

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПОШУКУ МЕНТОРА
DEVELOPMENT OF MENTOR SEARCH SOFTWARE

Виконав(ла): студент(ка) 4 курсу, групи КНТ-211сп

Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

СВІДЕНКО В.О.

(ПРИЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

СВІДЕНКА Владислава Олексійовича
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення пошуку ментора.
Development of Mentor Search Software

керівник проєкту (роботи) к.т.н., доцент, СКРУПСЬКИЙ Степан Юрійович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 24 ” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області пошуку ментора. 2. Проєктування програмного забезпечення пошуку ментора. 3. Опис програмного забезпечення пошуку ментора. 4. Рекомендації із впровадження програмного забезпечення пошуку ментора.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СКРУПСЬКИЙ С.Ю., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 15 ” квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Владислав СВИДЕНКО

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Степан СКРУПСЬКИЙ

(ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 77 с., 11 табл., 26 рис., 3 дод., 16 джерел.

МОБІЛЬНИЙ ЗАСТОСУНОК, ANDROID, KOTLIN, JAVA, POSTGRESQL, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, МЕНТОР, МЕНТОРСТВО, ПОШУК, ПРОФЕСІЙНИЙ РОЗВИТОК.

Об'єкт дослідження – процес пошуку ментора за допомогою мобільних технологій.

Предмет дослідження – методи та технології розробки мобільних застосунків для пошуку ментора.

Мета роботи – розробка мобільного застосунку для пошуку ментора, який дозволить користувачам знаходити досвідчених професіоналів у різних галузях, спілкуватися з ними та отримувати цінні знання і поради.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мови програмування Kotlin та Java, фреймворк Ktor, система керування базами даних PostgreSQL, середовище розробки Android Studio.

Результати. Розроблено мобільний застосунок для пошуку ментора на платформі Android, який надає користувачам зручні інструменти для пошуку менторів, комунікації, планування та відстеження навчальних сесій, доступу до навчальних ресурсів та інших функцій, необхідних для ефективного менторства та професійного розвитку.

Висновки. У результаті проведеної роботи було розроблено мобільний застосунок для пошуку ментора, який відповідає вимогам та потребам цільової аудиторії.

Галузь використання – професійний розвиток, менторство, мобільні технології.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 77 pages, 11 tables, 26 figures, 3 appendices, 16 sources.

MOBILE APPLICATION, ANDROID, KOTLIN, JAVA, POSTGRESQL, CLIENT-SERVER ARCHITECTURE, MENTOR, MENTORSHIP, SEARCH, PROFESSIONAL DEVELOPMENT.

Object of research - the process of finding a mentor using mobile technologies.

Subject of research - methods and technologies for developing mobile applications for finding a mentor.

The purpose of the work - to develop a mobile application for finding a mentor that will allow users to find experienced professionals in various fields, communicate with them, and receive valuable knowledge and advice.

Materials, methods, and technical means: object-oriented programming, Kotlin and Java programming languages, Ktor framework, PostgreSQL database management system, Android Studio development environment.

Results. A mobile application for finding a mentor on the Android platform has been developed, providing users with convenient tools for searching for mentors, communication, planning and tracking training sessions, accessing educational resources, and other functions necessary for effective mentoring and professional development.

Conclusions. As a result of the work carried out, a mobile application for finding a mentor was developed that meets the requirements and needs of the target audience.

Field of application - professional development, mentoring, mobile technologies.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області пошуку ментора	10
1.1 Огляд існуючих мобільних застосунків для пошуку ментора	10
1.2 Аналіз потреб користувачів у мобільному застосунку для пошуку ментора.....	15
1.3 Формування вимог до мобільного застосунку пошуку ментора	18
1.4 Висновки за розділом	19
2 Проєктування програмного забезпечення пошуку ментора.....	21
2.1 Розробка архітектури мобільного застосунку.....	21
2.2 Проєктування бази даних для зберігання інформації про менторів та учнів.....	22
2.3 Висновки за розділом	27
3 Опис програмного забезпечення пошуку ментора.....	29
3.1 Вибір технологій та інструментів розробки мобільного застосунку пошуку ментора.....	29
3.2 Реалізація серверної частини мобільного застосунку.....	32
3.3 Реалізація клієнтської частини мобільного застосунку	33
3.4 Висновки за розділом	42
4 Рекомендації із впровадження програмного забезпечення пошуку ментора.....	44
4.1 Тестування та налагодження мобільного застосунку	44
4.2 Розробка інструкцій для користувачів мобільного застосунку.....	45
4.3 Висновки за розділом	46
Висновки	48
Перелік джерел посилання	50
Додаток А Текст програми.....	64
Додаток Б Інструкції користувача.....	64
Додаток Б Слайди презентації	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	- Application Programming Interface;
FAQ	- Frequently Asked Questions;
IDE	- Integrated Development Environment;
JSON	- JavaScript Object Notation;
JWT	- JSON Web Token;
PWA	- Progressive Web Apps;
URL	- Uniform Resource Locator.

ВСТУП

У сучасному світі стрімкого розвитку інформаційних технологій та зростаючої потреби в професійному розвитку, все більше людей шукають менторів для отримання знань, досвіду та підтримки у своїй кар'єрі. Ментор - це досвідчений професіонал, який може допомогти менш досвідченій людині (учню) розвинути свої навички, отримати цінні поради та рекомендації [1].

На сьогоднішній день, ІТ-індустрія пропонує широкий спектр технологій та інструментів для створення програмних рішень, які допомагають людям знаходити менторів. Веб-платформи, мобільні застосунки та соціальні мережі дозволяють користувачам шукати менторів, спілкуватися з ними та отримувати корисні знання.

