефективності системи;
- Перевірка безпеки системи.

Після того, як система протестована і готова до використання, її запускають в експлуатацію. Система стає доступною для користувачів, які можуть її використовувати для оцінки свого психологічного стану. Замовник збирає відгуки від користувачів, щоб зрозуміти, чи відповідає система їхнім потребам. Якщо в результаті збору відгуків виявляються помилки в системі, їх виправляють розробники.

Технічне обслуговування - це етап, на якому система підтримується в робочому стані і забезпечується її ефективне використання. На цьому етапі система постійно перевіряється на наявність помилок і проблем. Також на цьому етапі можуть вноситися зміни в систему для її поліпшення. Ці зміни можуть бути спрямовані на:

- усунення виявлених проблем;
- розширення функціональності системи;
- вдосконалення інтерфейсу користувача;
- підвищення продуктивності системи;
- забезпечення безпеки системи.

## 1.2 Дерево цілей

Основною метою є розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану. Дерево цілей опису даної мети наведено на рис. 1.1.

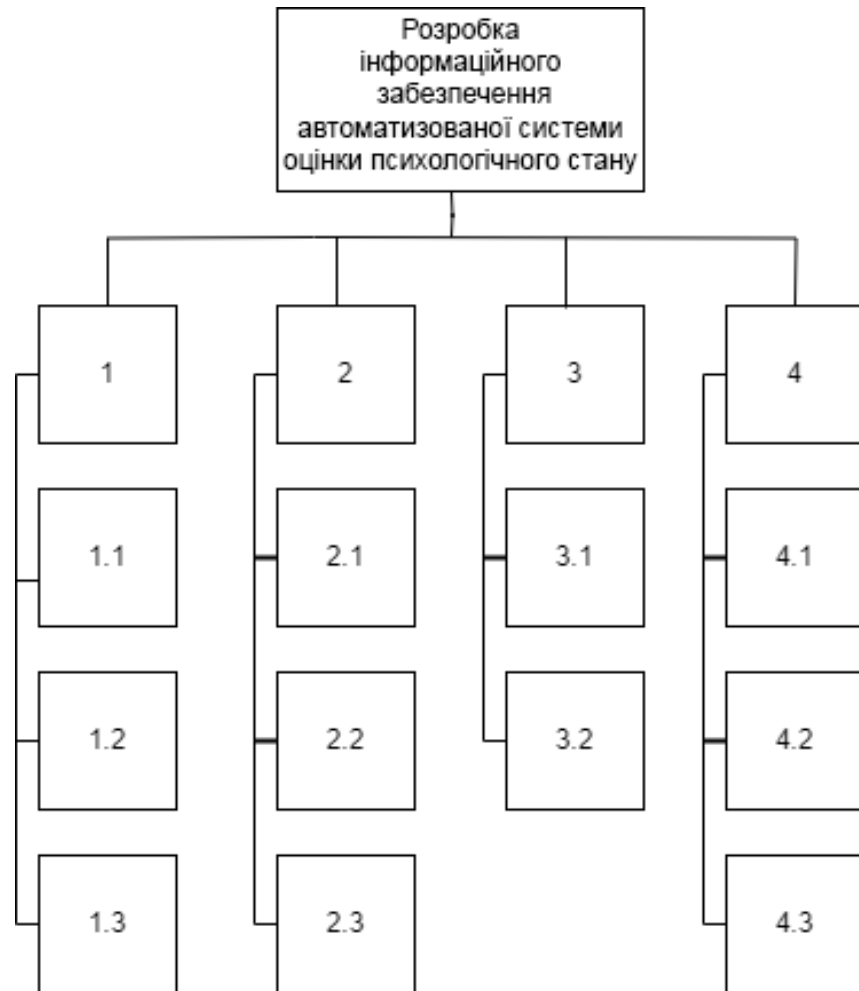


Рис. 1.1. Дерево цілей

1. Функціональні вимоги:
  - 1.1 Зберігання інформації про користувачів системи: імена, прізвища, адреса електронної пошти, дата народження, номер телефону.
  - 1.2 Зберігання інформації про результати тестів, які проходили користувачі: результати тестів на ПТСР, ПКС інші.

1.3 Зберігання інформації про взаємодію користувачів з системою: дати і часи доступу користувачів до системи, тести, які проходили користувачі системи.

2. Надійність та безпека

2.1. Розробка системи автентифікації та авторизації користувачів.

2.2. Захист даних користувачів та надану їм інформацію

2.3 Виявлення та виправлення можливих вразливостей системи безпеки та захисту даних.

3. Масштабованість та розширюваність

3.1 Розширення функціональності бази даних.

3.2 Розширення кількості даних.

4. Підтримка та обслуговування

4.1 Забезпечення підтримки та обслуговування бази даних протягом всього її життєвого циклу

4.2 Забезпечення можливості оновлення програмного забезпечення бази даних

4.3 Вирішення помилок у разі їх виявлення.

### **1.3 Вимоги та особливості проєктування системи**

Вимоги до баз даних поділяються на функціональні та нефункціональні. Функціональні вимоги визначають, які дані повинні зберігатися, як вони мають бути структуровані, і які операції мають підтримуватися. Для автоматизованої системи оцінки психологічного стану це може включати:

- зберігання інформації про користувачів;
- зберігання результатів психологічних тестів;
- зберігання історії змін;
- зберігання звітів та рекомендацій;
- виведення даних у різних розрізах, включаючи часові серії та індивідуальні профілі користувачів.

Нефункціональні вимоги до баз даних стосуються аспектів, які впливають на якість і надійність системи. Продуктивність бази даних є критично важливою, оскільки система повинна швидко обробляти великі обсяги даних і забезпечувати низький час відгуку на запити користувачів і внутрішніх процесів. Масштабованість бази даних також важлива, щоб забезпечити здатність системи обробляти зростаючі обсяги даних і підвищене навантаження. Це може вимагати:

- використання горизонтального або вертикального масштабування;
- ефективних методів індексації і кешування.

Надійність бази даних включає забезпечення її стійкості до збоїв і можливості відновлення даних у випадку аварійних ситуацій. Це досягається за рахунок:

- регулярного резервного копіювання;
- реплікації даних;
- використання методів відновлення після збоїв.

Безпека бази даних є ще одним важливим аспектом, особливо коли мова йде про чутливі дані, такі як результати психологічних оцінок. Безпека бази даних включає наступні вимоги:

- захист даних від несанкціонованого доступу;
- забезпечення конфіденційності;
- цілісність даних;
- управління доступом.

Особливості проєктування баз даних для інформаційного забезпечення автоматизованої системи оцінки психологічного стану включають вибір підходящої моделі даних. Вибір технології бази даних залежить від конкретних вимог системи, типу даних і очікуваних навантажень.

Архітектура бази даних має бути ретельно спланована для забезпечення ефективного зберігання і доступу до даних. Це включає нормалізацію даних для уникнення дублювання і забезпечення цілісності даних, а також денормалізацію в певних випадках для покращення продуктивності запитів.

Вибір технологій і інструментів для управління базами даних є критичним. Важливо враховувати сучасні тенденції і доступність ресурсів для підтримки обраних технологій. Інструменти для моніторингу і управління базами даних допомагають виявляти і вирішувати проблеми продуктивності, а також забезпечувати безперервний моніторинг безпеки даних.

Забезпечення високої якості даних включає впровадження методів тестування і верифікації даних. Автоматизовані тести для перевірки цілісності даних, правильності запитів і відповідності даних вимогам допомагають виявляти помилки на ранніх етапах і забезпечувати стабільність системи. Документація є невід'ємною частиною проектування баз даних. Вона включає опис структури бази даних, схем, індексів, процедур і тригерів, а також політик безпеки і процедур резервного копіювання.[2]

Проектування системи для інформаційного забезпечення автоматизованої системи оцінки психологічного стану є складним і багатоетапним процесом, який вимагає детального аналізу вимог, ретельного планування і врахування багатьох технічних аспектів. Від якості проектування баз даних залежить не тільки функціональність і ефективність системи, але й її здатність адаптуватися до змін і розвиватися разом з потребами користувачів. Успішне проектування потребує командної роботи, використання сучасних методів і інструментів, а також постійного вдосконалення.

## **1.4 Аналіз існуючих розробок**

### **1.4.1 Аналіз існуючих розробок автоматизованих систем оцінки психологічного стану**

У сучасному світі автоматизовані системи оцінки психологічного стану набувають все більшої актуальності та значущості для забезпечення ментального здоров'я населення. Вони дозволяють швидко та ефективно оцінювати психологічний стан, що сприяє ранньому виявленню та лікуванню різних психічних розладів. Ці системи надають можливість людям отримувати

діагностику та психологічну підтримку у зручний спосіб, використовуючи технології, що базуються на наукових методиках та дослідженнях. У цьому аналізі розглянуто найвідоміші автоматизовані системи оцінки психологічного стану, які вже доступні на ринку, що включають різноманітні методики та технологічні рішення для забезпечення надійної та точної діагностики.

1. Система «Я-ПСИХОЛОГ» – це інформаційна платформа для проведення психологічного тестування та діагностики онлайн. Її основна перевага полягає не лише у визначенні психологічного стану, але й у створенні індивідуальних психологічних характеристик для кожного користувача. Платформа використовує методики та тести, рекомендовані Міністерством освіти і науки, молоді та спорту, Міністерством охорони здоров'я, Міністерством оборони України та іншими важливими установами.

2. Проєкт MARTA, анонсований Міністерством охорони здоров'я у 2021 році під час зустрічі з представниками Apple, представляє інноваційну систему для психологічної підтримки та оцінки психічного здоров'я учасників бойових дій. Демонстраційна версія дозволила бійцям отримувати психологічну підтримку та оцінювати свій стан через смартфон або комп'ютер. Згодом MARTA стала доступною для всіх користувачів, пропонуючи онлайн-тести для оцінки психологічного стану.

3. Beck Depression Inventory (BDI) Online – це онлайн-версія відомого тесту Аарона Бека для визначення рівня депресії. Тест включає питання, спрямовані на виявлення різних аспектів депресивних симптомів. Його переваги полягають у науково обґрунтованій методиці та широкому застосуванні в клінічній практиці, хоча він вимагає від користувача високого рівня самосвідомості.

4. Anima – це онлайн-платформа, яка оцінює психологічне здоров'я за допомогою веб-камери. Під час трьоххвилинного тесту користувач дивиться на різні зображення (їжа, пейзажі, портрети тощо), а програма аналізує рухи очей і реакцію на зображення. Ця технологія допомагає визначити рівень тривожності чи депресії за допомогою математичних моделей, що фіксують рухи очей.

5. Сайт Самопоміч розроблено Національним інститутом психічного здоров'я Чеської Республіки в співпраці з іншими національними та міжнародними організаціями. Змістовне наповнення сайту було створене на основі наукових даних. Програма «Самопоміч» є спільним проектом з міжнародним фондом «Права людини в психічному здоров'ї». Він підтримується рядом міжнародних донорів, напр. Норвезький фонд дому прав людини, міністерство закордонних справ Нідерландів через посольство у Варшаві та декілька приватних донорів.[3]

Опис існуючих розробок автоматизованих систем оцінки психологічного стану представлено на табл. 1.1.

Таблиця 1.1. Опис існуючих розробок автоматизованих систем оцінки психологічного стану.

Назва	Метод оцінювання	Країна	Аудиторія	Особливості
«Я-ПСИХОЛОГ»	Тести з питаннями	Україна	Цивільні	Створення індивідуальних психологічних характеристик, використовуючи рекомендовані методики.
MARTA	Тести з питаннями	Україна	Військові	Створена для учасників бойових дій
Beck Depression Inventory (BDI) Online	Тести з питаннями	Чехія	Цивільні	Відома своєю науковою обґрунтованістю та широким застосуванням у клінічній практиці
Anima	Тест за допомогою веб-камери	Україна	Цивільні	Оцінка психологічного здоров'я за допомогою аналізу рухів очей під час перегляду зображень

Самопоміч	Тести з питаннями	Чехія	Цивільні	Розроблена на основі наукових даних Національним інститутом психологічного здоров'я Чеської Республіки.
-----------	-------------------	-------	----------	---

Порівнюючи різні системи тестування психологічного стану, можна виділити кілька ключових аспектів, які слід враховувати при розробці власної автоматизованої системи оцінки психологічного стану. Усі розглянуті системи використовують тести з питаннями як основний метод оцінювання. Однак Anima вирізняється використанням веб-камери для аналізу рухів очей, що може надати додаткову інформацію про реакцію користувача. MARTA спеціалізується на підтримці військових, що робить її специфічною для цього сегменту користувачів. У той час як "Я-ПСИХОЛОГ", Beck Depression Inventory, Anima та "Самопоміч" орієнтовані на цивільну аудиторію.

Розробка цих систем велась у різних країнах, що може впливати на особливості їх функціоналу та адаптацію до конкретного культурного і соціального середовища. Beck Depression Inventory та "Самопоміч" вирізняються науковою обґрунтованістю своїх методик, що підсилює достовірність наданої інформації. Всі ці фактори вказують на те, що вибір можливостей та особливостей для системи тестування психологічного стану має здійснюватися з урахуванням конкретних потреб користувачів, їх контексту та специфіки ситуації, для якої вони шукають рішення.

#### **1.4.2 Аналіз існуючих баз даних**

База даних — це організована сукупність даних, що зберігаються і керуються за допомогою комп'ютерних систем. Вона дозволяє користувачам ефективно зберігати, маніпулювати і витягувати інформацію, необхідну для різних завдань. Теоретично, база даних — це структурована колекція даних, яка забезпечує

швидкий і зручний доступ до них, що є ключовим для багатьох сфер діяльності, від бізнесу до науки.[4]

Будь яка база даних має такі важливі аспекти: організація даних, легкість доступу, керування та оновлення. База даних є фундаментальною складовою інформаційних систем, оскільки вона надає спосіб систематизовано зберігати інформацію таким чином, щоб її було легко використовувати.

Основна мета бази даних — забезпечити ефективне і надійне зберігання великої кількості даних, а також можливість швидкого доступу до них. Для цього використовуються різні методи організації даних, які дозволяють максимально оптимізувати процеси зберігання і пошуку. Незалежно від конкретної моделі даних, всі бази даних мають спільні характеристики, такі як:

1. Збереження даних: Дані зберігаються в електронному вигляді і доступні для тривалого зберігання.

2. Структурованість: Дані організовані у певний спосіб, що полегшує їх подальше використання. Це може бути таблиця, документ, граф або інша структура, але основна ідея полягає в наявності певної організації.

3. Маніпуляція даними: Бази даних дозволяють виконувати різні операції з даними, такі як додавання, оновлення, видалення та пошук. Це важливо для підтримки актуальності та точності інформації.

4. Доступність: Дані повинні бути доступні для користувачів і додатків, що їх потребують, із забезпеченням необхідного рівня безпеки і конфіденційності.

5. Управління даними: Системи управління базами даних (СУБД) забезпечують інструменти і засоби для керування даними, включаючи механізми для резервного копіювання, відновлення, а також контролю доступу.[5]

На ринку існує багато рішень для управління базами даних, кожне з яких має свої переваги та недоліки. У цьому розділі розглянуті деякі з найбільш поширених та ефективних рішень, доступних сьогодні, їх основні функції, а також можливості, які вони надають для покращення роботи з даними.

1. Oracle Database є однією з найпотужніших і наймасштабованіших систем управління базами даних. Вона підтримує широкий спектр застосувань, від бізнес-

аналітики до високошвидкісних транзакційних систем. Серед її переваг – масштабованість, висока доступність, а також великий набір інструментів для розробки та адміністрування баз даних. Недоліком може бути висока вартість ліцензій.

2. Microsoft SQL Server – популярна реляційна система управління базами даних, що підтримує різні типи даних та надає розширені інструменти для аналізу та візуалізації даних. Вона забезпечує високу продуктивність при роботі з великими обсягами транзакцій і пропонує засоби для масштабування та високої доступності. Основний мінус – інтеграція переважно з продуктами Microsoft.

3. MySQL – одна з найпоширеніших відкритих реляційних систем управління базами даних, особливо популярна серед розробників веб-додатків та електронної комерції. Вона відзначається своєю швидкістю, надійністю та простотою у використанні. MySQL підтримує масштабування та високу доступність, але може мати обмеження у функціональності порівняно з більш просунутими СУБД.

4. MongoDB – нереляційна система управління базами даних, яка дозволяє зберігати та обробляти документи в JSON-подібному форматі. Це робить її ідеальною для роботи з нереляційними даними та великими обсягами даних. MongoDB підтримує горизонтальне масштабування та високу доступність завдяки вбудованому механізму шардування. Одним з недоліків може бути складність у запитах порівняно з SQL.

5. PostgreSQL – потужна відкрита реляційна система управління базами даних, яка підтримує різноманітні типи даних та розширювану архітектуру. Вона забезпечує високу продуктивність, надійність та масштабованість, а також надає широкий набір інструментів для розробки та адміністрування. PostgreSQL особливо цінується за свою відповідність стандартам SQL та розширюваність.

6. SQLite – легка, безсерверна система управління базами даних, яка реалізує автономний, нульовий конфігураційний, транзакційний движок SQL. Вона є відкритою та безкоштовною для використання у приватних та комерційних цілях. SQLite ідеально підходить для вбудованих систем та мобільних додатків, але не

підходить для великих або розподілених систем через обмеження у масштабованості.

Вибір системи управління базами даних залежить від конкретних потреб користувачів, масштабів проекту та специфіки даних, з якими необхідно працювати. Кожна з розглянутих систем має свої унікальні можливості, що дозволяє знайти оптимальне рішення для будь-якого завдання.

Таблиця 1.2 Опис рішень баз даних для розробки автоматизованої системи оцінки психологічного стану.