Особливо актуальним є розробка мобільних застосунків для пошуку ментора, оскільки мобільні пристрої стали невід'ємною частиною нашого життя і забезпечують зручний доступ до інформації та послуг у будь-який час та у будь-якому місці. Сучасні мобільні технології, такі як React Native, Flutter, iOS та Android, дозволяють створювати високопродуктивні та інтуїтивно зрозумілі застосунки, які можуть задовольнити потреби користувачів у пошуку ментора [2].

Метою даної дипломної роботи є розробка мобільного застосунку для пошуку ментора, який дозволить користувачам знаходити досвідчених професіоналів у різних галузях, спілкуватися з ними та отримувати цінні знання і поради.

Об'єктом дослідження є процес пошуку ментора за допомогою мобільних технологій.

Предметом дослідження є методи та технології розробки мобільних застосунків для пошуку ментора.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- проаналізувати предметну область пошуку ментора та існуючі рішення;

- сформулювати вимоги до мобільного застосунку пошуку ментора;
- спроектувати архітектуру та базу даних мобільного застосунку;
- розробити серверну та клієнтську частини мобільного застосунку;
- провести тестування та налагодження мобільного застосунку;
- розробити інструкції для користувачів мобільного застосунку.

Результатом даної роботи буде мобільний застосунок для пошуку ментора, який дозволить користувачам ефективно знаходити та взаємодіяти з досвідченими професіоналами, отримувати цінні знання та поради для свого професійного розвитку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПОШУКУ МЕНТОРА

1.1 Огляд існуючих мобільних застосунків для пошуку ментора

У сучасному світі пошук ментора для особистісного та професійного розвитку є важливою задачею, яка допомагає особам досягти своїх цілей. З розвитком мобільних технологій з'явилася можливість використовувати спеціалізовані застосунки для знаходження ідеального ментора. Ці застосунки надають користувачам широкий спектр функцій, включаючи фільтрацію за досвідом, галузевою спеціалізацією та географічним розташуванням менторів [2].

Мобільні застосунки для пошуку ментора зазвичай пропонують інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко орієнтуватися в доступних опціях. Наприклад, багато з цих застосунків включають головну сторінку з пошуковою панеллю, де можна ввести ключові слова або категорії, які відповідають потребам користувача. Інша важлива особливість - це профілі менторів, які надають детальну інформацію про їхній досвід, освіту, області експертизи та відгуки від інших користувачів.

Безпека та конфіденційність також є ключовими аспектами цих застосунків. Вони забезпечують захист даних користувачів за допомогою сучасних методів шифрування та верифікації профілів, щоб забезпечити, що взаємодія між ментором і менті відбувається в безпечному середовищі.

Ці застосунки часто включають додаткові інструменти, такі як календарі для планування зустрічей, чати для спілкування та навіть модулі навчання, які допомагають користувачам максимально використовувати час з ментором.

Розглянемо приклади інтерфейсу мобільних застосунків.

На рисунку 1.1 можна побачити інтерфейс мобільного застосунку, який має кілька ключових функцій для планування навчального процесу, що можуть бути важливими також і для застосунків пошуку ментора:

- розклад - ця функція дозволяє користувачам відслідковувати свої навчальні сесії, зустрічі та інші події в календарі; може бути інструментом для планування зустрічей з менторами, відстеження їх тривалості та перегляду запланованих активностей;
- організований навчальний матеріал - дозволяє користувачам зберігати та організовувати навчальні матеріали, такі як підручники, замітки та відео; у контексті пошуку ментора може бути використана для зберігання ресурсів, рекомендованих менторами, а також для відстеження прогресу в навчанні;
- онлайн-чат - дозволяє користувачам спілкуватися зі своїми менторами в режимі реального часу; важлива функція для застосунків пошуку ментора, оскільки допомагає підтримувати безперервний діалог між ментором і менті, дозволяючи швидко вирішувати запитання або обговорювати ідеї;
- онлайн-тести - користувачі можуть проходити тести для оцінки своїх знань і навичок; у застосунку для пошуку ментора це може бути інструментом для оцінки розвитку навичок менті під керівництвом ментора, а також способом виявлення областей, що потребують додаткової уваги.

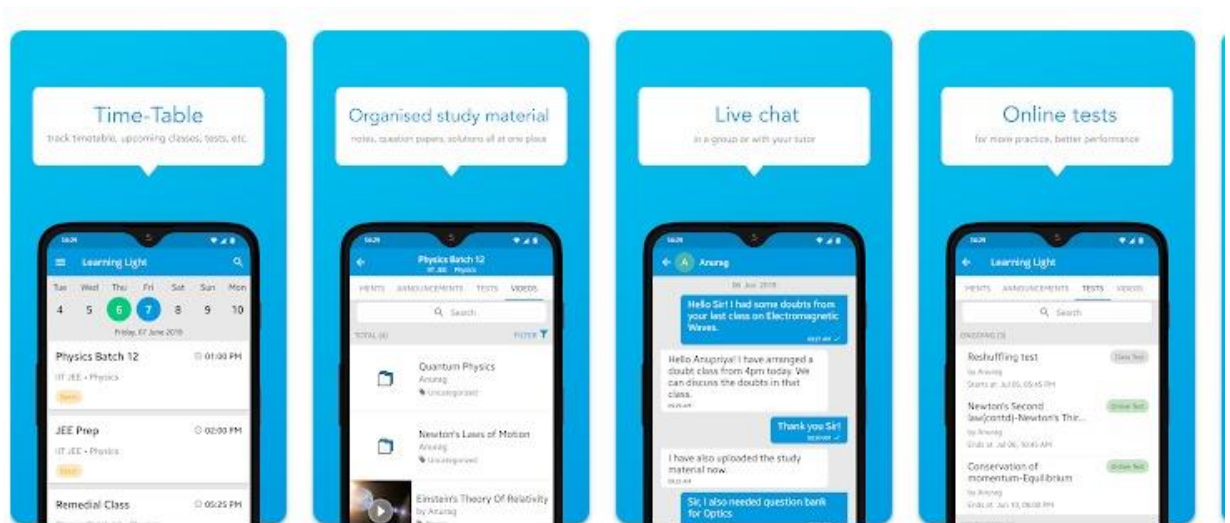


Рисунок 1.1 – Mentor App by The Soultalk [3]

На рисунку 1.2 надано інтерфейс мобільного застосунку, який надає різноманітні функції для навчання та підготовки з ментором:

- досвід персоналізованого навчального шляху - дозволяє користувачам отримати індивідуальний план навчання, який враховує їхні особисті цілі та потреби; користувачі можуть відстежувати свої досягнення та прогрес через інтерактивні дашборди, що включають детальні звіти про засвоєні навички і знання;

- підготовка до майбутніх зустрічей з ментором - надає інструменти для ефективної підготовки до зустрічей з менторами, включно з можливістю записувати питання, планувати порядок денний і встановлювати нагадування про майбутні сесії; спрямована на підвищення продуктивності взаємодії з ментором;

- навчання власним темпом з допомогою коротких відео - пропонує велику бібліотеку відео, які охоплюють широкий спектр тем; короткі відеоматеріали дозволяють користувачам зручно навчатися, коли їм зручно, і вивчати матеріал маленькими порціями для кращого засвоєння інформації;

- практика нових концепцій у симульованих реальних сценаріях - дозволяє користувачам відточувати засвоєні навички через інтерактивні симуляції та сценарії, які імітують реальні виклики та ситуації; сприяє глибшому розумінню та застосуванню навчального матеріалу в практичних умовах.

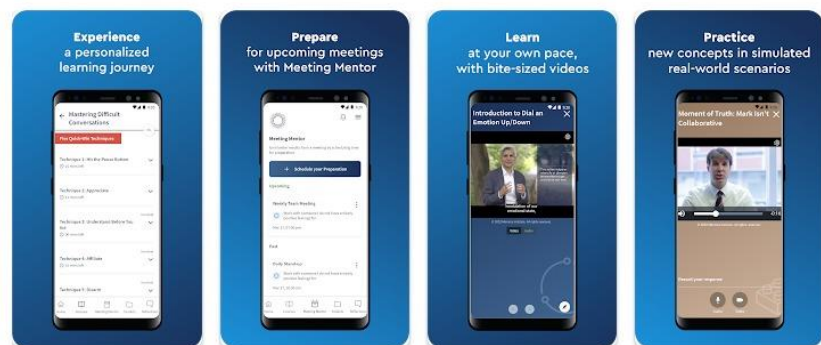


Рисунок 1.2 -Mentora [4]

На рисунку 1.3 інтерфейс мобільного застосунку PushFar, який пропонує широкий спектр функцій для ефективного пошуку та взаємодії з менторами:

- світовий лідер серед застосунків для менторства - відзначається, що застосунок надав 15 мільйонів годин менторства, що підкреслює його популярність та велику базу активних користувачів;
- пошук менторів, обмін досвідом і подальший розвиток - застосунок дозволяє знаходити менторів, обмінюватися з ними досвідом та вести замітки про менторські сесії, сприяючи таким чином глибшому навчанню та професійному розвитку;
- безперервне відстеження менторської діяльності та зв'язок - включає відстеження цілей та нагадувань, що допомагає користувачам залишатися на зв'язку з менторами та відслідковувати свої досягнення і прогрес;
- створення власного профілю та перегляд профілів понад 100,000 менторів з різних галузей по всьому світу - застосунок пропонує можливість створити деталізований профіль, що включає професійний досвід та освіту, а також переглядати профілі потенційних менторів для вибору найбільш підходящого кандидата для менторства.

Цей застосунок є комплексним інструментом, який забезпечує всебічну підтримку в процесі менторства, від пошуку менторів до планування та відстеження менторських сесій.

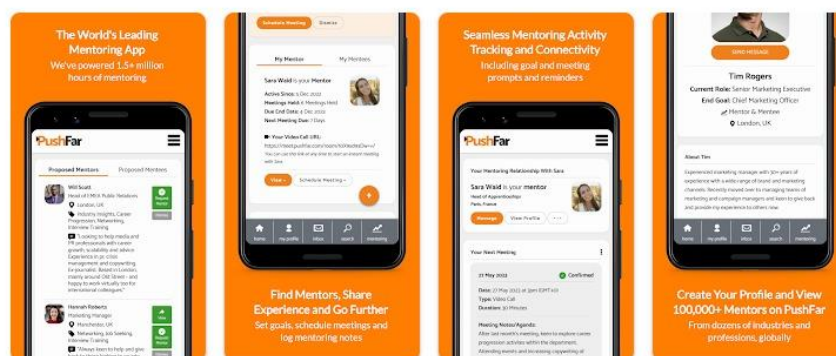


Рисунок 1.3 - PushFar - The Mentoring App [5]

Таблиця 1.1 порівнює ключові функціональні можливості трьох обговорених мобільних застосунків для менторства.

Таблиця 1.1 – Порівняння мобільних застосунків

Функціонал / Застосунок	Mentor	Mentora	PushFar
Персоналізований навчальний план	✓	✓	
Підготовка до зустрічей з ментором		✓	✓
Онлайн-навчання через відео	✓	✓	
Інтерактивні симуляції для практики	✓	✓	
Відстеження менторських сесій			✓
Обмін досвідом із менторами		✓	✓
Профіль користувача і доступ до менторів	✓	✓	✓
Онлайн-чат з менторами			✓

Ця таблиця допоможе вам легше порівняти функціональні можливості та вибрати застосунок, який найкраще відповідає вашим потребам у менторстві.