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
Oracle Database	Relational	SQL	Windows, Linux, macOS	Підтримка великих обсягів даних, можливість використовувати більшість функцій SQL
MySQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, гарна підтримка індексів
MongoDB	NoSQL	MongoDB	Windows, Linux, macOS	Гнучкість у роботі зі структурованими та неструктурованими даними
PostgreSQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, можливість використовувати різні типи даних
SQLite	Relational	SQL	Windows, Linux, macOS	Легкість використання та налаштування, відсутність необхідності в окремому сервері для роботи з базою даних, швидкий доступ до даних

На основі порівняльного аналізу доступних баз даних для розробки автоматизованих систем оцінки психологічного стану можна зробити висновок, що кожна з них має свої переваги та недоліки. Вибір конкретної бази даних повинен здійснюватися відповідно до потреб проекту. Наприклад, для задач, що вимагають великої продуктивності та підтримки великої кількості користувачів, можна віддати перевагу PostgreSQL або MongoDB. З іншого боку, якщо основними критеріями є простота розробки та налаштування, то можна вибрати SQLite. Вибір бази даних також залежить від характеру даних, що будуть зберігатися, а також від вимог до безпеки та необхідності зберігання історії змін. Отже, важливо ретельно враховувати всі ці фактори при виборі бази даних для конкретного проекту. У розробці автоматизованої системи оцінки психологічного стану необхідно уважно враховувати ці аспекти та обирати ту базу даних, яка краще відповідає потребам проекту.

### **1.5 Постановка задачі**

Метою дипломного проекту є створення інформаційної системи та відповідної бази даних для автоматизованої системи оцінки психологічного стану. Ця система має надавати можливість зберігати дані, отримані з тестування, та надавати відповідні рекомендації на основі цих даних.

Основною задачею є розробка бази даних, яка буде зберігати всю необхідну інформацію для оцінки психологічного стану. Ця база даних повинна включати в себе дані про користувачів, тести з питаннями та відповідями, інформацію про проведені тести, рекомендації для осіб на основі результатів тестів, а також інформацію про симптоми.

Очікуваними результатами проекту є розроблена та налагоджена база даних для автоматизованої системи оцінки психологічного стану.

Основні завдання дипломного проекту:

1. Аналіз вимог користувачів та фахівців у галузі психологічного здоров'я для визначення потреб і функціональності системи, визначення типів даних, які потрібно зберігати в базі даних;
2. Вибір технології, вивчення особливостей та можливостей обраної технології;
3. Проектування схеми бази даних, визначення структури документів, які будуть зберігатися в базі даних;
4. Створення бази даних;
5. Наповнення даними;
6. Оптимізація;
7. Забезпечення безпеки;
8. Тестування системи;
9. Впровадження системи.

Для успішної реалізації проекту також важливо визначити вхідні та вихідні дані, які будуть оброблятися системою.

Вхідні дані:

1. Дані користувачів: інформація про пацієнтів (ім'я, вік, стать, контактна інформація).
2. Результати тестів: відповіді користувачів на психологічні тести.
3. Інформація про симптоми: дані про конкретні симптоми, які відчувають користувачі.
4. Рекомендації: рекомендації та висновки, надані спеціалістами на основі певного психологічного захворювання.
5. Дані про тести: опис тестів, включаючи інформацію про доступні переклади, структуру питань і відповідей.

Вихідні дані:

1. Збережені оцінки: результати аналізу тестів, що зберігаються у базі даних для подальшого використання.

2. Збережені рекомендації: індивідуальні рекомендації для користувачів на основі результатів тестів, що зберігаються для подальшого доступу.

## **2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ**

### **2.1 Основні питання проектування інформаційного забезпечення**

Для написання інформаційного забезпечення автоматизованої системи оцінки психологічного стану необхідно розглянути декілька ключових питань, таких як:

1. Визначення дерева функцій автоматизованої системи оцінки психологічного стану: це допомагає структурувати та організувати всі функції системи для досягнення її основної мети - оцінки психологічного стану користувачів. Дерево функцій визначає ієрархію функціональних блоків системи, а також взаємозв'язки між ними, що дозволяє зрозуміти, як система повинна працювати для ефективного виконання своїх завдань.

2. Визначення типу бази даних: це ключовий етап у розробці системи. Реляційні бази даних забезпечують структурованість та гнучкість у роботі з даними, тоді як нереляційні бази даних надають можливість зберігати та обробляти

різноманітні дані, такі як документи, графі чи ключ-значення. Вибір типу бази даних залежить від конкретних потреб проекту, його обсягу та швидкості розвитку.

3. Визначення хостингу для бази даних: вибір правильного хостингу для бази даних є критично важливим, оскільки він визначає доступність, надійність і продуктивність системи. Відповідно підібраний хостинг може забезпечити швидкий доступ до даних, регулярні резервні копії та захист від втрати даних, що важливо для успішної роботи системи.

4. Проектування моделі бази даних (концептуальна, фізична та логічні представлення): Для успішної реалізації інформаційного забезпечення та баз даних в автоматизованій системі оцінки психологічного стану, важливо визначити оптимальні концептуальну, логічну та фізичну форму представлення бази даних. Це дозволить забезпечити ефективну продуктивність та надійність зберігання даних.

## **2.2 Дерево функцій**

У контексті розробки інформаційного забезпечення та баз даних для автоматизованої системи оцінки психологічного стану, дерево функцій є важливим інструментом. Це методологічний засіб, який описує ієрархічну структуру функцій, необхідних для досягнення основної мети системи. Основна мета такої системи полягає в розробці інформаційного забезпечення автоматизованої системи оцінки психологічного стану осіб.

Функціональна декомпозиція системи ґрунтується на поділі її на окремі функціональні компоненти, кожен з яких відіграє конкретну роль у загальній структурі. В цьому контексті, основні компоненти можна розділити на декілька ключових функціональних блоків: збір інформації, зберігання та обробка даних, захист та безпека інформації.

Збір інформації включає підготовку та проведення тестування користувачів, збір результатів тестів, а також запити про симптоми психологічного стану і

пережиті події. Цей процес забезпечує збирання необхідних даних для подальшого аналізу і оцінки психологічного стану осіб. Зберігання та обробка даних передбачає збереження результатів тестів у базі даних, аналіз та класифікацію зібраної інформації, формування бази даних користувачів, а також надання доступу до результатів тестів користувачам і фахівцям. Це забезпечує ефективне управління та використання даних для підтримки та прийняття рішень. Захист та безпека інформації включає шифрування та зберігання паролів, контроль доступу до особистої інформації користувачів, а також захист бази даних від несанкціонованого доступу. Ці заходи гарантують конфіденційність і цілісність даних, забезпечуючи безпечне використання системи.

Таким чином, дерево функцій слугує важливим інструментом для структуризації та організації завдань всередині автоматизованої системи оцінки психологічного стану. Кожен елемент цієї ієрархії відображає конкретну функцію або завдання, яке спрямоване на досягнення загальної мети системи - створення інформаційного забезпечення системи оцінки психологічного стану.

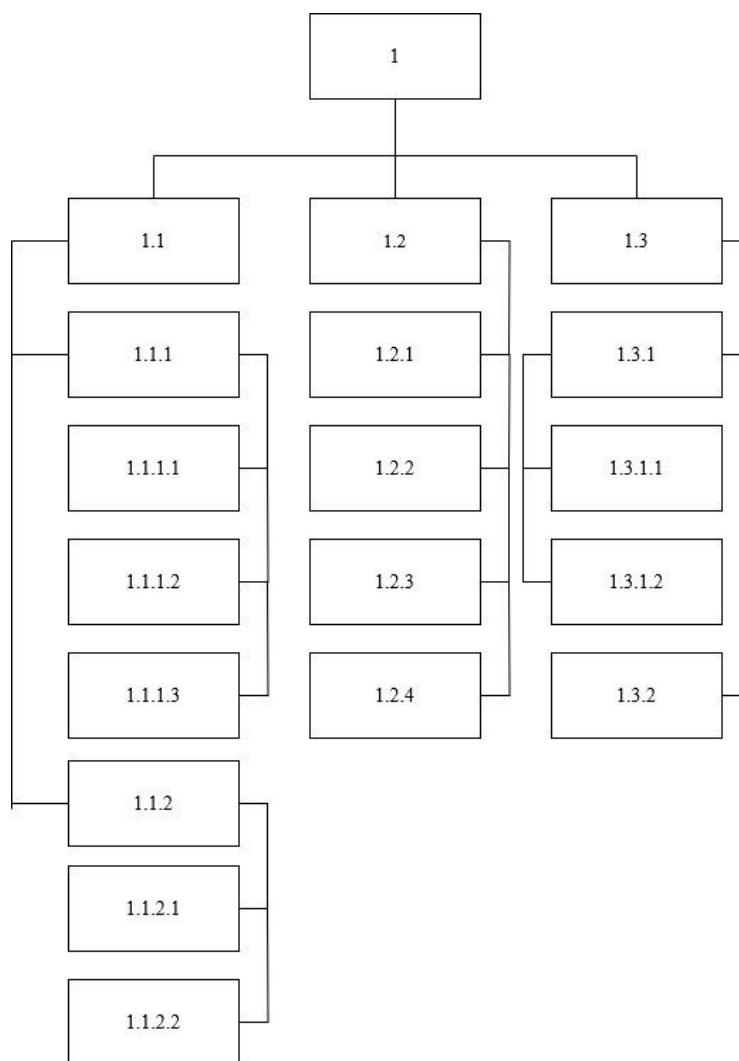


Рис. 2.1. Дерево функцій інформаційного забезпечення.

## 1 Створення інформаційного забезпечення системи оцінки психологічного стану

### 1.1 Збір інформації

#### 1.1.1 Збір даних з тестів

1.1.1.1 Підготовка тестів для оцінки психологічного стану

1.1.1.2 Проведення тестування користувачів

1.1.1.3 Збір результатів тестів

#### 1.1.2 Запити до користувачів

1.1.2.1 Запит про симптоми психологічного стану

1.1.2.2 Запит про пережиті події

### 1.2 Зберігання та обробка даних

- 1.2.1 Зберігання результатів тестів
- 1.2.2 Аналіз та класифікація даних з тестів
- 1.2.3 Формування бази даних користувачів системи
- 1.2.4 Надання доступу до результатів користувачам та фахівцям
- 1.3 Захист та безпека інформації
  - 1.3.1 Захист персональних даних користувачів
    - 1.3.1.1 Шифрування та зберігання паролів
    - 1.3.1.2 Контроль доступу до особистої інформації
  - 1.3.2 Захист бази даних від несанкціонованого доступу

## **2.3 Вибір типу бази даних**

Вибір оптимального типу бази даних є ключовим етапом у розробці інформаційного забезпечення для автоматизованої системи оцінки психологічного стану. Різні типи баз даних мають свої унікальні особливості та переваги, що можуть відігравати важливу роль у забезпеченні ефективності, швидкодії та безпеки системи. У цьому підрозділі буде проведений аналіз різних типів баз даних з урахуванням їхнього відповідності вимогам та характеристикам проєкту. Розглядатимуться реляційні та нереляційні бази даних, зокрема їхні особливості та потенційні переваги та обмеження у контексті системи оцінки психологічного стану.

### **2.3.1 Аналіз реляційних баз даних**

Реляційна база даних - це тип бази даних, яка зберігає та надає доступ до точок даних, пов'язаних між собою. Реляційні бази даних засновані на реляційній моделі - інтуїтивно зрозумілому, простому способі представлення даних у таблицях. У реляційній базі даних кожен рядок таблиці - це запис з унікальним ідентифікатором, який називається ключем. Стовпці таблиці містять атрибути

даних, і кожен запис зазвичай має значення для кожного атрибуту, що полегшує встановлення зв'язків між точками даних.[6]

Реляційна модель означає, що логічні структури даних - таблиці, подання та індекси - відокремлені від фізичних структур зберігання. Таке розділення означає, що адміністратори бази даних можуть керувати фізичним сховищем даних, не впливаючи на доступ до цих даних як до логічної структури. Наприклад, перейменування файлу бази даних не призводить до перейменування таблиць, що зберігаються в ньому.

Різниця між логічним і фізичним також стосується операцій над базами даних, які є чітко визначеними діями, що дозволяють додаткам маніпулювати даними і структурами бази даних. Логічні операції дозволяють програмі вказати, який вміст їй потрібен, а фізичні операції визначають, як отримати доступ до цих даних, і потім виконують поставлене завдання.

Основна перевага реляційної моделі даних полягає в можливості витягувати з даних значущу інформацію, наприклад підраховувати, сортувати та групувати дані в таблиці, а також виконувати математичні операції та створювати складні звіти. Реляційні бази даних залишаються найбільш популярною моделлю зберігання даних і сьогодні.

Попри те, що реляційні бази даних є стандартом у багатьох сферах, їх використання може мати певні обмеження та недоліки в контексті розробки системи оцінки психологічного стану. Наведені нижче недоліки варто врахувати при виборі типу бази даних для даного проекту.

1. **Складність моделювання даних:** Реляційні бази даних потребують чіткого та докладного моделювання даних за допомогою таблиць, стовпців та зв'язків між ними. У випадку складних або невизначених структур даних, моделювання може бути витратним за часом та ресурсами.

2. **Обмежена гнучкість у роботі з неструктурованими даними:** Реляційні бази даних оптимізовані для роботи зі структурованими даними, такими як числа, рядки або дати. Однак в системі оцінки психологічного стану можуть зустрічатися

неструктуровані дані, такі як тексти, аудіо- або відеозаписи, які можуть бути складніше зберігати та обробляти у реляційних базах даних.

3. **Масштабованість:** Реляційні бази даних можуть мати обмеження у масштабованості, особливо при великому обсязі даних або підвищених навантаженнях. Підтримка високої доступності та реплікації даних також може бути складною та витратною.

4. **Потреба в нормалізації:** Для забезпечення цілісності та ефективності бази даних часто потрібно проводити процес нормалізації, що може призвести до складності структури даних та високої складності запитів для отримання необхідної інформації.

5. **Витрати на обслуговування та розширення:** У порівнянні з нереляційними базами даних, розгортання, обслуговування та розширення реляційних баз може бути більш витратним та складним процесом, особливо у випадку зростаючого обсягу даних або зміни вимог.

### **2.3.2 Аналіз нереляційних баз даних**

Окрім вже традиційних реляційних баз даних, з кінця 2000-х років став популярним ще один метод зберігання даних. Нереляційна база даних — це база даних, у якій на відміну більшості традиційних систем баз даних не використовується таблична схема рядків і стовпців. У цих базах даних застосовується модель зберігання, оптимізована під конкретні вимоги типу даних, що зберігаються. Наприклад, дані можуть зберігатися як прості пари "ключ - значення", документи JSON або граф, що складається з ребер та вершин.

Термін NoSQL застосовується до сховищ даних, які не використовують мову запитів SQL. Натомість вони запитують дані за допомогою інших мов програмування та конструкцій. На практиці NoSQL означає "нереляційна база даних", навіть незважаючи на те, що багато з цих баз даних підтримують запити, сумісні з SQL. Однак базова стратегія виконання запитів SQL зазвичай значно відрізняється від застосовуваної системи управління реляційною базою даних. [6]

Документоорієнтовані бази даних:

Цей тип баз даних використовує структуровані формати, такі як XML, JSON або BSON, для зберігання даних. Кожен документ може мати різний набір властивостей і представляє собою самодостатню одиницю інформації. Такі бази даних ідеально підходять для швидкої розробки систем, які працюють з різноманітними структурованими даними. MongoDB, RavenDB, CouchDB та DocumentDB є прикладами таких баз даних. [8]

Бази даних "Ключ-значення":

Це найпростіший тип нереляційних баз даних, де дані зберігаються як словник, а ключі використовуються для доступу до значень. Ці бази даних дозволяють зберігати та обробляти різні типи даних, такі як файли, рядки, текст, числа, JSON-об'єкти тощо. Вони відзначаються високою швидкістю доступу до даних завдяки адресному зберігання. Приклади цього типу баз даних включають DynamoDB, Redis, Riak, LevelDB та Memcached.

Графові бази даних:

Цей тип баз даних призначений для моделювання складних відносин за допомогою графів, де об'єкти представлені вузлами, а зв'язки між ними - ребрами графа. Графові бази даних знаходять застосування у соціальних мережах, аналізі профілів користувачів, виявленні шахрайства та інших областях. Вони відзначаються високою продуктивністю, оскільки обхід ребер і вершин графа здійснюється швидше, ніж аналіз безлічі таблиць у реляційних базах даних. Приклади графових баз даних - Neo4J, JanusGraph, Dgraph та OrientDB.

Ці типи нереляційних баз даних надають розробникам широкі можливості для реалізації різноманітних проєктів, забезпечуючи високу швидкість та гнучкість у роботі з даними.

Для розробки інформаційного забезпечення системи оцінки психологічного стану було обрано нереляційний тип бази даних, а саме документоорієнтований тип. Цей вибір зумовлений рядом переваг, які надає документоорієнтована база даних у контексті вимог та особливостей даної системи.

Документоорієнтовані бази даних дозволяють зберігати дані у вигляді документів JSON. Це означає, що структура даних може бути динамічною і легко змінюватись, що є важливим у випадках, коли вимоги до даних можуть змінюватись або доповнюватись новими елементами. Наприклад, нові типи тестів або змінені параметри користувачів можуть бути легко додані без необхідності модифікації існуючих структур бази даних.

У системі оцінки психологічного стану дані можуть включати різноманітну інформацію, таку як відповіді на питання, результати тестів, особисті дані користувачів. Документоорієнтована база даних дозволяє зберігати ці різноманітні дані в одному документі, що спрощує їх зберігання і доступ до них.

Документоорієнтовані бази даних зазвичай мають високу горизонтальну масштабованість. Це означає, що по мірі зростання обсягу даних можна легко додавати нові сервери для розподілу навантаження. Для системи, яка може мати великий обсяг даних користувачів і їхніх оцінок, така масштабованість є критичною.

Завдяки відсутності складних зв'язків між таблицями, які характерні для реляційних баз даних, документоорієнтовані бази даних забезпечують швидкий доступ до даних. Це особливо важливо для системи оцінки психологічного стану, де важливо забезпечити швидке реагування на запити користувачів і оперативне оновлення даних.