Створюючи новий застосунок для менторства, важливо включити функції, які максимально підвищать його ефективність та корисність для користувачів. На основі аналізу існуючих застосунків можна виділити наступні ключові функції, які доцільно перенести та адаптувати у новому застосунку:

- персоналізований навчальний план - забезпечення можливості користувачам створювати індивідуальні навчальні шляхи на основі їхніх особистих цілей та потреб є ключовим для підвищення ефективності менторства;

- підготовка до зустрічей з ментором - інструменти для планування зустрічей, ведення заміток, та нагадувань про майбутні сесії допоможуть

користувачам ефективно організувати свій час і підвищити продуктивність взаємодії з менторами;

- відстеження менторських сесій і цілей - інтеграція інструментів для відстеження прогресу, встановлення та досягнення цілей забезпечить користувачам чітке розуміння своїх досягнень і напрямків для подальшого розвитку;

- онлайн-навчання через відео - застосування бібліотеки коротких навчальних відео дозволить користувачам навчатися в зручний для них час і темп, підвищуючи гнучкість і доступність навчання;

- онлайн-чат з менторами - можливість спілкування з менторами в режимі реального часу є критично важливою для забезпечення оперативної підтримки та обговорення важливих питань;

- інтерактивні симуляції для практики - інтеграція інтерактивних вправ та симуляцій, що імітують реальні ситуації, дозволить користувачам відточувати набуті навички в контрольованих умовах;

- профіль користувача та доступ до менторів - створення деталізованих профілів з можливістю перегляду та вибору менторів з усього світу забезпечить користувачам широкі можливості для налагодження корисних контактів та розвитку мережі професійних зв'язків.

1.2 Аналіз потреб користувачів у мобільному застосунку для пошуку ментора

Для розробки ефективного мобільного застосунку для пошуку ментора необхідно глибоко зрозуміти потреби користувачів. Основні потреби можна класифікувати та аналізувати за кількома ключовими напрямками:

- легкість використання - користувачі очікують, що застосунок буде інтуїтивно зрозумілим та легким у навігації. Важливо, щоб інтерфейс був організований так, щоб користувачі могли швидко знаходити потрібні функції без додаткових пояснень;

- персоналізація- наявність можливості налаштування профілю, фільтрів пошуку та навчальних планів з урахуванням особистих інтересів і цілей користувача є критично важливою. Це допомагає створити більш особистий і залучаючий досвід використання застосунку;
- доступність менторів - користувачі прагнуть мати доступ до широкого спектру менторів з різних галузей та з різними рівнями досвіду. Застосунок повинен забезпечувати легкий доступ до профілів менторів, їхніх кваліфікацій та відгуків від інших користувачів;
- комунікація - наявність засобів для легкої та ефективної комунікації між ментором та менті, таких як месенджери або відеодзвінки, є необхідною. Це дозволяє підтримувати зв'язок без необхідності використання зовнішніх платформ;
- відстеження прогресу - користувачі цінують можливість відслідковувати свій прогрес у навчанні та розвитку навичок. Інтеграція інструментів для ведення звітів, оцінювання успіхів і встановлення цілей є важливою;
- навчальні ресурси- доступ до навчальних матеріалів, таких як відео, статті, електронні книги та інтерактивні курси, що допомагають у саморозвитку та підготовці до зустрічей з ментором;
- гнучкість у виборі ментора - здатність вибирати менторів на основі специфічних параметрів, таких як галузь знань, досвід роботи, географічне розташування, і мова спілкування;
- безпека та конфіденційність - гарантії безпеки особистих даних та конфіденційності комунікацій між користувачами є обов'язковими. Важливо впровадити надійні заходи захисту даних;
- підтримка користувачів - ефективна служба підтримки, яка може швидко реагувати на запити користувачів, допомагати у вирішенні технічних проблем або відповідати на питання про використання застосунку;
- інтеграція з іншими сервісами - можливість інтеграції з іншими платформами та інструментами, такими як календарі, соціальні мережі та

професійні сервіси, для забезпечення зручності та розширення функціональності.

Врахування цих аспектів допоможе розробити мобільний застосунок для пошуку ментора, який буде відповідати потребам широкої аудиторії користувачів та сприяти їхньому ефективному навчанню та професійному розвитку (табл. 1.2).

Таблиця 1.2 - Аналіз потреб користувачів

Категорія	Опис
Легкість використання	Інтуїтивно зрозумілий інтерфейс, легкість навігації
Персоналізація	Індивідуалізація налаштувань профілю та пошуку
Доступність менторів	Широкий вибір менторів з різних галузей
Комунікація	Засоби для легкої комунікації між ментором та менті
Відстеження прогресу	Інструменти для оцінки та відстеження навчального прогресу
Навчальні ресурси	Доступ до різноманітних навчальних матеріалів
Гнучкість у виборі ментора	Можливість вибору менторів за різними параметрами
Безпека та конфіденційність	Заходи для захисту особистих даних та приватності комунікацій
Підтримка користувачів	Ефективна служба підтримки для вирішення запитів та проблем
Інтеграція з іншими сервісами	Сумісність з іншими платформами для забезпечення більшої зручності

Ця таблиця надає чітке уявлення про ключові аспекти, які слід врахувати при розробці.

1.3 Формування вимог до мобільного застосунку пошуку ментора

На основі аналізу потреб користувачів, можна сформулювати конкретні вимоги до мобільного застосунку для пошуку ментора, який буде відповідати очікуванням і потребам цільової аудиторії. Нижче перераховані основні вимоги до функціональності та інтерфейсу застосунку:

- інтуїтивно зрозумілий інтерфейс - застосунок має бути простим у використанні для користувачів усіх вікових категорій, з чіткою навігацією та мінімалістичним дизайном;

- персоналізовані налаштування - користувачі повинні мати змогу налаштовувати застосунок під свої особисті та професійні потреби, включаючи фільтри пошуку менторів, налаштування нотифікацій та кастомізацію профілю;

- широкий вибір менторів - застосунок має забезпечити доступ до різноманітної бази менторів з різних спеціалізацій і культур, дозволяючи користувачам вибирати тих, хто найкраще відповідає їхнім цілям;

- вбудовані засоби комунікації - необхідно інтегрувати засоби для текстового, голосового та відео спілкування в рамках застосунку для забезпечення ефективної взаємодії між ментором і менті;

- функції відстеження прогресу - застосунок повинен включати інструменти для відстеження цілей, планування сесій та звітності про навчальний прогрес, з можливістю оцінки та відгуків;

- доступ до навчальних ресурсів - інтеграція навчальних матеріалів, таких як відео, статті, електронні книги та інтерактивні курси, повинна бути однією з ключових функцій;

- гнучкість у виборі ментора - користувачі повинні мати можливість вибирати менторів на основі детальних параметрів, включаючи рейтинги, відгуки та їх професійний досвід;

- безпека та конфіденційність - необхідно забезпечити високий рівень захисту особистих даних та забезпечити безпечне спілкування в рамках застосунку;

- ефективна підтримка користувачів - служба підтримки повинна бути доступною 24/7 для вирішення технічних проблем і відповідей на запитання користувачів;

- інтеграція з іншими платформами - застосунок має бути сумісним з іншими використовуваними користувачами сервісами, такими як календарі та соціальні мережі, для підвищення зручності використання.

Задоволення цих вимог забезпечить створення мобільного застосунку, який буде корисним і зручним для широкого кола користувачів, сприяючи ефективному менторству та професійному розвитку.

1.4 Висновки за розділом

У першому розділі було проведено аналіз предметної області пошуку ментора та розглянуто існуючі мобільні застосунки, які надають таку можливість.

Огляд існуючих рішень показав, що сучасні мобільні застосунки для пошуку ментора пропонують різноманітні функції, такі як персоналізовані навчальні плани, підготовка до зустрічей з ментором, онлайн-навчання через відео та інтерактивні симуляції, відстеження менторських сесій та цілей, онлайн-чат з менторами та інші. Ці функції спрямовані на підвищення ефективності та зручності взаємодії між менторами та учнями.

Аналіз потреб користувачів дозволив визначити ключові аспекти, які слід врахувати при розробці нового мобільного застосунку для пошуку ментора. Серед них - легкість використання, персоналізація, доступність менторів, ефективна комунікація, відстеження прогресу, доступ до навчальних ресурсів, гнучкість у виборі ментора, безпека та конфіденційність, підтримка користувачів та інтеграція з іншими сервісами.

На основі аналізу потреб користувачів були сформовані конкретні вимоги до функціональності та інтерфейсу мобільного застосунку пошуку ментора. Ці вимоги включають інтуїтивно зрозумілий інтерфейс, персоналізовані налаштування, широкий вибір менторів, вбудовані засоби комунікації, функції відстеження прогресу, доступ до навчальних ресурсів, гнучкість у виборі ментора, високий рівень безпеки та конфіденційності, ефективну підтримку користувачів та інтеграцію з іншими платформами.

Таким чином, проведений аналіз предметної області, огляд існуючих рішень та визначення потреб користувачів дозволили сформулювати чітке розуміння вимог та функціональних можливостей, які повинен мати новий мобільний застосунок для пошуку ментора. Це є необхідною основою для подальшого проєктування та розробки ефективного та зручного програмного забезпечення, яке відповідатиме очікуванням і потребам цільової аудиторії.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПОШУКУ МЕНТОРА

2.1 Розробка архітектури мобільного застосунку

Архітектура мобільного застосунку є ключовим аспектом при його проєктуванні, оскільки вона визначає загальну структуру системи, включаючи спосіб взаємодії між різними компонентами програми та користувачем. Серед різних архітектур, які можна застосувати для мобільних застосунків, включають монолітну, мікросервісну, та клієнт-серверну [6].

Монолітна архітектура використовує єдину базу коду для всіх функцій, що може ускладнити масштабування та оновлення. Мікросервісна архітектура забезпечує високу гнучкість і легкість в управлінні окремими компонентами, але вимагає більш складного управління зв'язками та залежностями між сервісами [7].

Клієнт-серверна архітектура, яку я обрав для розробки мобільного застосунку для пошуку ментора, передбачає розділення функціональності застосунку на дві основні частини: клієнт (мобільний застосунок, встановлений на пристрої користувача) та сервер (віддалений сервер, який обробляє запити від клієнта і керує централізованою базою даних) [8].

Ця модель має кілька переваг для мобільного застосунку, орієнтованого на менторство. По-перше, вона забезпечує високу швидкість обробки даних, оскільки запити обробляються на потужному сервері, а не на мобільному пристрої. По-друге, забезпечується централізоване управління даними, що є важливим для застосунків, які використовують велику кількість даних та потребують їх частого оновлення. Централізація даних також сприяє кращій безпеці, оскільки всі дані зберігаються і обробляються на сервері, що дозволяє застосувати сучасні методи шифрування та захисту інформації.

Також, ця архітектура сприяє легшому масштабуванню застосунку. Збільшення кількості користувачів та запитів може бути ефективно

оброблено шляхом збільшення ресурсів сервера, без необхідності зміни клієнтської частини застосунку.

Це дозволяє застосунку зростати відповідно до збільшення користувальницької бази, не втрачаючи у продуктивності або якості обслуговування.

На завершення, клієнт-серверна архітектура виявляється найбільш підходящою для застосунків, що вимагають великої обробки даних та високого рівня взаємодії з користувачами, як це і є у випадку мобільного застосунку для пошуку ментора, що забезпечує надійність, масштабованість та високу продуктивність.