Документоорієнтовані бази даних добре інтегруються з сучасними веб- та мобільними додатками. Це полегшує розробку інтерфейсів та сервісів, які будуть використовувати базу даних для зберігання та обробки інформації про користувачів і результати тестів.

Вибір документоорієнтованої бази даних для системи оцінки психологічного стану користувачів обумовлений її гнучкістю, масштабованістю, швидкістю операцій та легкістю інтеграції. Ці переваги дозволяють ефективно зберігати та обробляти різноманітні дані, швидко реагувати на зміни та забезпечувати надійну роботу системи в умовах великих обсягів даних.

## 2.4 Вибір розміщення бази даних

Розміщення баз даних є критично важливою складовою будь-якої інформаційної системи, оскільки ці дані становлять основу системи. Один із способів забезпечити доступність та надійність баз даних – це розміщення бази даних на власному сервері. Але саме цей варіант може мати кілька недоліків. По-перше, це може вимагати значних витрат на придбання та підтримку обладнання, програмного забезпечення та інфраструктури. Крім того, такий підхід може призвести до складнощів у встановленні і підтримці безпеки, оновлення програмного забезпечення та резервного копіювання даних. Також, власний сервер може не забезпечити необхідного рівня доступності та масштабованості, особливо при збільшенні обсягу даних або трафіку.

У таких випадках використання хостингу може бути більш ефективним рішенням, оскільки воно забезпечить надійну інфраструктуру, безпеку та масштабованість за відповідну плату.

Хостинг баз даних виник як альтернатива запуску бази даних і додатків до неї в стандартному локальному середовищі, на обчислювальній інфраструктурі, що належить і експлуатується самою організацією. При хостингу баз даних третя сторона пропонує обладнання та інфраструктуру для запуску бази даних на вибір клієнта, часто в хмарі. Вони також налаштовують середовище для безпечного доступу, забезпечують наявність ресурсів для масштабування бази даних за потреби та пропонують керовані послуги відповідно до вимог.

Хостинг баз даних має багато переваг:

- Можливість налаштувати базу даних, що використовується, а також рівень підтримки та обслуговування.
- Автоматичне масштабування, як збільшення, так і зменшення на основі попиту.
- Пропонує високу доступність, що означає відсутність простоїв для користувачів.
- Часто надають плани реагування на інциденти, які включають моніторинг.

– Не потрібно платити за дороге обладнання та обслуговувати його.

Провайдери хостингу баз даних є в достатку у всьому світі. Найбільшими провайдерами хостингу є Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP), а також багато місцевих та регіональних провайдерів.

При виборі слід враховувати різні фактори, такі як вартість, місцезнаходження та доступні послуги.

При розгляді варіантів хостингу для бази даних, одним з розглянутих варіантів було використання хмарного хостингу, так званого Database-as-a-Service (DBaaS).

DBaaS - це послуга хмарних обчислень, що надає доступ до бази даних без необхідності встановлювати фізичне обладнання, інсталиювати програмне забезпечення або налаштовувати базу даних. Натомість постачальник послуг виконує більшість завдань з адміністрування та обслуговування бази даних. Користувачі можуть швидко запускати базу даних, завантажувати та аналізувати дані, як правило, з невеликим втручанням або без нього. Варіанти хостингу баз даних доступні для всіх типів баз даних, включаючи й нереляційний тип документоорієнтованих баз даних.[7]

## **2.5 Проєктування моделі бази даних**

База даних інформаційного забезпечення системи оцінки психологічного стану містить 3 рівні представлення: концептуальний, фізичний та логічний.

- Концептуальна форма представлення бази даних описує зв'язки між сутностями та їх атрибутами в системі. Це допомагає визначити, які сутності слід включити до бази даних, і зв'язки між ними. Концептуальні моделі допомагають зрозуміти, які дані потрібні системі, і допомагають уникнути надлишкових даних. У випадку проєктування даної системи, використання нереляційних баз даних призводить до того, що зв'язок між сутностями (колекціями) буде відсутнім. Тому буде представлено лише зв'язок колекції з її об'єктами, оскільки в нереляційних

базах даних дані зберігаються без чіткої взаємозв'язаності між сутностями (колекціями) та ключові атрибути об'єктів.

- Логічна форма представлення бази даних у нереляційних системах, описує структуру документів в колекціях. На цьому етапі визначаються не лише колекції, які потрібно створити, але й атрибути (поля), що будуть використовуватись для зберігання даних в цих колекціях. Це дозволяє встановити структуру даних і зв'язки між ними таким чином, щоб забезпечити ефективну обробку та доступ до інформації в майбутньому.

- Фізична форма представлення бази даних доповнює логічну форму представлення та описує, які типи даних повинні використовуватись для зберігання даних.

Наступні розділи роботи наводять детальний опис кожного рівня представлення бази даних інформаційного забезпечення системи оцінки психологічного стану.

### **2.5.1 Створення концептуальної моделі бази даних**

Концептуальне представлення бази даних системи оцінки психологічного стану повинно починатись з аналізу предметної області. Основне завдання аналізу полягає у визначенні об'єктів та їх взаємозв'язків в контексті системи оцінки психологічного стану. Для досягнення цієї мети були вибрані такі об'єкти, як дані про користувачів, питання, відповіді, та рекомендації. Що привело до створення таких сутностей як "користувачі", "тести" та "рекомендації".

Далі були розроблені структури зв'язків між цими сутностями для кожної окремо, оскільки в документ-орієнтованих базах даних дані зберігаються без жорсткої взаємозв'язаності між сутностями.

Структура зв'язків сутності "Користувач" включає зв'язок з "Особистою інформацією про користувача", "Датою народження" та "Результатами проходження тестів", як показано на рисунку 2.1.



Рис. 2.1. Структура зв'язків сутності «Користувач»

На концептуальній моделі опису зв'язків сутності «Тести» представлено зв'язок між сутністю «Тести» і її об'єктами «Групи тестів», «Питання», «Переклади питань» та «Відповіді», «Переклади відповідей». Концептуальна модель сутності «Тести» представлена на рис. 2.2.



Рис. 2.2. Структура зв'язків сутності «Тести»

На концептуальній моделі опису зв'язків сутності «Рекомендації» представлено зв'язок між сутністю «Рекомендації» і її об'єктом «Переклади рекомендацій». Концептуальна модель сутності «Рекомендації» наведена на рис. 2.2.

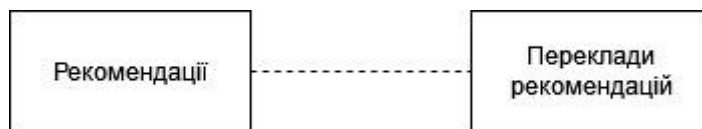


Рис. 2.3. Структура зв'язків сутності «Рекомендації»

Також треба визначити основні атрибути кожного об'єкту. Детальний опис кожного атрибуту показано на рис. 2.4.

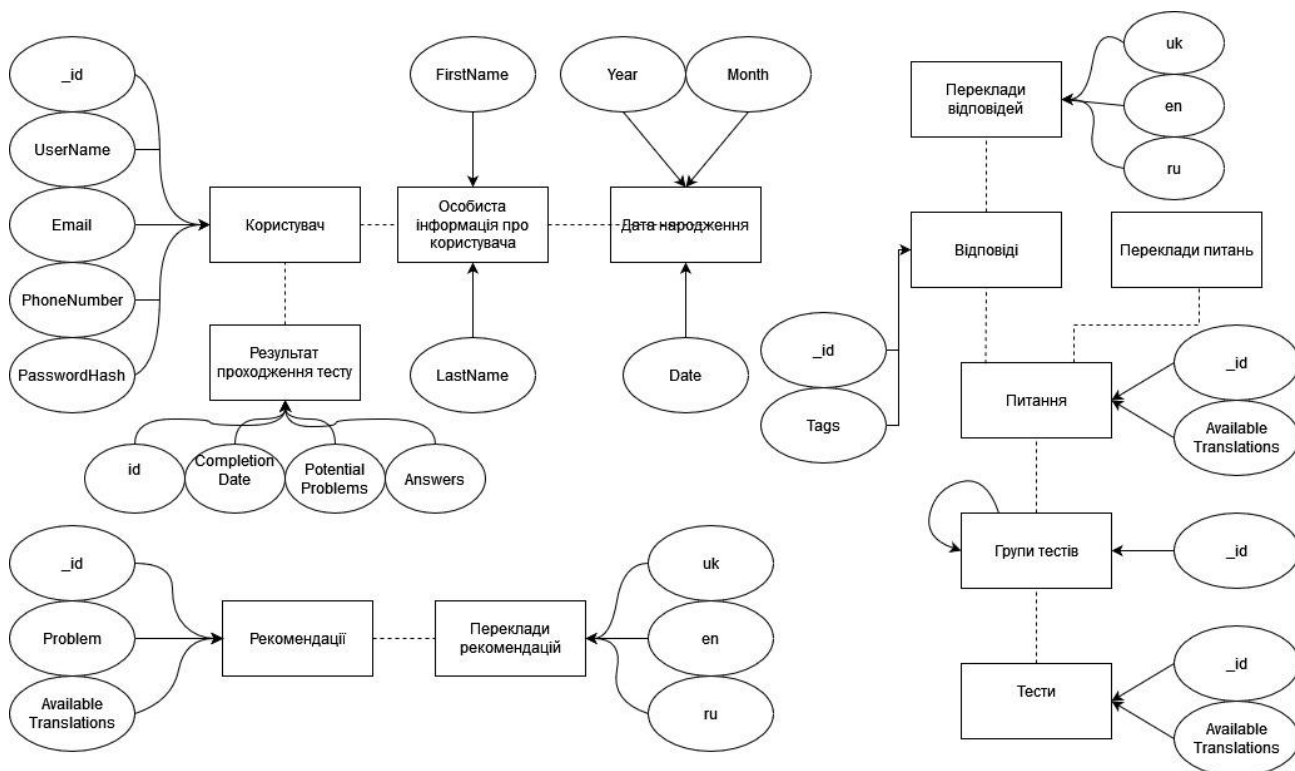


Рис. 2.4. Концептуальна модель опису зв'язків бази даних

Завершуючи розробку концептуального представлення бази даних системи оцінки психологічного стану людини, були визначені всі сутності та їх



На етапі створення логічного представлення для системи оцінки психологічного стану людини визначено сутності, усі атрибути та зв'язки між ними, згідно з концептуальною моделлю. При цьому були враховані особливості використовуваної системи управління базою даних. Цей етап забезпечив структуроване уявлення про необхідні дані та їх взаємодію в контексті розроблюваної системи.

Під час створення логічного представлення бази даних системи оцінки психологічного стану визначені обмеження цілісності даних, такі як вимоги до обов'язковості заповнення певних атрибутів, обмеження на діапазон значень атрибутів та обмеження на видалення записів, що мають зв'язки з іншими записами. Після завершення цього етапу були визначені обмеження цілісності даних, які впроваджено в програмуванні проєкту. Це сприяло створенню структури бази даних, яка відповідає концептуальній моделі та потребам системи управління базою даних, і дозволило перейти до подальшого створення фізичного представлення.

### **2.5.3 Створення фізичної моделі бази даних**

Після закінчення проєктування логічної моделі представлення бази даних наступним етапом є створення фізичного представлення. Це етап передбачає перетворення логічної моделі в модель, яка відображає реальні характеристики фізичної реалізації бази даних та типів даних.

Під час створення фізичного представлення визначена структура та параметри фізичної реалізації бази даних. Цей процес включав налаштування параметрів бази даних та визначення способу зберігання даних. Додатково на етапі проєктування фізичного представлення бази даних визначені такі параметри забезпечення безпеки даних: управління користувачами, авторизація, двофакторна автентифікація та шифрування даних.

Детальний опис фізичної моделі представлення наведено на рис. 2.6.

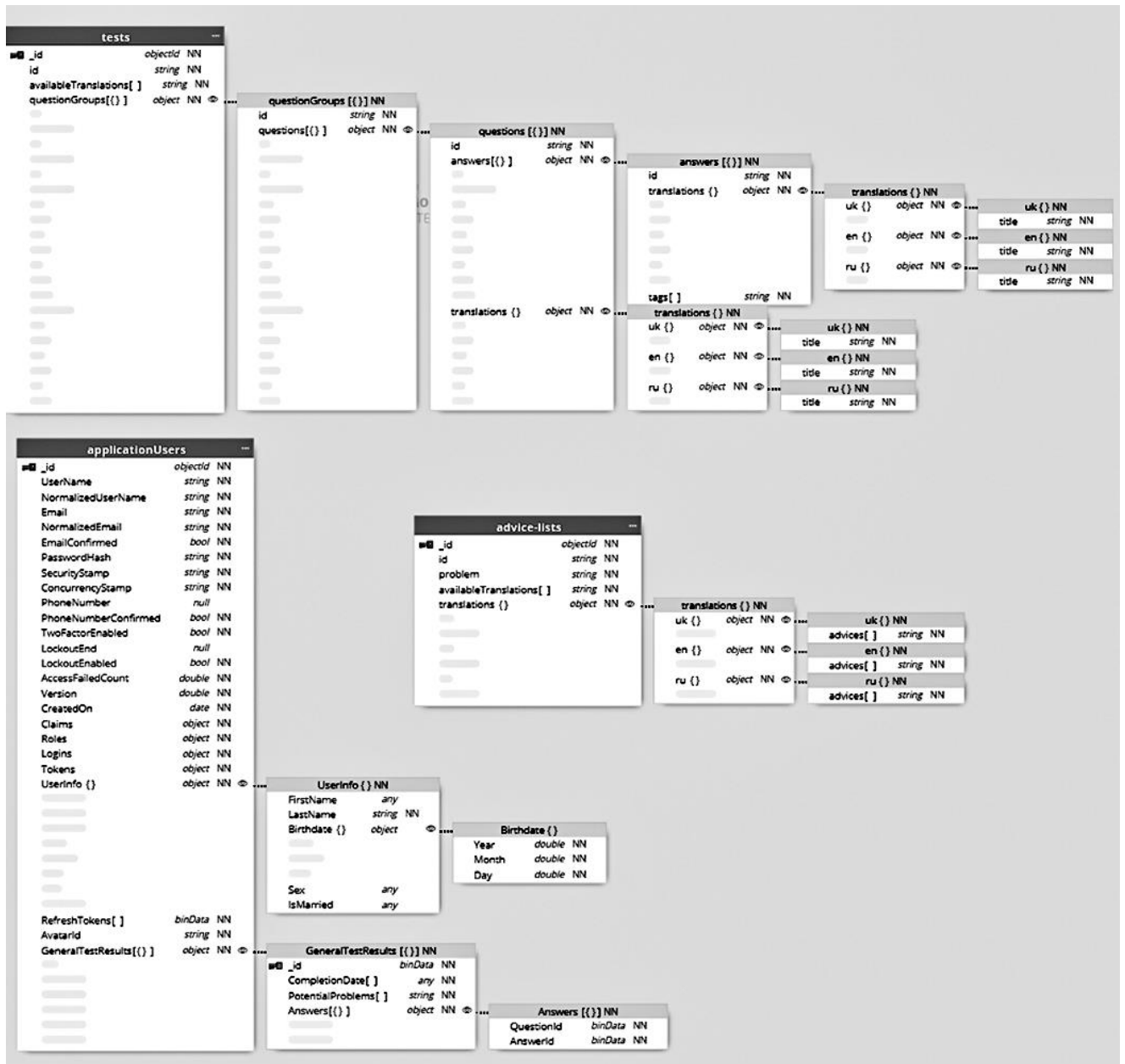


Рис. 2.6. Фізична модель бази даних

Управління користувачами: важливо мати можливість створювати нові облікові записи користувачів та керувати існуючими. Це включає зберігання особистих даних. З метою безпеки паролі повинні зберігатися у вигляді хешів, що ускладнює їх розшифровку у разі витоку даних.

Авторизація: управління ролями та правами доступу необхідні для визначення ролі користувачів (наприклад, адміністратор, користувач) та надати різні рівні доступу до функцій системи залежно від цих ролей. Це допомагає контролювати, які дії можуть виконувати користувачі, підвищуючи безпеку

системи. Також можна створювати складні правила доступу на основі різних критеріїв, таких як підтвердження електронної пошти.

Двофакторна автентифікація: Додатковий рівень безпеки, що вимагає введення додаткового коду (наприклад, отриманого через SMS або генератор кодів) на додаток до пароля. Це значно ускладнює несанкціонований доступ до облікових записів, навіть якщо зловмисник отримав пароль.

Шифрування даних: Забезпечення конфіденційності даних користувачів за допомогою шифрування як на рівні зберігання даних, так і під час їх передачі. Це допомагає захистити дані від несанкціонованого доступу.

Завершуючи створення фізичного представлення бази даних для системи оцінки психологічного стану користувачів, забезпечено відповідні параметри безпеки та конфіденційності даних. Також здійснена правильна інтеграція з системою управління базою даних, щоб забезпечити надійність та безпеку збереження і обробки інформації.

## **2.6 Загальний вигляд сутностей бази даних**

Після завершення процесу проектування концептуальної, логічної та фізичної моделей бази даних, було сформовано загальний набір сутностей, що слугуватиме основою для подальшої розробки та програмування інформаційного забезпечення.

Колекція «Користувачі» (`applicationUsers`) містить в собі документи з інформацією про користувачів. Кожен користувач представляє собою окремий документ. Документи мають таку структуру:

`_id (objectId)` – унікальний ідентифікатор документу (користувача);

`UserName (string)` - ім'я користувача;

`NormalizedUserName (string)` - нормалізоване ім'я користувача;

`Email (string)` - електронна пошта користувача;

`NormalizedEmail (string)` - нормалізована електронна пошта;

`EmailConfirmed (bool)` – наявність підтвердження електронної пошти;

PasswordHash (string) - хеш паролю;

SecurityStamp (string) – штамп безпеки;

ConcurrencyStamp (string) - штамп конкурентності;

PhoneNumber (int) - номер телефону;

PhoneNumberConfirmed (bool) – наявність підтвердження номера телефону;

TwoFactorEnabled (bool) - статус двофакторної аутентифікації;

LockoutEnd (date) – дату та час закінчення блокування;

LockoutEnabled (bool) - можливість блокування;

AccessFailedCount (double) – кількість невдалих спроб доступу;

CreatedOn (date) - дата створення;

Claims (object) – містить різноманітну інформацію про користувача у формі ключ-значення, де ключі - це назви прав або властивостей, а значення - це їхні значення. Ці права можуть бути використані для визначення доступу користувача до різних ресурсів або функціональності програми;

Roles (object) – роль користувача, наприклад адміністратор;

Logins (object) – зберігає інформацію про зовнішні облікові записи користувача;

Tokens (object) – зберігає унікальний ключ, який надається користувачеві після успішної аутентифікації;

UserInfo (object) – об'єкт, що містить інформацію про користувача:

FirstName (string) - ім'я;

LastName (string) – прізвище;

Sex (bool) - стать

IsMarried (bool) - статус шлюбу

Birthdate (object) – об'єкт, що містить інформацію про дату народження:

Year (double) - рік

Month (double) - місяць

Day (double) - день

RefreshTokens (binData) – спеціальні токени, які використовуються для оновлення або поновлення доступу до ресурсів після закінчення строку дії основного токена доступу.

AvatarId (string) – унікальний ідентифікатор зображення профілю користувача;

GeneralTestResult (object) - об'єкт, що містить результати проходження тесту користувачем:

Id (string)- унікальний ідентифікатор результату;

CompletionDate (date) – час завершення проходження;

PotentialProblems (string) – виявлені проблеми;

Answers (object) – об'єкт, що зберігає відповіді користувача:

QuestionId (binData) – унікальний ідентифікатор питання;

AnswerId (binData) - унікальний ідентифікатор відповіді;

Колекція «Тести» (tests) має в собі документ з групами питань, що містять в собі питання та варіанти відповідей з певними припущеннями психічних порушень. Деякі групи питань також розбиті на групи питань. Документ має таку структуру:

\_id (objectId) – унікальний ідентифікатор документу;

AvailableTranslations (string) – наявні переклади;

QuestionGroups (object) – об'єкт, який містить в собі групи питань для визначення певних психічних розладів, всього є 5 груп питань на визначення: моральної травми, посттравматичного стресового розладу (ПТСР), постконтузійного синдрому (ПКС), хронічного травматичного стресового розладу (ХТСП та суїцидального синдрому (СЦ):

id (string) – унікальний ідентифікатор групи питань;

questions (object) – об'єкт що зберігає інформацію про питання:

id (string) – унікальний ідентифікатор питання;

answers (object) - об'єкт що зберігає інформацію про відповідь:

id (string) - унікальний ідентифікатор відповіді;

tags (string) – можливі проблеми, що ховаються за даною відповіддю;

translation (string) - переклади відповіді;

translations (object) – переклади питань.

Колекція «Рекомендації» (advice-lists) містить в собі документи з рекомендаціями щодо певних психічних порушень. Кожен документ представляє собою рекомендації на окремо взяти психічний розлад:

\_id (objectId) – унікальний ідентифікатор документу (психологічного порушення);

Problem (string) – психічне порушення;

AvailableTranslations (string) – наявні переклади;

Translations (object) – об'єкт, що містить в собі поради до певного порушення на різних мовах.

## **3 ПРАКТИЧНА РЕАЛІЗАЦІЯ**

### **3.1 Вибір програмного інструментарію**

#### **3.1.1 Система управління базами даних**

Визначено, що для розробки інформаційного забезпечення системи оцінки психологічного стану доцільно використовувати документоорієнтовий тип бази даних, через її гнучкість, масштабованість, швидкість операцій та легкість інтеграції.

Було розглянуто 3 найпопулярніші документоорієнтовані бази даних MongoDB, CouchDB та RavenDB.

MongoDB є одним з найпопулярніших документоорієнтованих сховищ даних, що надає гнучкість у моделюванні даних, високу продуктивність завдяки ефективній індексації та кешуванню, а також можливість горизонтального масштабування через шардінг. Вона також забезпечує надійність за допомогою реплікаційних наборів, що підвищують доступність даних. Активна спільнота користувачів та докладна документація роблять MongoDB зручним вибором для багатьох проектів. Однак, обмежена підтримка транзакцій і можливі проблеми з консистентністю даних можуть стати недоліками в деяких випадках. [9]

CouchDB акцентує увагу на простоті використання та надійності реплікації. Вона використовує HTTP/REST API для доступу до бази даних, що спрощує інтеграцію з іншими системами. CouchDB зберігає дані у форматі JSON, що спрощує роботу з документами. Недоліком є менш розвинена екосистема та обмежена масштабованість порівняно з MongoDB, особливо в великих розподілених системах.[10]

RavenDB підтримує ACID-транзакції, що робить його надійним вибором для додатків, які потребують високої надійності даних. Інтерфейс та інструменти для розробки в RavenDB зручні у використанні, а також є підтримка гібридного зберігання як у хмарі, так і на локальних серверах. Недоліком RavenDB є вища вартість використання через необхідність ліцензування та менша популярність, що

означає меншу кількість доступних ресурсів у порівнянні з MongoDB та CouchDB.[11]

Однією з ключових переваг MongoDB є її здатність забезпечувати гнучкість та масштабованість, що є критично важливим для нашої системи, яка потребує зберігання структурованих та неструктурованих даних у різних форматах. Висока продуктивність MongoDB дозволяє швидко обробляти запити та записувати дані, що є необхідним для системи, яка повинна працювати в режимі реального часу. Реплікація та надійність MongoDB забезпечують безперебійний доступ до даних навіть у разі відмови одного з серверів.

Важливим аспектом є можливість хмарного хостингу. MongoDB Atlas, хмарне рішення для MongoDB, підтримує всі три основні хмарні провайдери (AWS, Azure та GCP), включаючи мультихмарне налаштування, що дозволяє розміщувати дані в різних регіонах або центрах обробки даних у різних провайдерів. Це забезпечує високу доступність та стійкість до збоїв. MongoDB Atlas надає керовані послуги, що звільняє від необхідності самостійно налаштовувати та підтримувати інфраструктуру бази даних, дозволяючи зосередитися на розробці та вдосконаленні самої системи.[12]

Порівняння основних особливостей баз даних проедставлено на табл. 3.1.

Таблиця 3.1. Порівняння основних особливостей баз даних

Особливість	MongoDB	CouchDB	RavenDB
Модель даних	Документоорієнтована (JSON)	Документоорієнтована (JSON)	Документоорієнтована (JSON)
Масштабованість	Висока (шардінг)	Помірна	Висока
Транзакції	Обмежена підтримка транзакцій	Обмежена підтримка транзакцій	Повна підтримка ACID-транзакцій

## Продовження таблиці 3.1. Порівняння основних особливостей баз даних

Реплікація	Реплікаційні набори для високої доступності	Надійна реплікація	Підтримка реплікації та високої доступності
Інтерфейс доступу	MongoDB API, драйвери для багатьох мов	HTTP/REST API	HTTP API, драйвери для .NET
Схема	Гнучка, схему не вимагає	Гнучка, схему не вимагає	Гнучка, схему не вимагає
Хмарний хостинг	MongoDB Atlas (AWS, Azure, GCP)	IBM Cloudant	Підтримка гібридного хостингу
Продуктивність	Висока	Помірна	Висока
Підтримка ком'юніті	Широка, активна спільнота	Менша, але стабільна спільнота	Менша, але зростаюча спільнота
Керовані послуги	Atlas (повний спектр керованих послуг)	Cloudant (керовані послуги)	Ліцензовані рішення для керування
Документація та ресурси	Вичерпна документація та велика кількість ресурсів	Документація та підтримка від IBM	Гарна документація, але менше ресурсів

Таким чином, MongoDB є найкращим вибором для розроблюваного проекту через свою гнучкість у моделюванні даних, високу продуктивність і можливість горизонтального масштабування. Вона забезпечує надійність і високу доступність даних завдяки реплікаційним наборам. MongoDB Atlas, хмарне рішення для MongoDB, надає повний спектр керованих послуг, що дозволяє зосередитися на розробці системи, звільняючи від необхідності самостійно налаштовувати та

підтримувати інфраструктуру бази даних. MongoDB Atlas підтримує всі основні хмарні провайдери (AWS, Azure, GCP), що забезпечує високу доступність та стійкість до збоїв, а також дозволяє легко масштабувати систему за потреби.

### 3.1.2 Вибір мови програмування

Для проєктування інформаційного забезпечення системи оцінки психологічного стану також необхідно обрати мову програмування, яка є найбільш підходящою для розробки інформаційного забезпечення. Всього було розглянуто 3 найпопулярніші мови програмування C#, C++ та Python.

C# - це сучасна, об'єктно-орієнтована мова програмування, яка використовується для розробки різноманітних додатків на платформі .NET.

Переваги:

1. C# розроблений для роботи з платформою .NET, що забезпечує велику бібліотеку класів, інструментів і компонентів для розробки додатків.
2. C# підтримує об'єктно-орієнтоване програмування (ООП), асинхронне програмування, а також нові функції, такі як синтаксичний цукор для запитів LINQ.
3. Система типів C# забезпечує строгий контроль типів під час компіляції, що знижує кількість помилок.
4. Завдяки JIT-компіляції (Just-In-Time) код C# може бути дуже продуктивним.
5. Visual Studio забезпечує потужні можливості для розробки, налагодження та тестування.
6. .NET Core (тепер .NET 5+) дозволяє запускати додатки на різних платформах, включаючи Windows, macOS та Linux.

Недоліки:

1. Для новачків C# може бути складним через свою багатофункціональність і великий синтаксис.
2. Потрібно встановлювати .NET для запуску додатків C#.[13]

C++ - це потужна мова програмування, яка поєднує в собі низькорівневий доступ до пам'яті з високим рівнем продуктивності. Вона використовується для розробки системного програмного забезпечення, ігор, драйверів та інших додатків, де важлива швидкодія та ефективне управління ресурсами. [14]

Переваги:

1. C++ дозволяє розробляти дуже продуктивні програми з низьким рівнем доступу до апаратного забезпечення.
2. Програміст має великий контроль над пам'яттю і ресурсами, що дозволяє оптимізувати програми.
3. C++ широко використовується в індустрії, особливо для системного програмування, розробки ігор, графіки та інших областей, що вимагають високої продуктивності.

1. C++ є дуже складною мовою, особливо для новачків, через її складний синтаксис та концепції управління пам'яттю.
2. Відсутність автоматичного управління пам'яттю може призвести до витоків пам'яті та інших помилок.
3. Через низькорівневі особливості розробка на C++ може займати більше часу.

Python - це високоуровнева мова програмування, відома своєю простотою та зручністю використання. Вона широко використовується у веб-розробці, науці про дані, машинному навчанні та автоматизації. [15]

Переваги:

1. Python відомий своєю простотою і зрозумілим синтаксисом, що робить його легким для вивчення та використання.
2. Python має величезну кількість бібліотек і фреймворків для різних завдань, включаючи аналіз даних, машинне навчання та веб-розробку.
3. Python працює на більшості сучасних операційних систем.

Недоліки:

1. Інтерпретований характер Python робить його повільнішим у порівнянні з компільованими мовами, такими як C++ або C#.

2. Через GIL (Global Interpreter Lock), багатонитковість у Python є обмеженою, що може впливати на продуктивність у багатониткових середовищах.

3. Хоча Python чудово підходить для швидкої розробки прототипів, його використання для великих і складних додатків може бути менш ефективним.[13]

Для розробки інформаційного забезпечення системи оцінки психологічного стану обрано C# через кілька ключових причин. По-перше, C# разом із платформою .NET забезпечує високу продуктивність і масштабованість, що критично важливо для розробки комплексних систем. По-друге, C# надає вбудовані механізми для управління пам'яттю і типізування даних, що підвищує безпеку розроблюваної системи. По-третє, платформа .NET пропонує широкий спектр бібліотек та інструментів, що спрощують розробку і підтримку програмного забезпечення. Нарешті, завдяки потужним інструментам розробки, таким як Visual Studio, процес розробки на C# стає більш зручним та ефективним, скорочуючи час та витрати на створення програмного забезпечення.

### **3.1.3 Вибір програмного середовища**

При розробці інформаційного забезпечення для автоматизованої системи оцінки психологічного стану людини, вибір середовища програмного забезпечення є критично важливим для забезпечення ефективності та продуктивності розробки. Серед найпопулярніших середовищ для розробки коду на C# виділяються Microsoft Visual Studio та JetBrains Rider. Обидва ці IDE мають свої переваги і можуть бути корисними у різних контекстах, але для цього проєкту найбільш доцільним вибором є Microsoft Visual Studio.

Microsoft Visual Studio є потужним і всеосяжним інтегрованим середовищем розробки (IDE), яке пропонує широкий спектр інструментів для створення, налагодження та розгортання додатків на C#. Visual Studio забезпечує інтеграцію з платформою Azure, що може бути корисним для хмарного зберігання та обробки даних, характерних для систем оцінки психологічного стану. Крім того, Visual Studio підтримує роботу з багатьма іншими мовами програмування і платформами,

що робить його універсальним інструментом для розробників, які можуть залучати різні технології для комплексного вирішення задач.[15]

JetBrains Rider також є потужним середовищем розробки, яке поєднує можливості IntelliJ IDEA та ReSharper, надаючи потужні функції аналізу коду, налагодження та рефакторингу. Він також підтримує кросплатформену роботу на Windows, macOS та Linux, що робить його привабливим для розробників, які працюють у різних середовищах. Однак, Rider не має такої тісної інтеграції з Microsoft екосистемою, яка є важливою для C# розробки, особливо в контексті використання специфічних інструментів та сервісів від Microsoft.

Обираючи між цими двома середовищами, було вирішено обрати Microsoft Visual Studio з кількох причин. По-перше, Visual Studio надає більш інтегроване та зручне середовище для розробки з широким спектром функцій, які полегшують процес створення складних додатків. По-друге, тісна інтеграція з Azure дозволяє ефективно використовувати хмарні сервіси для зберігання та обробки даних, що є критичним для нашої системи оцінки психологічного стану. По-третє, Visual Studio має багаторічну історію і велику спільноту користувачів, що забезпечує наявність великої кількості ресурсів, плагінів та підтримки.

Таким чином, вибір на користь Microsoft Visual Studio для розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини є обґрунтованим і зваженим рішенням, що забезпечує високу продуктивність, зручність та інтеграцію з необхідними інструментами і сервісами.

## **3.2 Розробка бази даних**

Процес створення бази даних є ключовим етапом розробки інформаційного забезпечення для автоматизованої системи оцінки психологічного стану користувачів. Вибір правильної бази даних, налаштування її інфраструктури та організація даних всередині системи мають вирішальне значення для забезпечення ефективної, надійної та безпечної роботи системи. Для даного проєкту було обрано документоорієнтовану базу даних MongoDB, яка забезпечує гнучкість у зберіганні

та обробці великих обсягів даних. Використання MongoDB Atlas дозволяє скористатися хмарними можливостями для легкого масштабування та надійного зберігання даних.

Першим кроком у створенні бази даних є налаштування кластера. Кластер - це сукупність обчислювальних ресурсів, що розміщують базу даних і дозволяють їй функціонувати в хмарному середовищі.[16] Для цього ми обрали безкоштовний план MongoDB Atlas, який надає достатньо ресурсів для початкових етапів розробки та тестування. Обрано провайдера AWS (Amazon Web Services) для розміщення нашого кластера, оскільки він пропонує високу надійність та безпеку.

Для створення кластера необхідно увійти до облікового запису MongoDB Atlas та перейти до розділу створення нового кластера. У меню вибору провайдерів обираємо AWS та встановлюємо необхідні параметри для кластера. Після цього створюємо нового користувача, який матиме доступ до бази даних. В процесі створення користувача вказуємо його ім'я, пароль та права доступу.

Для того, щоб завантажити початкові дані в нашу базу даних, ми створимо кілька JSON-файлів. JSON, або JavaScript Object Notation, - це зрозумілий для людини формат обміну даними, визначений на початку 2000-х років. Хоча JSON базується на підмножині стандарту мови програмування JavaScript, він повністю незалежний від мови. Об'єкти JSON є асоціативними контейнерами, в яких ключ-рядок зіставляється зі значенням (яке може бути числом, рядком, логічним числом, масивом, порожнім значенням - нулем або навіть іншим об'єктом). Практично будь-яка мова програмування має реалізацію цієї абстрактної структури даних - об'єкти в JavaScript, словники в Python, хеш-таблиці в Java і C#, асоціативні масиви в C++ і так далі. [17]

Необхідно розробити 2 моделі JSON файлів для колекції тестів, та поради. Для розробки коду використовується Visual Studio.