2.2 Проєктування бази даних для зберігання інформації про менторів та учнів

Ефективне зберігання та керування даними про менторів та учнів є критично важливим для успіху мобільного застосунку пошуку ментора. Оптимальне рішення бази даних має відповідати вимогам великої кількості запитів, високого рівня безпеки, а також забезпечувати гнучкість у зберіганні складних взаємозв'язків між різними типами даних.

Існують різні типи баз даних, такі як реляційні, NoSQL, і графові бази даних:

- реляційні бази даних (наприклад, MySQL, PostgreSQL) забезпечують строгу структуру та міцну транзакційну підтримку, що робить їх ідеальними для систем, де потрібна надійність і консистентність [10];

- NoSQL бази даних (наприклад, MongoDB, Cassandra) надають велику гнучкість у зберіганні даних та масштабуванні, що є корисним для застосунків з великою кількістю неструктурованих даних;

- графові бази даних (наприклад, Neo4j, Amazon Neptune) ідеально підходять для зберігання та аналізу складних мереж взаємозв'язків та взаємодій між об'єктами [11].

Для мобільного застосунку пошуку ментора обрано реляційну базу даних. Вибір зумовлений кількома перевагами, які реляційні бази даних пропонують.

Структурованість та організованість даних. Реляційні бази даних дозволяють ефективно організовувати дані у вигляді таблиць, що мають чітко визначені поля та зв'язки. Це забезпечує легкість у керуванні даними про менторів і учнів, їх освітній досвід, кваліфікації, відгуки тощо.

Надійність та інтегритет даних. Реляційні бази даних підтримують транзакційність та забезпечують цілісність даних, що є важливим для збереження точної інформації про користувачів і менторські взаємодії.

Масштабованість. Сучасні реляційні бази даних, такі як PostgreSQL, добре масштабуються, що дозволяє збільшувати обсяги даних без значних витрат на ресурси.

Безпека. Реляційні бази даних надають високий рівень безпеки, включаючи контроль доступу на рівні користувача та шифрування даних, що важливо для захисту персональної інформації користувачів.

Завдяки цим перевагам, реляційна база даних є відмінним вибором для застосунку, орієнтованого на створення та підтримку широкої мережі менторів та учнів.

PostgreSQL є однією з найпопулярніших реляційних систем управління базами даних, що відзначається своєю масштабованістю, надійністю та гнучкістю. Ця система підтримує складні запити, транзакційну цілісність та має велику кількість розширень. Використання PostgreSQL дозволяє ефективно виконувати операції з великою кількістю транзакцій, що ідеально підходить для динамічного та вимогливого середовища мобільного застосунку. Завдяки вбудованій підтримці JSON, геопросторових даних і великій кількості доступних типів даних, PostgreSQL забезпечує великі можливості для зберігання різноманітної інформації, що є важливим для комплексного зберігання профілів користувачів, менторів, відгуків та інших даних у застосунку.

Таблиця "Users" зберігає основну інформацію про всіх користувачів застосунку, включно з менторами та учнями. Кожен запис містить дані про особу, такі як ім'я, прізвище, електронна пошта, а також інформацію для входу та ідентифікації ролі користувача в системі (табл. 2.1).

Таблиця 2.1 - Users (Користувачі)

Назва атрибуту	Тип даних	Опис
id	serial	Унікальний ідентифікатор користувача
username	varchar	Ім'я користувача
password_hash	varchar	Хеш пароля
email	varchar	Електронна пошта
first_name	varchar	Ім'я
last_name	varchar	Прізвище
profile_picture	varchar	URL до фотографії профілю
role	varchar	Роль користувача (mentor, mentee)
created_at	timestamp	Дата створення облікового запису
updated_at	timestamp	Дата останнього оновлення профілю

Таблиця "Mentors" розширює інформацію про користувачів, які виступають в ролі менторів. Вона включає деталі про біографію ментора, його спеціалізації та загальний рейтинг, отриманий від учнів (табл. 2.2).

Таблиця 2.2 - Mentors (Ментори)

Назва атрибуту	Тип даних	Опис
id	serial	Унікальний ідентифікатор ментора
user_id	int	Ідентифікатор користувача, який є ментором
bio	text	Біографія ментора
rating	numeric	Рейтинг ментора
specializations	varchar[]	Масив спеціалізацій

Таблиця "MenteePreferences" зберігає персоналізовані побажання учнів, що включають інформацію про їхні переваги у навчанні, спеціалізації менторів, яких вони шукають, та інші критерії (табл. 2.3).

Таблиця 2.3 - MenteePreferences (Побажання менти)

Назва атрибуту	Тип даних	Опис
id	serial	Унікальний ідентифікатор запису
user_id	int	Ідентифікатор користувача, який є менти
preferences	jsonb	JSON об'єкт з побажаннями менти

Таблиця "Sessions" використовується для організації сесій між менторами та учнями. Вона включає інформацію про час проведення сесії, учасників, статус та відгуки (табл. 2.4).

Таблиця 2.4 - Sessions (Сесії)

Назва атрибуту	Тип даних	Опис
id	serial	Унікальний ідентифікатор сесії
mentor_id	int	Ідентифікатор ментора
mentee_id	int	Ідентифікатор менти
scheduled_time	timestamp	Запланований час сесії
status	varchar	Статус сесії (planned, completed, cancelled)
feedback	text	Відгук про сесію
created_at	timestamp	Дата створення сесії

Таблиця "Messages" зберігає всі повідомлення, що обмінюються між учасниками сесій. Це забезпечує зберігання історії спілкування та можливість її аналізу для покращення сервісу (табл. 2.5).

Таблиця 2.5 - Messages (Повідомлення)

Назва атрибуту	Тип даних	Опис
id	serial	Унікальний ідентифікатор повідомлення
session_id	int	Ідентифікатор сесії
sender_id	int	Ідентифікатор відправника повідомлення
content	text	Текст повідомлення
timestamp	timestamp	Час відправлення повідомлення

Таблиця "Resources" включає інформацію про навчальні матеріали, які доступні у застосунку. Це можуть бути відео, статті, електронні книги або курси, які ментори можуть рекомендувати своїм учням або які можуть бути використані для самостійного навчання (табл. 2.6).