JSON-файл AdviceList буде містити дані про поради. Кожен документ у цьому файлі містить інформацію про окрему проблему та відповідні поради, а також доступні переклади. JSON код з порадами представлено на рис. 3.1.

```

{
  "id": "658c5b3183e2fbb93fd6a006",
  "problem": "SS",
  "availableTranslations": [
    "uk",
    "en",
    "ru"
  ],
  "translations": {
    "uk": {
      "advices": [
        "Терміново зверніться за допомогою професіоналів у сфері психічного здоров'я, таких як психіатр, психотерапевт або психолог.",
        "Якщо у Вас або у когось іншого виникли думки про самогубство чи негайна загроза життю, негайно зверніться до фахівця.",
        "Залишайтеся на зв'язку, не залишайтеся наодинці.",
        "Усуньте усі засоби, які можуть бути використані для самогубства, такі як ліки або предмети для саморізання.",
        "Зверніться до родини та близьких друзів, які можуть надати підтримку та допомогу в складний момент.",
        "Зверніться до служби кризової допомоги, які надають невідкладну допомогу та консультації. Номери гарячої лінії.",
        "Ця інформація не замінює консультацію з професійним фахівцем. Терміново зверніться до медичних фахівців.",
        "Запам'ятайте, що звернення за допомогою – це прояв сили, а не слабкості. Пам'ятайте, що існують фахівці, які допоможуть."
      ]
    },
    "en": {
      "advices": [
        "Immediately seek the help of mental health professionals, such as a psychiatrist, psychotherapist or psychologist.",
        "If you or someone else is having suicidal thoughts or is in immediate danger of death, seek help immediately.",
        "Stay in touch, do not be alone.",
        "Eliminate all means that could be used to commit suicide, such as medications or self-harm objects.",
        "Reach out to family and close friends who can provide support and help in times of need.",
        "Contact a crisis service that provides emergency assistance and counseling. Crisis hotline numbers are available.",
        "This information is not a substitute for consulting a professional. Please contact medical professionals for help.",
        "Remember that seeking help is a sign of strength, not weakness. Remember that there are professionals who can help."
      ]
    },
    "ru": {
      "advices": [
        "Срочно обратитесь за помощью профессионалов в сфере психического здоровья, таких как психиатр, психотерапевт или психолог.",
        "Если у Вас или у кого-то другого возникли мысли о самоубийстве или немедленная угроза жизни, немедленно обратитесь к специалисту.",
        "Оставайтесь на связи, не оставайтесь наедине.",
        "Устраните все средства, которые могут быть использованы для самоубийства, такие как лекарства или предметы для самоповреждения.",
        "Обратитесь к семье и близким друзьям, которые могут оказать поддержку и помощь в сложный момент.",
        "Свяжитесь со службой кризисной помощи, которая предоставляет неотложную помощь и консультации. Горячие линии.",
        "Эта информация не заменяет консультацию с профессионалом. Срочно обратитесь к медицинским специалистам.",
        "Помните, что обращение за помощью – это проявление силы, а не слабости. Помните, что существуют специалисты, которые помогут."
      ]
    }
  }
}

```

Рис. 3.1. JSON-файл «GeneralTest»

JSON-файл «GeneralTest» буде містити дані про тести. Документ у цьому файлі містить інформацію про тест, включаючи унікальний ідентифікатор, версію тесту, доступні переклади та групи питань, які входять до тесту. Всього є 6 головних груп питань, на визначення певних психічних розладів, а саме: моральної травми, посттравматичного стресового розладу, постконтузійного синдрому, хронічного травматичного стресового розладу та суїцидального синдрому. Деякі з цих груп питань, також розбиті на групи питань на визначення основних симптомів розладу. Кожна група питань містить окремі питання, а кожне питання – відповіді до нього. Всього тест має 133 питання. Фрагмент JSON коду з головним тестом представлено на рис. 3.1.

```

    }
  }
},
"translations": {
  "uk": {
    "title": "Навіть знаходячись у безпеці, Вас супроводжує відчуття ймовірної небезпеки у майбутньому?"
  },
  "en": {
    "title": "Even when you are safe, do you have a sense of potential danger in the future?"
  },
  "ru": {
    "title": "Даже когда вы в безопасности, есть ли у вас ощущение потенциальной опасности в будущем?"
  }
},
{
  "id": "8adfb0ac-df61-4224-bd25-c9a0f5335400",
  "answers": [
    {
      "id": "fca6fd86-88a0-4884-9362-51337937f1d8",
      "tags": [
        "PTSD",
        "MT",
        "PCS"
      ],
      "translations": {
        "uk": {
          "title": "Завжди"
        },
        "en": {
          "title": "Always"
        },
        "ru": {
          "title": "Всегда"
        }
      }
    }
  ],
  {
    "id": "e99bfacc-424b-483c-8cca-47a2741f22a8",
    "tags": [

```

Рис. 3.2. JSON-файл «GeneralTest»

Наступним етапом є процес створення нової бази даних «AppDB» у MongoDB Atlas і додавання колекцій «advice-lists», «tests» та «applicationUsers». Для колекції «applicationUsers» JSON-файл не створюється, тому що бекенд-частина автоматично заповнює колекцію «applicationUsers» даними під час реєстрації користувачів.

Для завантаження даних у колекції потрібно забезпечити наявність підготовлених JSON-файлів. Завантаження нових даних виконувалось за допомогою веб-інтерфейсу MongoDB Atlas. Також можна виконувати додавання нових даних через командний рядок, що також буде описано.

Створена база даних в MongoDB Atlas з вище описаними колекціями представлена на рис. 3.3.

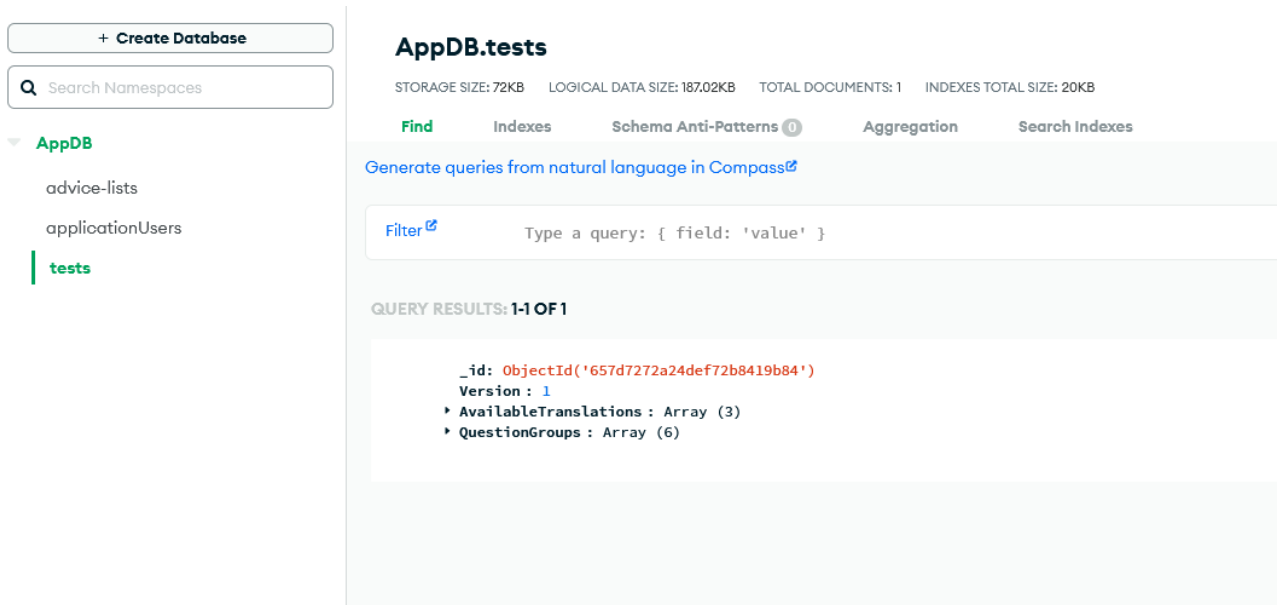


Рис. 3.3. Створена база даних

### 3.3 Інструменти взаємодії з базою даних

Для взаємодії з базою даних MongoDB існує кілька інструментів, серед яких MongoDB Compass займає важливе місце. MongoDB Compass — це графічний інтерфейс користувача (GUI), який дозволяє легко працювати з базами даних MongoDB. Він забезпечує можливість візуалізації структури даних, виконання запитів, індексування і багато іншого без необхідності писати код. Це робить його особливо корисним для тих, хто віддає перевагу візуальному підходу до управління базами даних.[18]

Однією з основних функцій MongoDB Compass є вбудований MongoDB Shell. MongoDB Shell — це інтерактивний інтерфейс командного рядка, що дозволяє виконувати команди для роботи з базою даних. Використовуючи Shell, користувачі можуть виконувати ті самі операції, що і в Compass, але через командний рядок. Це включає створення, читання, оновлення та видалення документів (CRUD операції), а також виконання складних запитів та операцій агрегації.

Для підключення до MongoDB Compass кластера, розгорнутого на MongoDB Atlas, потрібно в MongoDB Compass вставити URI у поле підключення, ввівши ім'я користувача та пароль.

Важливо розуміти різницю між використанням MongoDB Compass і MongoDB Atlas. MongoDB Atlas — це хмарна платформа, яка надає можливість розгортання, управління та масштабування кластерів MongoDB в хмарі. Compass, у свою чергу, є інструментом для локального або віддаленого підключення до бази даних, включаючи бази даних, розгорнуті на Atlas. Основна різниця полягає в тому, що Atlas управляє інфраструктурою, тоді як Compass надає зручний інтерфейс для роботи з даними. Інтерфейс MongoDB Compass представлений на рис. 3.4.

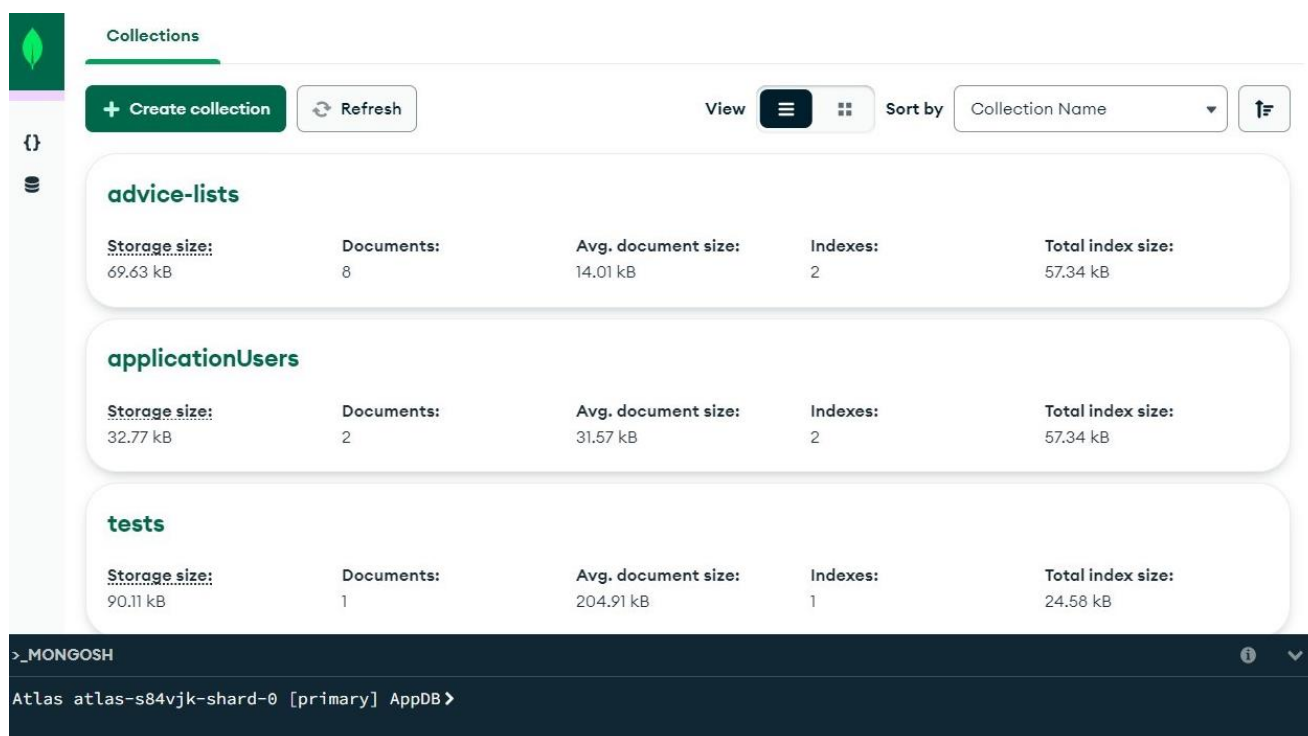


Рис. 3.4. Інтерфейс MongoDB Compass

У MongoDB Compass користувачі можуть виконувати різні команди для взаємодії з базою даних. Основні команди для взаємодії з базою даних включають `show collections` для перегляду колекцій, `insertOne` для створення нового документа, `updateOne` для оновлення документа, `find` для пошуку по полю та `deleteOne` для видалення документа. Робота з базою даних в MongoDB Compass представлена на рис. 3.5.

```

> show collections
< advice-lists
  applicationUsers
  tests
> db["advice-lists"].insertOne({ problem: "PP", "availableTranslations": [
  "uk"
  ],
  "translations": {
    "uk": {
      "advices": [
        "Зверніться до родини та близьких друзів, які можуть надати підтримку та допомогу в складний момент.",
        "Зверніться до служби кризової допомоги, які надають невідкладну допомогу та консультації. Номери гарячих ліній
        "Ця інформація не заміняє консультацію з професійним фахівцем. Терміново зверніться до медичних фахівців або с
        "Запам'ятайте, що звернення за допомогою – це прояв сили, а не слабкості. Пам'ятайте, що існують фахівці, які
      ]
    }
  }
})
< {
> db["advice-lists"].find({ problem: "PP" })
< {
  _id: ObjectId("664dec0a0352d3e9f65c5707"),
  problem: 'PP',
  availableTranslations: [
    'uk'
  ],
  translations: {
    uk: {
      advices: [
        'Зверніться до родини та близьких друзів, які можуть надати підтримку та допомогу в складний момент.',
        'Зверніться до служби кризової допомоги, які надають невідкладну допомогу та консультації. Номери гарячих ліній
        'Ця інформація не заміняє консультацію з професійним фахівцем. Терміново зверніться до медичних фахівців або слу
        'Запам'ятайте, що звернення за допомогою – це прояв сили, а не слабкості. Пам'ятайте, що існують фахівці, які го
      ]
    }
  }
}
> db["advice-lists"].updateOne(
  { title: "PP" },
  { $set: { translations: {
    uk: {
      advices: [
        "Запам'ятайте, що звернення за допомогою – це прояв сили, а не слабкості. Пам'ятайте, що існують фахівці, які го
      ]
    }
  } }
})
> db["advice-lists"].deleteOne({ problem: "PP" })

```

Рис. 3.5. Робота з базою даних в MongoDB Compass

Таким чином, MongoDB Compass разом з вбудованим MongoDB Shell надають потужні інструменти для роботи з базами даних MongoDB. Використання Compass з його графічним інтерфейсом робить взаємодію з даними більш інтуїтивною, тоді як Shell дозволяє виконувати точні та складні операції через командний рядок. Обидва інструменти доповнюють один одного і забезпечують ефективну роботу з базами даних як локально, так і в хмарному середовищі, такому як MongoDB Atlas.

### 3.4 Створення індексів для пришвидшення запитів

Індекси - це спеціальні структури даних, які зберігають невелику частину набору даних колекції у зручному для перегляду вигляді. Індекси MongoDB використовують B-дерево структуру даних. Індекс зберігає значення певного поля або набору полів, впорядкованих за значенням поля. Впорядкування записів індексу підтримує ефективні операції рівності та запити на основі діапазону. Без індексів, MongoDB повинна сканувати кожен документ у колекції, щоб повернути результати запиту. Якщо для запиту існує відповідний індекс, MongoDB використовує індекс, щоб обмежити кількість документів, які потрібно просканувати. [19]

Серед різних типів індексів, які надає MongoDB, можна виділити наступні:

1. Часткові індекси створюються на підмножині документів у колекції, що відповідають певному фільтру. Це дозволяє зменшити розмір індексу та підвищити продуктивність. Вони корисні, коли лише підмножина документів потребує індексації, наприклад, активні користувачі або документи з високими значеннями.

2. Рідкі індекси включають лише документи, що містять індексоване поле. Документи без цього поля не включаються в індекс, що дозволяє зменшити розмір індексу та підвищити швидкість запису. Вони корисні для колекцій з документами, де індексоване поле є необов'язковим.

3. Індекси з терміном життя автоматично видаляють документи після заданого періоду часу. Вони використовуються для зберігання тимчасових даних, що дозволяє автоматично управляти старими даними і звільняти простір в базі даних. Це особливо корисно для логів, сесій та інших тимчасових даних.

4. Двовимірні індекси використовуються для геопросторових запитів на двовимірних площинах. Вони зберігають координати точок і дозволяють виконувати геопросторові запити, такі як пошук у радіусі або визначення близьких точок. Вони корисні для додатків, які працюють з геолокацією та картографічними сервісами.

Також при створенні індексу можна вказувати опції, що дозволяють налаштувати різні аспекти його поведінки та властивостей. Ось короткий опис деяких з найбільш поширених опцій:

- background: Дозволяє створювати індекс в фоновому режимі, не блокуючи операції запису в базу даних. Це корисно для великих колекцій, де блокування може вплинути на продуктивність.

- unique: Вказує, чи повинні бути значення в індексі унікальними. Якщо це встановлено в true, то дублікати значень не допускаються в індексі.

- name: Дозволяє вам задати ім'я для індексу. Якщо ви не вказуєте ім'я, MongoDB автоматично створить його.

- sparse: Вказує, чи мають документи без вказаного поля бути включені в індекс. Якщо встановлено в true, тільки документи з вказаним полем будуть включені в індекс.

- expireAfterSeconds: Ця опція визначає час життя документа після його закінчення. Вона корисна для створення TTL (Time-To-Live) індексів, які автоматично видаляють документи після певного часу.

- partialFilterExpression: Ця опція дозволяє вам вказати фільтр, який визначає, які документи повинні бути включені в індекс. Тільки документи, що відповідають умовам фільтра, будуть включені в індекс.

- collation: Ця опція дозволяє вам визначити специфічну культуру сортування для рядкових значень в індексі, що допомагає впорядковувати дані відповідно до потреб вашого додатку.

Ці опції дають можливість налаштувати індекс з урахуванням конкретних вимог системи, що підвищує його ефективність та продуктивність. Для розроблюваної системи було обрано використовувати опцію unique.

Для інформаційного забезпечення системи прийняття рішень та оцінки психологічного стану було обрано створити індекси на ключових полях, що забезпечить ефективний і швидкий доступ до необхідної інформації. Створені індекси на ключових полях у колекціях бази даних вказані на табл. 3.2

Таблиця 3.2. Створені індекси на ключових полях

Колекція	Поле	Призначення
advice-lists	problem	Індекс на цьому полі дозволяє швидко знаходити документи з порадами для конкретної проблеми, що часто запитуватимуть користувачі.
applicationUsers	Email	Індекс на цьому полі дозволяє швидко знаходити користувачів за електронними адресами.
	UserName	Індекс на цьому полі допомагає ефективно здійснювати пошук користувачів за іменами.
	NormalizedEmail	Індекс на цьому полі дозволяє швидко перевіряти унікальність електронних адрес і уникати дублювання.
	GeneralTestResults_id	Індекс на цьому полі забезпечує швидкий доступ до результатів тестування.

Для створення індексів використовується командний рядок в MongoDB Compass. Запити для створення потрібних індексів наведено на рис. 3.4:

```

> db.applicationUsers.createIndex(
  { UserName: 1, Email: 1, NormalizedEmail: 1, "GeneralTestResults.id": 1 }, {
    unique: true, }
)
< UserName_1_Email_1_NormalizedEmail_1_GeneralTestResults.id_1
> db["advice-lists"].createIndex(
  { problem: 1 }, { unique: true }
)
< problem_1

```

Рис. 3.4. Процес створення індексів

В підсумку, індекси є критично важливим інструментом для оптимізації роботи баз даних. Вони значно підвищують продуктивність системи, зменшуючи час виконання запитів та забезпечуючи ефективну роботу з великими обсягами даних. В даному випадку вибір полів для індексування був зроблений на основі аналізу найчастіших запитів, що дозволяє досягти максимальної ефективності та швидкодії.

### 3.5 Налаштування безпеки при взаємодії з базою даних

Налаштування безпеки є важливим аспектом при взаємодії з базою даних, особливо в хмарних сервісах, таких як MongoDB Atlas. У MongoDB Atlas є два основних способи керування безпекою: керування доступом до бази даних (Database Access) та керування доступом до мережі (Network Access). Керування доступом до бази даних дозволяє створювати нових користувачів і налаштовувати їхні права доступу. Це дозволяє точно визначити, які дії можуть виконувати різні користувачі бази даних, що допомагає захистити дані від несанкціонованого доступу.

MongoDB Atlas надає можливість створювати користувачів бази даних з різними рівнями доступу. Користувачі можуть мати права на читання (read-only), права на запис (read/write), або бути адміністраторами з повним доступом до бази даних. Для забезпечення безпеки важливо надати користувачам тільки ті права, які необхідні для виконання їхніх завдань. Для бекенд-системи було створено користувача з обмеженими правами доступу, що дозволяє виконувати тільки ті операції, які потрібні для функціонування системи, а саме читання і запис. Це зображено на рис. 3.5.

Database Users		Custom Roles	
User Name ↕	Authentication Method ↕	MongoDB Roles	Resources
👤 app	SCRAM	readWriteAnyDatabase@admin	All Resources

Рис. 3.5. Користувачі бази даних

Керування доступом до мережі в MongoDB Atlas дозволяє контролювати, які IP-адреси можуть підключатися до бази даних. За замовчуванням, доступ до бази даних дозволений лише з однієї IP-адреси, що забезпечує базовий рівень захисту від несанкціонованих підключень. Адміністратори можуть додавати додаткові IP-адреси або діапазони IP-адрес, з яких дозволено підключення. Було вирішено дозволити доступ до бази даних з будь-якої точки світу, тому було додано IP-адресу 0.0.0.0/0, що дозволить доступ з будь-якого місця.

## Network Access

IP Access List	Peering	Private Endpoint
You will only be able to connect to your cluster from the following list of IP Addresses:		
IP Address	Comment	Status
0.0.0.0/0 (includes your current IP address)		● Active

Рис. 3.6. Дозволені IP-адреси

Впровадження таких налаштувань допомагає забезпечити безпеку даних, що є критично важливим для будь-якої сучасної системи, особливо для тих, які працюють з чутливою інформацією, такою як дані про психічний стан користувачів. Таким чином, налаштування безпеки при взаємодії з базою даних у MongoDB Atlas є важливим кроком у розробці надійної та безпечної системи.

### 3.6 Створення класів документів

Розробка класів документів є важливим етапом в процесі створення інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини. Хоча JSON-файли можуть бути використані для початкового завантаження даних, класи в мові програмування C# надають додаткові можливості

для управління даними, такі як валідація, трансформація, зручність доступу та інтеграція з іншими компонентами системи.

Основна причина розробки цих класів — полягає у створенні чітко структурованої та легко керованої моделі даних, яка дозволяє ефективно працювати з інформацією у програмному середовищі. Використання класів у C# дозволяє забезпечити типізацію даних, що знижує ризик помилок під час роботи з ними, а також забезпечує можливість розширення та модифікації структури даних без суттєвих змін в коді. Класи також надають механізми для управління доступом до даних через властивості `get` і `set`, які дозволяють контролювати процес читання та запису значень.

Клас `ApplicationUser` представляє користувача системи. Він містить особисті дані користувача (`UserInfo`), аватар, токени для оновлення сесії, та результати загальних тестів. Це дозволяє зберігати всю необхідну інформацію про користувача в одному документі. Код `ApplicationUser` наведений в додатку А.

Клас `GeneralTest` представляє загальний тест, який складається з груп питань. Кожна група питань (`QuestionGroup`) може містити інші групи питань або окремі питання. Кожне питання (`Question`) може мати кілька варіантів відповідей (`Answer`). Така структура дозволяє гнучко налаштовувати тести і їхні питання. Код `GeneralTest` наведений в додатку Б.

Клас `AdviceList` представляє список порад для вирішення певної проблеми. Він містить ідентифікатор, версію, опис проблеми, доступні переклади, і самі поради. Це дозволяє зберігати всі необхідні дані для надання користувачам релевантних рекомендацій. Код `AdviceList` наведений в додатку В.

Було розроблено класи документів, які забезпечують зручність і ефективність роботи з інформацією в програмному середовищі. Написані класи ``ApplicationUser``, ``GeneralTest`` та ``AdviceList`` організовують зберігання, обробку та передачу даних, відповідаючи потребам системи оцінки психологічного стану користувачів. Властивості ``get`` і ``set`` у C# забезпечують інкапсуляцію даних і контроль доступу до них, покращуючи безпеку та надійність системи.

На рис. 3.7 описано UML діаграму класів документів. Класи включають `UserInfo`, `ApplicationUser`, `GeneralTest`, `QuestionGroup`, `Question`, `Answer` та інші, де кожен клас має певні атрибути. Взаємозв'язки між класами показують, що, наприклад, `ApplicationUser` містить `UserInfo` і має багато `GeneralTestResults`, кожен `GeneralTestResult` включає відповіді на питання (`AnsweredQuestion`), а самі тести (`GeneralTest`) складаються з груп питань (`QuestionGroup`), які, в свою чергу, містять питання (`Question`) та відповіді на них (`Answer`). Клас `GeneralTest` має композиційний зв'язок із класом `QuestionGroup`, який також має рекурсивний зв'язок із власним класом, включаючи інші `QuestionGroups`. Клас `Question` містить відповіді (`Answer`) та переклади (`QuestionTranslation`), а клас `Answer` містить переклади (`AnswerTranslation`). Аналогічно, клас `AdviceList` містить переклади (`AdviceListTranslation`) через словник `Translations`.

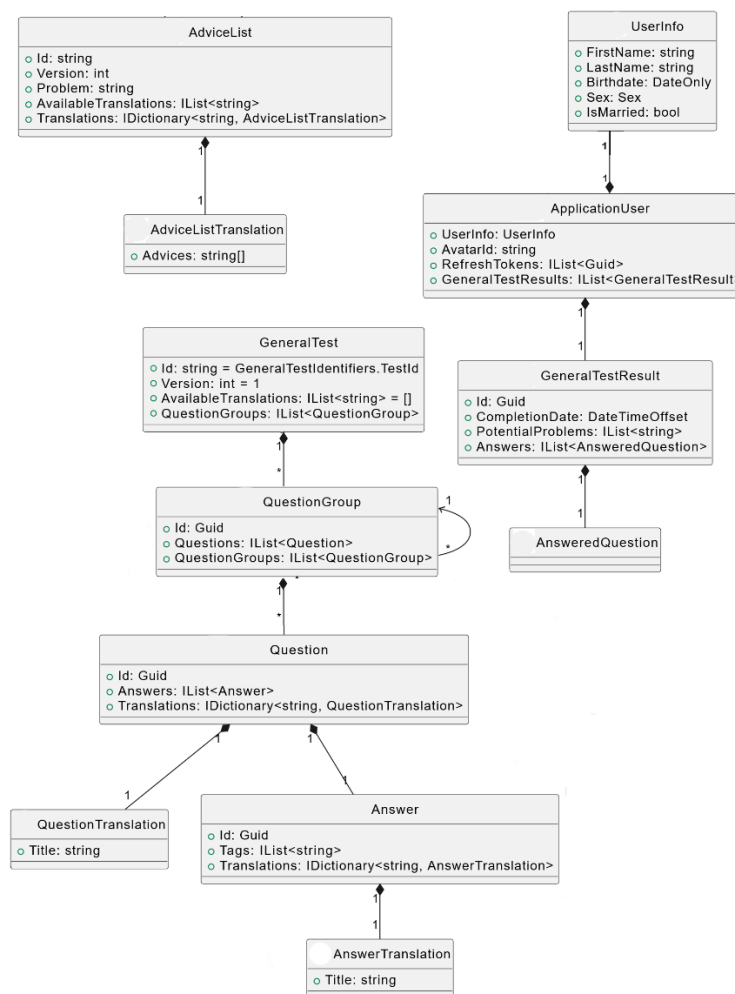


Рис. 3.7. UML-діаграма документів класів

### 3.6 Розробка репозиторіїв для взаємодії з даними

Розробка репозиторіїв для взаємодії з даними є важливим аспектом розробки інформаційного забезпечення. Такі системи вимагають ефективного і надійного зберігання та обробки даних, щоб забезпечити точність і своєчасність аналізу інформації, необхідної для прийняття рішень.

Репозиторій є шаблоном проектування, який абстрагує доступ до даних. Це означає, що він забезпечує інтерфейс для роботи з даними, незалежно від того, де і як вони зберігаються – в базах даних, файлах чи через веб-сервіси. Репозиторій дозволяє ізолювати бізнес-логіку додатку від деталей доступу до даних, що підвищує модульність і підтримуваність коду.

Інтерфейс репозиторію визначає набір методів, які повинні бути реалізовані для роботи з певними типами даних. Це дозволяє забезпечити узгоджений і передбачуваний спосіб доступу до даних, що важливо для підтримки складних систем, таких як системи прийняття рішень та оцінки психологічного стану.

Розробка репозиторіїв для даного проекту включала розробку інтерфейсів `IAdviceListRepository`, `ITestRepository`, `IUserRepository`.

Інтерфейс `IAdviceListRepository` визначає методи для роботи зі списками порад. Наприклад:

- `GetAdviceLists`: повертає колекцію об'єктів `AdviceList` для заданого коду мови, що важливо для підтримки багатомовних додатків.

- `CheckAdviceListsExistence`: перевіряє, чи існують списки порад у системі.

- `SaveAdviceListsAsync`: зберігає колекцію списків порад у сховище даних.

Цей інтерфейс забезпечує всі необхідні операції для керування списками порад, які можуть використовуватися у процесі прийняття рішень та оцінки психологічного стану пацієнтів.

Інтерфейс `ITestRepository` визначає методи для роботи з тестами. Наприклад:

- `CheckGeneralTestExistence`: перевіряє, чи існує загальний тест у системі.

- ``GetGeneralTest``: повертає об'єкт ``GeneralTest``, який представляє загальний тест.

- ``SaveGeneralTest``: зберігає загальний тест у сховище даних.

- ``GetGeneralTestQuestions``: повертає колекцію запитань для тесту на основі заданого коду мови.

Цей інтерфейс дозволяє здійснювати основні операції з тестами, які є ключовими для оцінки психологічного стану пацієнтів.

Інтерфейс ``IUserRepository`` визначає методи для роботи з користувачами та їх результатами тестів. Наприклад:

- ``AddRefreshToken``: додає новий токен оновлення для користувача, що важливо для підтримки безпеки та управління сесіями.

- ``RemoveRefreshToken``: видаляє токен оновлення.

- ``CheckTokenExistence``: перевіряє наявність токена для користувача.

- ``SaveGeneralTestResult``: зберігає результат тесту для користувача.

- ``GetGeneralTestResults``: повертає всі результати тестів для користувача.

- ``GetGeneralTestResult``: повертає конкретний результат тесту для користувача.

Цей інтерфейс забезпечує управління аутентифікацією та зберіганням результатів тестів, що є критично важливим для систем оцінки психологічного стану.

Репозиторії дозволяють створити чітку структуру для взаємодії з даними. Вони забезпечують модульність, що полегшує тестування і підтримку коду.

У системах прийняття рішень та оцінки психологічного стану репозиторії грають ключову роль, оскільки вони забезпечують ефективний і надійний доступ до критично важливих даних. Вони дозволяють швидко отримувати необхідну інформацію для аналізу та прийняття рішень, зберігати результати тестувань і забезпечувати безпеку даних користувачів.

Таким чином, розробка репозиторіїв є важливою складовою інформаційного забезпечення систем прийняття рішень та оцінки психологічного стану, забезпечуючи структурованість, модульність і ефективність роботи з даними.

### 3.8 Ініціалізація бази даних

Ініціалізація бази даних є важливим процесом, що забезпечує підготовку середовища для зберігання і обробки даних у системі оцінки психологічного стану. Щоб уникнути необхідності вручного введення та заповнення бази даних, можна розробити спеціальну процедуру ініціалізації бази даних. Це включає створення автоматизованих скриптів для налаштування структури бази даних, таких як створення таблиць, визначення їх полів і типів даних, а також автоматичне наповнення бази даних початковими даними.

Під час ініціалізації створюється структура даних, яка включає всі необхідні таблиці, індекси та зв'язки між ними. Це фундамент для подальшого зберігання і управління даними. Наприклад, у системі, яка займається оцінкою психологічного стану, можуть бути створені таблиці для зберігання інформації про користувачів, їхні результати тестувань, різні тести, а також списки порад. Кожна з цих таблиць повинна бути налаштована відповідно до вимог цілісності даних, що включає налаштування зовнішніх ключів, обмежень та індексів для забезпечення ефективного доступу до даних.

Завантаження початкових даних – це ще один ключовий аспект ініціалізації бази даних. Початкові дані можуть включати базові конфігураційні налаштування, попередньо зібрані дані для тестування або демонстраційні дані, які потрібні для початкового запуску додатку. У системі оцінки психологічного стану це можуть бути початкові тести та списки порад, які будуть використовуватися для аналізу та рекомендацій. Ці дані часто зберігаються у вигляді JSON-файлів або інших форматів, які можуть бути легко імпортовані в базу даних під час ініціалізації.

Код для ініціалізації бази даних часто включає скрипти або класи, які виконують перевірку наявності необхідних даних і завантажують їх у разі відсутності. У даному випадку, в класі `DatabaseInitializer`, реалізованому у файлі `DatabaseInitializer.cs`, використовується два репозиторії – `ITestRepository` та `IAdviceListRepository` – для роботи з тестами і списками порад відповідно. Спочатку перевіряється наявність загального тесту у базі даних. Якщо тест

відсутній, він завантажується з файлу, десеріалізується і зберігається у базу даних. Аналогічний процес відбувається для списків порад: якщо вони відсутні, дані завантажуються з файлу, десеріалізуються і зберігаються у базу даних. Використання механізмів серіалізації та десеріалізації дозволяє легко працювати з даними у форматі JSON, що є зручним для зберігання і передачі структурованої інформації.

Процес ініціалізації також включає налаштування параметрів конфігурації, таких як шляхи до файлів з початковими даними, що зберігаються в конфігураційних файлах або передаються через опції. Це дозволяє зробити процес ініціалізації гнучким і адаптивним до змін, що можуть виникнути у майбутньому. Наприклад, зміна шляху до файлу з тестами або списками порад не потребує змін у коді класу `DatabaseInitializer`, достатньо змінити відповідну опцію в конфігураційному файлі. Код процесу ініціалізації даних наведено в додатку Г.

Отже, ініціалізація бази даних є критично важливим етапом у розробці системи оцінки психологічного стану. Вона забезпечує створення необхідної структури бази даних, завантаження початкових даних та налаштування необхідних параметрів для коректного функціонування додатку. Використання відповідних інтерфейсів і класів для ініціалізації дозволяє зробити цей процес гнучким, модульним і легким у підтримці, що є важливим для ефективної роботи всієї системи.

## 4. БІЗНЕС ПЛАН

### 4.1 Впровадження продукту

Для впровадження продукту дипломної роботи "Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини" було вирішено створити такий теоретичний план:

1. Аналіз потреб і можливостей. У процесі роботи було проведено всебічний аналіз ринку систем для оцінки психологічного стану людини. Було вивчено конкурентні переваги та недоліки існуючих продуктів, визначено потенційних користувачів і їхні потреби, а також передбачено можливі відгуки про власний продукт. Це дало можливість сформулювати чітке уявлення про те, які функціональні можливості необхідні для успішного впровадження системи.

2. Розробка продукту. Основною метою стало створення інформаційного забезпечення для автоматизованої системи оцінки психологічного стану людини. Було розроблено базу даних для зберігання результатів тестів, даних користувачів та рекомендацій в залежності від їх проблеми, а також інші необхідні компоненти.

3. Тестування. Після завершення розробки було проведено ретельне тестування системи на різних етапах, щоб виявити та виправити можливі помилки та проблеми.