Таблиця 2.6 - Resources (Ресурси)

Назва атрибуту	Тип даних	Опис
id	serial	Унікальний ідентифікатор ресурсу
title	varchar	Назва ресурсу
description	text	Опис ресурсу
type	varchar	Тип ресурсу (video, article, ebook, course)
url	varchar	URL до ресурсу
created_at	timestamp	Дата додавання ресурсу

Ця структура бази даних відображає комплексну систему для ефективного керування інформацією у мобільному застосунку пошуку ментора, забезпечуючи необхідну гнучкість, безпеку, і масштабованість (рис. 2.1).

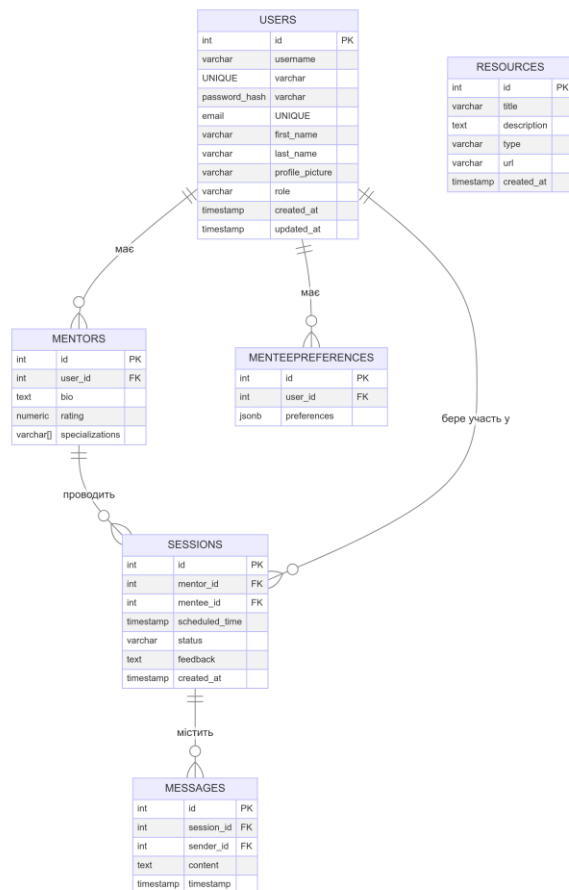


Рисунок 2.1 - Схема бази даних

2.3 Висновки за розділом

У другому розділі було розглянуто ключові аспекти проєктування програмного забезпечення пошуку ментора, а саме розробку архітектури мобільного застосунку та проєктування бази даних для зберігання інформації про менторів та учнів.

При розробці архітектури мобільного застосунку були розглянуті різні підходи, такі як монолітна, мікросервісна та клієнт-серверна архітектури. Для мобільного застосунку пошуку ментора було обрано клієнт-серверну архітектуру, яка передбачає розділення функціональності на клієнтську частину (мобільний застосунок) та серверну частину (віддалений сервер з централізованою базою даних). Ця архітектура має ряд переваг, зокрема

високу швидкість обробки даних, централізоване управління даними, масштабованість та безпеку.

Для зберігання та керування даними про менторів та учнів було обрано реляційну базу даних PostgreSQL. Вибір PostgreSQL зумовлений її перевагами, такими як структурованість та організованість даних, надійність та цілісність даних, масштабованість та безпека. Було спроектовано структуру бази даних, яка включає таблиці для зберігання інформації про користувачів (Users), менторів (Mentors), побажання учнів (MenteePreferences), сесії менторства (Sessions), повідомлення (Messages) та навчальні ресурси (Resources). Ця структура забезпечує ефективне керування та доступ до даних, необхідних для функціонування мобільного застосунку пошуку ментора.

Таким чином, у другому розділі було спроектовано архітектуру мобільного застосунку та структуру бази даних, які відповідають вимогам та потребам програмного забезпечення пошуку ментора. Обрана клієнт-серверна архітектура та реляційна база даних PostgreSQL забезпечать високу продуктивність, масштабованість, безпеку та ефективне керування даними. Це створює міцну основу для подальшої розробки та реалізації функціональності мобільного застосунку пошуку ментора.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПОШУКУ МЕНТОРА

3.1 Вибір технологій та інструментів розробки мобільного застосунку пошуку ментора

При розробці мобільного застосунку для пошуку ментора важливо обрати відповідні технології та інструменти, які дозволять ефективно реалізувати необхідний функціонал, забезпечать високу продуктивність та зручність використання застосунку.

Існує декілька популярних технологій для розробки мобільних застосунків:

- native (нативна) розробка - створення окремих застосунків для кожної мобільної платформи (iOS, Android) з використанням відповідних мов програмування (Swift/Objective-C для iOS, Java/Kotlin для Android) та інструментів (Xcode, Android Studio). Нативна розробка забезпечує найвищу продуктивність та доступ до всіх можливостей платформи [12];

- cross-platform (крос-платформна) розробка - створення єдиного застосунку, який може працювати на різних платформах. Популярні фреймворки для крос-платформної розробки включають React Native, Flutter, Xamarin. Ці інструменти дозволяють писати код на мовах JavaScript, Dart або C# відповідно, і генерувати нативні застосунки для iOS та Android. Крос-платформна розробка економить час і ресурси, але може мати деякі обмеження у продуктивності та доступі до специфічних можливостей платформи [13];

- progressive Web Apps (PWA) - це веб-застосунки, які використовують сучасні веб-технології (HTML, CSS, JavaScript) і можуть встановлюватись на мобільні пристрої та працювати як нативні застосунки. PWA мають хорошу продуктивність, працюють офлайн і легко розповсюджуються, але мають обмежений доступ до апаратних можливостей пристрою (табл. 3.1).