4. Маркетингові дослідження. Проведено маркетингові дослідження, щоб вивчити потреби та вимоги потенційних користувачів. Це дозволило визначити, які додаткові функції та можливості можуть зробити продукт більш привабливим для ринку. Було також розглянуто конкурентні продукти та їхні особливості.

5. Випробування в реальних умовах. Після успішного завершення тестування та маркетингових досліджень система була теоретично протестована в реальних умовах. Це включало перевірку роботи продукту в різних умовах та з різними даними для підтвердження його працездатності та сумісності з існуючими технологіями.

6. Впровадження. Після завершення всіх випробувань було розроблено бізнес-план для впровадження продукту на ринку. Використано різні стратегії

продажу в теорії, включаючи онлайн-продажі та продажі через торгові мережі та укладання угоди з партнерами та постачальниками для забезпечення надійного постачання та підтримки продукту

7. Підтримка та поновлення. Впровадження продукту створило необхідність у постійній технічній підтримці користувачів, виправленні помилок, додаванні нових функцій та регулярному оновленні програмного забезпечення. Це дозволяє забезпечити безперервну та ефективну роботу системи.

8. Моніторинг та аналіз. Після інтеграції продукту в реальне використання, проводиться постійний моніторинг та аналіз його роботи. Це дозволяє виявляти можливі проблеми та недоліки, а також вдосконалювати систему для кращого задоволення потреб користувачів.

9. Розвиток продукту. Останнім етапом впровадження є постійний розвиток продукту, щоб він відповідав новим технологічним вимогам та потребам користувачів. Це включає випуск нових версій програмного забезпечення, додавання нових функцій та розширення можливостей системи для різних категорій користувачів.

Такий підхід до впровадження інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини забезпечить її конкурентоспроможність, функціональність та зручність у використанні.

#### **4.1.1 Визначні риси по відношенню до продукту конкурентів**

Проект "Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини" потребує різноманітних підходів до залучення фінансування та реалізації. Один із способів отримання фінансової підтримки включає залучення інвестицій від неформальних інвесторів, венчурних фондів або корпорацій, що надасть необхідні ресурси для розробки та запуску продукту. Альтернативно, можна розглядати отримання банківського кредиту або участь у грантових програмах та державних ініціативах для підтримки стартапів, що забезпечить фінансову стійкість проекту.

Впровадження продукту включає кілька ключових етапів: від початкової концепції та прототипування до остаточного випуску та підтримки. На кожному етапі важливо забезпечувати високу якість розробки, ретельно оцінювати ризики та проводити всебічне тестування продукту. Доцільно залучати спеціалістів з різних сфер, таких як розробка інтерфейсу, управління базами даних, програмна логіка, що дозволяє створити інтегроване та комплексне рішення, яке відповідає потребам кінцевих користувачів.

Після виходу продукту на ринок важливо забезпечити його постійну підтримку та оновлення. Це включає технічну підтримку користувачів, виправлення виявлених помилок, регулярні оновлення та додавання нових функцій, що підвищують конкурентоспроможність продукту. Необхідно також розробити ефективну стратегію монетизації, яка забезпечить прибуток та подальший розвиток продукту.

Розглядається можливість випуску продукту як безкоштовного з підтримкою відкритої спільноти розробників та ентузіастів. Такий підхід може бути вигідним для проектів з відкритим кодом, де спільнота бере активну участь у розробці, вдосконаленні функціональності та підтримці продукту. Спільнота може забезпечувати технічну підтримку користувачів, вирішувати виникаючі проблеми та вносити необхідні зміни для покращення якості та функціональності продукту.

Проте, безкоштовний випуск продукту має свої ризики. Зокрема, може виникнути дефіцит фінансування для розвитку та підтримки, обмеження функціональних можливостей та зниження стабільності продукту. Тому перед прийняттям такого рішення необхідно ретельно зважити всі плюси та мінуси цього підходу.

Продукт вирізняється на фоні конкурентів кількома ключовими аспектами. По-перше, він розроблений з використанням найсучасніших технологій, що забезпечує високу точність оцінки психологічного стану людини. По-друге, база даних створені з акцентом на максимальну зручність та ефективність використання. По-третє, завдяки активному залученню спільноти та постійним

оновленням, продукт може швидко адаптуватися до ринкових змін та потреб користувачів, що гарантує його конкурентоспроможність.

Таким чином, підхід до розробки та впровадження продукту забезпечує його високу якість, функціональність та здатність до адаптації, що є критичними факторами успіху на ринку систем оцінки психологічного стану людини.

#### **4.1.2 Необхідність продукту споживачу**

Розробка інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини має на меті задовольнити низку потреб споживачів, які не повністю задовольняються існуючими продуктами на ринку. Порівняння з конкурентами дозволило виділити декілька ключових визначних рис, які роблять даний продукт привабливим для користувачів.

Однією з найважливіших характеристик продукту є його висока продуктивність та швидкодія. Система забезпечує більш швидкий та точний аналіз даних, що є важливим для користувачів, які потребують оперативних та надійних результатів. Висока продуктивність системи сприяє підвищенню ефективності роботи, що робить її зручнішою у використанні.

Функціональні можливості продукту також значно перевищують існуючі аналоги на ринку. Система пропонує широкий спектр функцій, які дозволяють більш точно та всебічно оцінювати психічний стан користувачів.

Використання MongoDB як основної бази даних забезпечує низку переваг. Ця база даних відзначається високою продуктивністю та гнучкістю, що дозволяє швидко обробляти великі обсяги даних та забезпечувати їхню надійність. Крім того, MongoDB підтримує горизонтальне масштабування, що дозволяє системі рости разом з кількістю користувачів без втрати продуктивності.

Іншою важливою перевагою є використання репозиторіїв для роботи з базою даних. Це дозволяє забезпечити структуровану та зручну роботу з даними, а також спрощує процес підтримки та оновлення системи. Репозиторії забезпечують зручний доступ до даних та спрощують виконання складних запитів.

Процес ініціалізації бази даних також є важливим аспектом системи. Він забезпечує автоматизований процес налаштування та підготовки бази даних до роботи, що знижує ризик помилок та підвищує ефективність роботи з системою. Автоматизована ініціалізація дозволяє швидко та ефективно розпочати роботу з базою даних без необхідності виконувати складні ручні налаштування.

Загалом, визначні риси продукту включають технічні характеристики, такі як продуктивність, гнучкість та масштабованість. Ці особливості дозволяють забезпечити високий рівень задоволеності користувачів та підвищують конкурентоспроможність продукту на ринку систем оцінки психологічного стану людини.

#### **4.1.3. Очікувані основні фінансові результати**

При розробці інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини необхідно врахувати кілька ключових факторів: витрати на розробку програмного забезпечення, інженерні рішення, маркетингові витрати та інші витрати на підтримку бізнесу.

Прогнозні об'єми продажу на найближчі роки залежать від багатьох змінних, таких як ринкова конкуренція, зміна технологічних стандартів та маркетингові стратегії. При оцінці можливих об'ємів продажів враховані ці фактори і встановлено приблизну кількість продукту, що може бути реалізована на ринку. Наприклад, очікується, що протягом першого року буде продано близько 500 ліцензій на програмне забезпечення, протягом другого року – 1500 ліцензій, а протягом третього року – 2500 ліцензій.

Прибуток від продажів можна визначити шляхом множення прогнозованої кількості проданого продукту на його ціну. Якщо, наприклад, ціна однієї ліцензії складає 500 доларів, то прибуток від продажу протягом першого року становитиме 250 тисяч доларів, протягом другого року – 750 тисяч доларів, а протягом третього року – 1,25 мільйона доларів.

Витрати на розробку включають витрати на матеріали, заробітну плату працівників, оренду приміщення та інші витрати, пов'язані з розробкою та підтримкою програмного забезпечення. Наприклад, витрати на розробку програмного забезпечення можуть скласти 200 тисяч доларів на перший рік, включаючи заробітну плату розробників, оренду офісу, витрати на обладнання та інші супутні витрати. На другий і третій роки, враховуючи підтримку та оновлення, витрати можуть скласти по 100 тисяч доларів на рік.

Валовий прибуток можна визначити шляхом віднімання витрат на розробку від прибутку від продажу. Наприклад, якщо вартість розробки та підтримки програмного забезпечення протягом першого року складає 200 тисяч доларів, а прибуток від продажу становить 250 тисяч доларів, то валовий прибуток за перший рік складе 50 тисяч доларів. За другий рік, з урахуванням витрат у 100 тисяч доларів і прибутку в 750 тисяч доларів, валовий прибуток складе 650 тисяч доларів. За третій рік, з урахуванням витрат у 100 тисяч доларів і прибутку в 1,25 мільйона доларів, валовий прибуток складе 1,15 мільйона доларів.

Термін повернення інвестицій можна розрахувати шляхом визначення чистого прибутку, який отримає підприємство протягом кожного року, та порівняння його з розміром початкових інвестицій. Наприклад, якщо початкові інвестиції складають 300 тисяч доларів, то з урахуванням чистого прибутку за перший рік у розмірі 50 тисяч доларів, за другий рік – 650 тисяч доларів, та за третій рік – 1,15 мільйона доларів, загальна сума прибутку за три роки становитиме 1,85 мільйона доларів. Відповідно, термін повернення інвестицій становитиме менше одного року, враховуючи значний прибуток, отриманий за другий рік.

Для успішної реалізації проекту з розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини необхідно враховувати очікувані об'єми продажу на найближчі три роки, прибуток від продажу, витрати на розробку, валовий прибуток, рівень прибутковості та термін повернення інвестицій. Такий аналіз допоможе оцінити фінансові ризики та прийняти обґрунтовані рішення щодо реалізації проекту.

## **4.2 Проектований продукт, або вид послуг**

### **4.2.1 Потреби продукту по відношенню до проєкту**

Продукт проєкту розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану людини повинен задовольняти наступні вимоги:

1. Комплексна оцінка психологічного стану: Забезпечення можливості оцінювати різні аспекти психологічного стану людини.

2. Автоматизація процесів: Оптимізація процесу збору та аналізу даних для надання швидких і точних результатів, що сприятиме своєчасній діагностиці та лікуванню.

3. Віддалений доступ: Забезпечення можливості віддаленого доступу до бази даних.

5. Безпека та конфіденційність: Забезпечення високого рівня безпеки та конфіденційності даних користувачів, зокрема шляхом шифрування та обмеження доступу до даних.

6. Інтерфейс: Використання інтерфейсу хмарної платформи даних для зручного керування базами даних, що дозволяє ефективно зберігати та обробляти великі обсяги даних з високою швидкістю і надійністю.

7. Стабільність і надійність: Забезпечення максимальної стабільності та безперебійної роботи інформаційної системи, щоб уникнути збоїв у роботі та забезпечити постійну доступність сервісів.

8. Можливість розширення та модернізації: Забезпечення можливості розширення та модернізації системи з метою вдосконалення її функціональності та впровадження нових методів оцінки психологічного стану.

### **4.2.2. Особливості і відмінні риси розроблюваного проєкту**

Особливості та відмінні риси, які дозволять віддати перевагу даному проєкту перед іншими реалізаціями, можуть бути наступні:

1. Інноваційність: Проєкт включає унікальні підходи та методи аналізу психологічного стану, що не використовуються в аналогічних продуктах конкурентів.

2. Широкий функціонал: Система пропонує розширені можливості для збору даних, що дозволить користувачам отримувати більш детальні та точні результати.

3. Відкритість та гнучкість: Система може бути інтегрована з різними платформами, що забезпечує її гнучкість у використанні та можливість адаптації до конкретних потреб користувачів.

4. Висока якість та надійність: Система демонструє високу точність та стабільність у роботі, що забезпечує користувачам надійність результатів та зменшує ризик помилок.

5. Зручність у користуванні: Використання хмарної платформи даних забезпечує інтуїтивно зрозумілий інтерфейс для адміністрування бази даних, що знижує складність у використанні та спрощує налаштування системи.

6. Підтримка та обслуговування: Проєкт забезпечує високий рівень технічної підтримки та регулярні оновлення, що гарантує швидке вирішення проблем та постійне вдосконалення системи.

7. Конкурентоспроможна ціна: Система може мати вигідну ціну у порівнянні з аналогічними рішеннями на ринку, що робить її доступнішою для широкого кола користувачів.

Ці особливості роблять розроблювану автоматизовану систему оцінки психологічного стану людини привабливим вибором для користувачів, забезпечуючи високу якість, інноваційність та доступність.

#### **4.2.3. Якісні показники продукту**

1. Забезпечується ефективно зберігання та обробку великих обсягів даних, що є критичним для системи оцінки психологічного стану людини;

2. Система баз даних забезпечує високий рівень надійності та безпеки збереження даних;

3. Масштабованість дозволяє легко масштабувати систему, що забезпечує можливість обробки зростаючої кількості даних без втрати продуктивності;

4. Система дозволяє легко налаштовувати структуру баз даних відповідно до специфічних потреб проекту, що забезпечує оптимальне зберігання та доступ до інформації;

5. Забезпечується висока швидкість обробки запитів до бази даних, що дозволяє забезпечити швидкий доступ до необхідної інформації для користувачів та системи в цілому;

### **4.3 Конкуренція**

#### **4.3.1. Найбільші виробники аналогічного продукту**

Системи автоматизованої оцінки психологічного стану, такі як «Я-ПСИХОЛОГ», MARTA, Beck Depression Inventory (BDI) Online, Самопоміч та Anima, є лідерами у своїй галузі в нашій країні. Проте, це не означає, що вони єдині на ринку, оскільки існує багато інших компаній, які також займаються розробкою систем для оцінки психологічного стану.

#### **4.3.2. Сутність предмету конкуренції**

Конкуренція є ключовим елементом ринкової економіки, стимулюючи інновації, підвищення якості та зниження цін. У сфері автоматизованих систем оцінки психологічного стану це особливо важливо, оскільки такі системи сприяють виявленню та вчасному лікуванню психічних порушень.

Автоматизовані системи оцінки психологічного стану, як «Я-ПСИХОЛОГ», MARTA, Beck Depression Inventory (BDI) Online, Самопоміч та Anima, використовуються для діагностики та моніторингу психологічного здоров'я. Вони допомагають виявити симптоми депресії, тривоги, стресу та інших психічних розладів, забезпечуючи можливість вчасного втручання.

Основні аспекти конкуренції у цій сфері включають:

1. Технологічна інновація: Впровадження нових алгоритмів для підвищення точності діагностики.
2. Користувацький досвід: Створення інтуїтивно зрозумілих та зручних інтерфейсів.
3. Надійність і точність: Забезпечення точних та надійних результатів через перевірені методики.
4. Доступність та ціна: Вартість системи і її доступність для різних користувачів.
5. Підтримка і обслуговування: Якість технічної підтримки та обслуговування після продажу.

Конкуренція у сфері автоматизованих систем оцінки психологічного стану включає технологічні інновації, зручність використання, точність діагностики, доступність продукту та якість підтримки. Лідери ринку, такі як «Я-ПСИХОЛОГ», MARTA, Beck Depression Inventory (BDI) Online, Самопоміч та Anima, встановлюють високі стандарти, але існує багато інших компаній, що роблять значний внесок у розвиток цієї галузі. Конкуренція сприяє загальному підвищенню якості продуктів і послуг на ринку.

## **4.4 Організаційний план**

### **4.4.1. Формування команди фахівців**

Формування команди фахівців для розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану є критичним етапом, який визначає успіх проекту. Правильний підбір команди, яка володіє необхідними знаннями, навичками та досвідом, забезпечить високу якість розробки та функціонування системи.

Перш за все, необхідно визначити, які фахівці потрібні для розробки такої системи. Основними спеціалістами, які будуть залучені до проекту, є керівник проекту, системний аналітик, програміст, фахівець з баз даних та психолог-консультант.

Керівник проекту повинен мати вищу освіту у сфері інформаційних технологій або управління проектами, а також досвід управління IT-проектами не менше п'яти років. Ця особа буде відповідати за стратегічне планування, управління ризиками та координацію роботи команди. Системний аналітик повинен мати вищу освіту у сфері інформаційних технологій або психології, а також досвід роботи з системами оцінки психологічного стану. Він аналізуватиме вимоги до системи, створюватиме технічні специфікації та забезпечуватиме їх відповідність потребам користувачів.

Програміст, який братиме участь у проекті, повинен мати вищу освіту у сфері комп'ютерних наук або програмування та досвід розробки програмного забезпечення не менше трьох років. Він буде відповідати за написання коду, тестування та налагодження системи.

Фахівець з баз даних повинен мати вищу освіту у сфері інформаційних технологій та досвід роботи з базами даних, такими як MongoDB та SQL, не менше трьох років. Його завданнями будуть проектування, оптимізація та підтримка баз даних системи.

Психолог-консультант, який є необхідним членом команди, повинен мати вищу освіту у сфері психології та досвід роботи у клінічній або консультативній психології не менше п'яти років. Він забезпечуватиме відповідність методів оцінки психологічного стану науковим стандартам та консультуватиме команду з питань психодіагностики.

Для формування команди необхідно пройти декілька етапів. Спочатку визначаються вимоги до кожної посади та розробляється детальний опис вакансій. Далі здійснюється пошук кандидатів через спеціалізовані платформи, соціальні мережі та професійні спільноти. Після цього проводиться первинний відбір резюме, організовуються інтерв'ю для оцінки професійних навичок та особистих якостей кандидатів, а також проводяться технічні інтерв'ю для програмістів та фахівців з баз даних. Важливим етапом є виконання кандидатами тестових завдань для оцінки їх практичних навичок. Останнім етапом є фінальний відбір кандидатів

на основі результатів співбесід та тестових завдань, після чого вибираються найкращі кандидати для кожної посади.

Загальна щомісячна заробітна плата команди фахівців, яка складається з шести основних спеціалістів, може варіюватися в межах від \$22,000 до \$33,000, залежно від їх кваліфікації та досвіду. Враховуючи важливість кожного члена команди, ці витрати є обґрунтованими та необхідними для забезпечення високої якості розроблюваного продукту.

Таким чином, формування команди фахівців для розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану вимагає ретельного підбору кваліфікованих спеціалістів та належного управління процесом їх відбору. Тільки за умов злагодженої роботи команди можна досягти високої ефективності та якості кінцевого продукту, що буде відповідати всім вимогам користувачів та забезпечувати точну і надійну оцінку психологічного стану.

#### **4.4.2. Організаційна структура підприємства**

Організаційна структура підприємства для розробки інформаційного забезпечення автоматизованої системи оцінки психологічного стану повинна бути побудована таким чином, щоб забезпечити ефективну координацію всіх етапів проекту, оптимізувати управління ресурсами та забезпечити високу якість кінцевого продукту. Враховуючи специфіку проекту та його вимоги, структура підприємства може бути організована наступним чином.

На верхньому рівні організаційної структури розташовується генеральний директор, який відповідає за стратегічне управління підприємством, ухвалення ключових рішень та загальну координацію роботи всіх підрозділів. Підпорядковуються генеральному директору кілька ключових керівників: технічний директор, фінансовий директор, директор з маркетингу та директор з операцій.

Технічний директор відповідає за технологічну сторону проекту, включаючи розробку, впровадження та підтримку інформаційного забезпечення. В його

підпорядкуванні знаходяться кілька технічних команд, кожна з яких має свого керівника. До складу технічних команд входять розробники програмного забезпечення, фахівці з баз даних, системні аналітики, тестувальники та фахівці з веб-дизайну. Кожна команда відповідає за конкретний аспект проекту: наприклад, одна команда займається розробкою ядра системи, інша – інтерфейсом користувача, ще одна – тестуванням та забезпеченням якості.

Фінансовий директор займається управлінням фінансовими ресурсами підприємства, бюджетуванням, фінансовим плануванням та звітністю. Під його керівництвом працює фінансовий відділ, який здійснює моніторинг витрат, аналіз фінансових показників та управління грошовими потоками.

Директор з маркетингу відповідає за розробку та реалізацію маркетингової стратегії підприємства, просування продукту на ринку, управління брендом та комунікаціями. В його підпорядкуванні знаходяться маркетинговий відділ, до якого входять спеціалісти з реклами, зв'язків з громадськістю, цифрового маркетингу та аналітики ринку.

Директор з операцій відповідає за операційну діяльність підприємства, забезпечення ефективної роботи внутрішніх процесів та управління ланцюгами постачань. Він керує операційним відділом, який включає менеджерів з логістики, постачання та підтримки клієнтів.

Крім того, у структурі підприємства передбачено наявність юридичного відділу, який підпорядковується безпосередньо генеральному директору. Цей відділ займається правовою підтримкою підприємства, укладанням контрактів, захистом інтелектуальної власності та забезпеченням відповідності законодавству.

Для забезпечення належного управління персоналом на підприємстві функціонує відділ кадрів, який підпорядковується директору з операцій. Відділ кадрів відповідає за підбір, наймання та навчання працівників, управління трудовими відносинами, мотивацію та розвиток персоналу.

Такий підхід до організаційної структури підприємства дозволяє забезпечити ефективну координацію всіх етапів розробки автоматизованої системи оцінки психологічного стану, оптимізувати управління ресурсами та забезпечити високу

якість кінцевого продукту. Злагоджена робота всіх підрозділів та ефективно управління проектом сприятимуть досягненню поставлених цілей та успішній реалізації проекту на ринку.

## **4.5 Стратегія маркетингу**

### **4.5.1 Схема розповсюдження товарів**

Схема розповсюдження інформаційного забезпечення системи людини передбачає кілька основних кроків для ефективного впровадження та використання системи. Головним чином, система може бути доступна через інтернет-платформи та мобільні додатки, що дозволяє користувачам безпосередньо взаємодіяти з нею зручним для них способом. Це забезпечить доступ до системи як для пацієнтів, так і для медичних працівників, які зможуть швидко і просто користуватися її функціями.

Крім того, система може поширюватися через партнерські мережі, такі як медичні установи, клініки або психологічні центри. Співпраця з цими організаціями передбачає укладення угод, що дозволяють інтегрувати систему в їхні існуючі інформаційні інфраструктури.

Схема розповсюдження інформаційного забезпечення цієї системи включає заходи для її ефективного доставки, інтеграції та підтримки через різні канали, роблячи систему доступною та надійною для всіх користувачів.

### **4.5.2 Ціноутворення та реклама**

Ціноутворення, або процес встановлення цін на товари або послуги, включає в себе різноманітні методики, які враховують конкурентну ситуацію на ринку, витрати виробництва, споживчі попити та інші фактори. Реклама є важливим інструментом просування продуктів або послуг, використовуючи різноманітні медіа та стратегії для привертання уваги цільової аудиторії. Тому важливо

розглянути методику визначення цін, методи організації реклами та величину засобів, необхідних для їх успішної реалізації.

#### 1. Методи визначення цін:

- Витратна орієнтація: Включає в себе врахування всіх витрат на виробництво продукту та додавання до ціни прибутку.
- Конкурентна орієнтація: Полягає у встановленні ціни на основі цін конкурентів.
- Попитна орієнтація: Ціна встановлюється на основі рівня попиту на товар.
- Психологічна орієнтація: Враховує психологічні фактори, такі як сприйняття якості товару.

#### 2. Методи організації реклами:

- Телевізійна реклама: Дозволяє досягти великої аудиторії, але може бути високою вартістю.
- Інтернет-реклама: Ефективна через велику кількість користувачів інтернету.
- Друкована реклама: Включає оголошення в газетах, журналах, брошурах тощо.
- Радіореклама: Може бути більш економічною, ніж телевізійна реклама.

#### 3. Величина засобів, необхідних для реалізації реклами:

- Бюджет реклами: Це сума грошей, яка виділяється на рекламні заходи. Вона повинна бути розподілена між різними медіа та кампаніями відповідно до їх ефективності та цілей.
- Креативні ресурси: Рекламні агентства або команди мають забезпечити створення рекламного контенту, який приверне увагу аудиторії.

При визначенні цін і організації реклами важливо бути стратегічним та враховувати конкурентну ситуацію на ринку, потреби цільової аудиторії та бюджет компанії. Ефективна рекламна кампанія разом із правильною ціновою політикою може сприяти збільшенню обсягів продажів та підвищенню популярності серед споживачів.[20]

## **4.6 Стратегія фінансування**

### **4.6.1. Фінансове планування**

Щоб визначити, яка кількість засобів для реалізації проєкту є необхідною потрібно ретельно проаналізувати всі аспекти проєкту, визначивши необхідні ресурси та їх вартість, такі як обладнання, найм працівників, оренда приміщення та інші витрати. Після цього складається детальний бюджет, де визначається сума необхідних коштів і джерела їхнього отримання, наприклад, інвестиційні кошти, банківські кредити, власні кошти підприємства та інше. У цьому процесі можуть бути корисні консультації з фінансовими експертами або фахівцями з бізнес-планування.

### **4.6.2. Джерела фінансових ресурсів**

Для розробки інформаційного забезпечення потрібні фінансові ресурси, які можна отримати з різних джерел.

Основними джерелами фінансових ресурсів є:

1. Інвестиційні кошти - один з потенційних шляхів фінансування розробки інформаційного забезпечення. Інвестори, які вбачають потенціал у розвитку цього напрямку, можуть надати фінансову підтримку в обмін на частку у проєкті або очікувані прибутки.
2. Гранти та субсидії - урядові та неприбуткові організації часто надають гранти та субсидії на розвиток інноваційних проєктів, спрямованих на покращення здоров'я та благополуччя громадян.
3. Приватні інвестиції - окрім інституційних інвесторів, можна також розглядати можливість приваблення приватних інвесторів, які зацікавлені у розвитку новаторського програмного забезпечення.
4. Краудфандинг - це може стати ефективним способом зібрати необхідні фінансові ресурси, залучаючи кошти від широкої аудиторії людей, які підтримують ідею розвитку інноваційного інформаційного забезпечення.

Розробка інформаційного забезпечення для оцінки психологічного стану потребує значних фінансових ресурсів. Однак існує ряд потенційних джерел фінансування, які можуть бути використані для забезпечення цього процесу. Залучення інвестицій, отримання грантів, приватні інвестиції та використання краудфандингу - це лише деякі зі шляхів забезпечення необхідних фінансових ресурсів. Важливо обрати оптимальні джерела фінансування та ефективно використовувати їх для успішного завершення проекту з розробки програмного забезпечення.

## ВИСНОВОК

У результаті виконання даного дипломного проєкту розроблено інформаційне забезпечення автоматизованої системи оцінки психологічного стану. Для досягнення цієї мети було проведено аналіз предметної області, проєктування задачі, практичну реалізацію та розробку бізнес-плану. У кожному розділі виконані певні завдання, описані нижче.

Проведено аналіз предметної області, що охоплює постановку та аналіз даної проблеми, представлення дерева цілей, вивчення існуючих аналогів автоматизованих систем оцінки психологічного стану, визначення поняття бази даних та аналіз різних систем управління ними, також визначено основні вимоги до інформаційного забезпечення автоматизованої системи. Це дозволило чітко встановити задачу, яку було необхідно вирішити.

У процесі проєктування інформаційного забезпечення створено дерево функцій, проаналізовано різні типи баз даних, з яких обрано документоорієнтований тип для розробки системи. Також розглянуто питання розміщення бази даних, що призвело до рішення використовувати хмарні сервери. Створено концептуальну, логічну та фізичну моделі бази даних, які допомогли детально структурувати дані та забезпечити їх ефективне зберігання і обробку.

Практична реалізація проєкту включає вибір програмного інструментарію та розробку бази даних. Для взаємодії з базою даних використано інтерфейс MongoDB Atlas, що забезпечує зручний доступ до даних та їх керування, також наведено методи роботи в терміналі MongoDB Compass. Налаштовно індекси для пришвидшення запитів, створено класи документів, репозиторії для взаємодії з даними та ініціалізація даних.

Крім технічних аспектів, у роботі розроблено бізнес-план впровадження системи. Він включає аналіз ринку, визначення основних конкурентних переваг розробленого продукту, прогнозування фінансових результатів та оцінку потреб продукту у відношенні до проєкту.

Результатом роботи є функціональне інформаційне забезпечення, яке задовольняє визначені цілі та вимоги. Розроблена система може стати основою для подальшого розвитку та вдосконалення систем оцінки психологічного стану, сприяючи зручності, безпеці та ефективності діагностики психологічного стану людини. Вона має потенціал для впровадження у медичну практику, що може значно покращити процеси діагностики та лікування психологічних захворювань, а також підвищити якість життя пацієнтів.

## Список використаних джерел

1. Етапи життєвого циклу розробки ПЗ. [Електронний ресурс] - Режим доступу: <https://icstudio.online/post/etapi-zhittyevogo-ciklu-rozrobki-pz>
2. Singh S. K., Database Systems. Concepts, Design and Applications - Pearson Education. 2009
3. Самопоміч [Електронний ресурс] - Режим доступу: <https://www.samopomi.ch/pro-nas>
4. Silberschatz A., Korth H. F., & Sudarshan S. Database System Concepts - McGraw-Hill. 2010
5. Kahate A., Introduction to Database Management Systems. 2004
6. Kaufmann, M., Meier A., SQL and NoSQL Databases - Springer Nature Switzerland. 2023
7. What is a Cloud Database? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/resources/basics/databases/cloud-databases>
8. Platt M. Document Oriented Databases and Text Processing - GRIN Verlag. 2014
9. What Is MongoDB? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/company/what-is-mongodb>
10. CouchDB [Електронний ресурс] - Режим доступу: <https://couchdb.apache.org/>
11. About RavenDB [Електронний ресурс] - Режим доступу: <https://ravendb.net/about>
12. What is MongoDB Atlas? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/docs/atlas/>
13. Hejlsberg A., Torgersen V., Wiltamuth S., Golde P. The C# Programming Language - Pearson Education. 2008
14. Chaudhary H., Introduction to C++ - CreateSpace Independent Publishing Platform. 2014
15. John M. Zelle. Python Programming - Franklin, Beedle. 2004

16. Visual Studio [Электронный ресурс] - Режим доступа:  
<https://visualstudio.microsoft.com/vs/>

17. JSON and BSON [Электронный ресурс] - Режим доступа:  
<https://www.mongodb.com/resources/basics/json-and-bson>

18. What is MongoDB Compass? [Электронный ресурс] - Режим доступа:  
<https://www.mongodb.com/docs/compass/current/>

19. Indexes [Электронный ресурс] - Режим доступа:  
<https://www.mongodb.com/docs/manual/indexes/>

20. Finch B., How to Write a Business Plan - Kogan Page. 2010

## Додатки

### Додаток А

```

using Application.Common.Models;
using Application.Primitives;
using AspNetCore.Identity.MongoDbCore.Models;
using MongoDB.Bson;

namespace Application.Infrastructure.Identity;

public sealed class ApplicationUser : MongoIdentityUser<ObjectId>, IDocument<ObjectId>
{
    Constructors

    public UserInfo? UserInfo { get; set; }
    public string? AvatarId { get; set; }
    public IList<Guid> RefreshTokens { get; set; } = [];
    public IList<GeneralTestResult> GeneralTestResults { get; set; } = [];
}

public sealed class UserInfo
{
    public string? FirstName { get; set; }
    public string? LastName { get; set; }
    public required DateOnly? Birthdate { get; set; }
    public required Sex? Sex { get; set; }
    public required bool? IsMarried { get; set; }
}

public enum Sex : byte
{
    Male = 0,
    Female = 1
}

public sealed class GeneralTestResult
{
    public Guid Id { get; set; } = Guid.NewGuid();
    public required DateTimeOffset CompletionDate { get; set; }
    public required IList<string> PotentialProblems { get; set; }
    public required IList<AnsweredQuestion> Answers { get; set; }
}

```

## Додаток Б

```

using Application.Common.Constants;
using MongoDB.Bson;
using MongoDB.Bson.Serialization.Attributes;
using MongoDBGenericRepository.Models;

namespace Application.Documents;

public class GeneralTest : IDocument<string>
{
    [BsonRepresentation(BsonType.ObjectId)]
    public string Id { get; set; } = GeneralTestIdentifiers.TestId;

    public int Version { get; set; } = 1;
    public IList<string> AvailableTranslations { get; set; } = [];

    public required IList<QuestionGroup> QuestionGroups { get; set; }
}

public class QuestionGroup
{
    public Guid Id { get; set; } = Guid.NewGuid();
    public IList<Question> Questions { get; set; } = [];
    public IList<QuestionGroup> QuestionGroups { get; set; } = [];
}

public class Question
{
    public Guid Id { get; set; } = Guid.NewGuid();
    public required IList<Answer> Answers { get; set; }

    public IDictionary<string, QuestionTranslation> Translations { get; set; } =
        new Dictionary<string, QuestionTranslation>();
}

public class QuestionTranslation
{
    public required string Title { get; set; }
}

public class Answer
{
    public Guid Id { get; set; } = Guid.NewGuid();
    public IList<string> Tags { get; set; } = [];

    public IDictionary<string, AnswerTranslation> Translations { get; set; } =
        new Dictionary<string, AnswerTranslation>();
}

public class AnswerTranslation
{
    public required string Title { get; set; }
}

```

## Додаток В

```
using Application.Primitives;
using MongoDB.Bson;
using MongoDB.Bson.Serialization.Attributes;

namespace Application.Documents;

public class AdviceList : IDocument<string>
{
    [BsonRepresentation(BsonType.ObjectId)]
    public string Id { get; set; } = ObjectId.GenerateNewId().ToString();

    public int Version { get; set; } = 1;
    public required string Problem { get; set; }
    public IList<string> AvailableTranslations { get; set; } = [];

    public IDictionary<string, AdviceListTranslation> Translations { get; set; } =
        new Dictionary<string, AdviceListTranslation>();
}

public class AdviceListTranslation
{
    public required string[] Advices { get; set; }
}
```

## Додаток Г

```

using System.Text.Json;
using Application.Common.Interfaces;
using Application.Common.Interfaces.Repositories;
using Application.Documents;
using Application.Options;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Options;

namespace Application.Infrastructure.Persistence;

public class DatabaseInitializer(
    IOptions<AssetOptions> assetOptions,
    ILogger<DatabaseInitializer> logger,
    ITestRepository testRepository,
    IAdviceListRepository adviceListRepository) : IDatabaseInitializer
{
    private readonly AssetOptions _assetOptions = assetOptions.Value;

    private readonly JsonSerializerOptions _jsonSerializerOptions = new()
    {
        PropertyNamingPolicy = JsonNamingPolicy.CamelCase
    };

    public async Task InitializeDatabase(CancellationToken cancellationToken = default)
    {
        if (!await testRepository.CheckGeneralTestExistence(cancellationToken))
        {
            await using var fileStream = File.OpenRead(_assetOptions.GeneralTestFilePath);

            var generalTest = await JsonSerializer.DeserializeAsync<GeneralTest>(fileStream, _jsonSerializerOptions,
                cancellationToken);

            if (generalTest == null)
            {
                logger.LogError("Failed to deserialize general test");
                throw new InvalidOperationException("Failed to deserialize general test.");
            }

            await testRepository.SaveGeneralTest(generalTest, cancellationToken);

            logger.LogInformation("General test has been saved during initialization");
        }

        if (!await adviceListRepository.CheckAdviceListsExistence(cancellationToken))
        {
            await using var fileStream = File.OpenRead(_assetOptions.AdviceListsFilePath);

            var adviceLists = await JsonSerializer.DeserializeAsync<IEnumerable<AdviceList>>(fileStream,
                _jsonSerializerOptions,
                cancellationToken);

            if (adviceLists == null)
            {
                logger.LogError("Failed to deserialize advice lists");
                throw new InvalidOperationException("Failed to deserialize advice lists.");
            }

            await adviceListRepository.SaveAdviceListsAsync(adviceLists, cancellationToken);

            logger.LogInformation("Advice lists have been saved during initialization");
        }
    }
}

```