Таблиця 3.1 - Порівняння технологій розробки мобільних застосунків

Технологія	Переваги	Недоліки
Native (нативна)	Висока продуктивність; Повний доступ до можливостей платформи; Широка підтримка пристроїв; Зрілість та стабільність платформи	Високі витрати на розробку для кожної платформи окремо; Складність в підтримці кількох кодових баз
Cross-platform (крос-платформна)	Економія часу і ресурсів; Єдина кодова база для обох платформ	Можливі обмеження у продуктивності; Обмежений доступ до специфічних функцій платформи
Progressive Web Apps (PWA)	Легка розповсюджуваність; Працюють офлайн; Хороша продуктивність	Обмежений доступ до апаратних можливостей пристрою; Менша продуктивність у порівнянні з нативними застосунками

Для розробки мобільного застосунку пошуку ментора обрано нативну розробку для платформи Android з використанням мов програмування Java та Kotlin. Цей вибір має такі переваги:

- висока продуктивність - нативна розробка дозволяє максимально використовувати ресурси пристрою і забезпечує найвищу швидкодію застосунку;

- повний доступ до можливостей платформи - нативна розробка надає доступ до всіх API та функцій операційної системи Android, що дозволяє створювати багатифункціональні застосунки;

- широка підтримка пристроїв - Android має найбільшу частку ринку мобільних операційних систем, тому нативний Android-застосунок зможе охопити значну аудиторію користувачів;

- зрілість та стабільність платформи - Android має тривалу історію та велику екосистему інструментів, бібліотек і документації, що полегшує розробку та підтримку застосунків;

- можливість використання Kotlin - сучасної та виразної мови програмування, яка повністю сумісна з Java і має ряд переваг, таких як лаконічність, null-безпека та розширені функції;

Отже, нативна розробка для Android з використанням Java та Kotlin є оптимальним вибором для створення високопродуктивного та функціонального мобільного застосунку пошуку ментора, який зможе повністю задовольнити потреби користувачів платформи (табл. 3.2).

Таблиця 3.2 - Переваги нативної розробки для Android

Перевага	Опис
Висока продуктивність	Нативна розробка дозволяє максимально використовувати ресурси пристрою і забезпечує найвищу швидкість застосунку.
Повний доступ до можливостей платформи	Нативна розробка надає доступ до всіх API та функцій операційної системи Android, що дозволяє створювати багатофункціональні застосунки.
Широка підтримка пристроїв	Android має найбільшу частку ринку мобільних операційних систем, тому нативний Android-застосунок зможе охопити значну аудиторію користувачів.
Зрілість та стабільність платформи	Android має тривалу історію та велику екосистему інструментів, бібліотек і документації, що полегшує розробку та підтримку застосунків.
Можливість використання Kotlin	Kotlin - сучасна та виразна мова програмування, яка повністю сумісна з Java і має ряд переваг, таких як лаконічність, null-безпека та розширені функції.

3.2 Реалізація серверної частини мобільного застосунку

Для реалізації серверної частини мобільного застосунку для пошуку ментора використана мова програмування Kotlin разом із фреймворком Ktor, який підходить для створення масштабованих серверних додатків. Нижче опис реалізації основних функцій (табл. 3.3).

Таблиця 3.3 – Таблиця функцій мобільного застосунку для пошуку ментора

Назва функції	Опис
registerUser	Реєстрація нового користувача в системі з основною інформацією.
authenticateUser	Вхід користувача в систему за допомогою логіну та пароля.
updateUserProfile	Зміна особистої та професійної інформації користувача.
setUserPreferences	Налаштування фільтрів пошуку, нотифікацій та інших особистих параметрів.
registerMentor	Реєстрація нового ментора в системі з основною інформацією та професійним досвідом.
updateMentorProfile	Зміна професійної інформації ментора.
addMentorSpecializations	Додавання та редагування спеціалізацій та навичок ментора.
searchMentors	Пошук менторів за спеціалізаціями, досвідом, рейтингом та іншими параметрами.
viewMentorProfile	Перегляд детальної інформації про ментора.
selectMentor	Вибір ментора для початку співпраці.
textChat	Обмін текстовими повідомленнями між ментором і менті.

Продовження таблиці 3.3

Назва функції	Опис
voiceCall	Здійснення голосових дзвінків між ментором і менті.
videoCall	Здійснення відео дзвінків між ментором і менті.
createSession	Планування нової навчальної сесії.
trackGoals	Відстеження досягнення навчальних цілей.
rateSession	Оцінка проведених сесій і залишення відгуків.
uploadResources	Завантаження навчальних матеріалів, таких як відео, статті, електронні книги.
viewResources	Перегляд і використання доступних навчальних матеріалів.
jwtAuthentication	Захист автентифікації користувачів за допомогою JWT токенів.
dataProtection	Захист особистих даних користувачів.
contactSupport	Надсилення запитів до служби підтримки.
resolveIssues	Вирішення технічних проблем користувачів.
calendarIntegration	Синхронізація з календарями для планування сесій.
socialMediaIntegration	Інтеграція з соціальними мережами для зручного використання застосунку.

3.3 Реалізація клієнтської частини мобільного застосунку

На рисунку 3.1 показано вітальну сторінку мобільного застосунку для пошуку ментора. Верхня частина екрану містить зображення мальовничого гірського пейзажу, що створює приємну атмосферу для користувачів. Під зображенням розташований вітальний текст, який запрошує користувачів до застосунку і обіцяє допомогу у пошуку ментора, який допоможе досягти їхніх цілей. Нижче тексту розташовані дві кнопки: "Увійти" для тих, хто вже має обліковий запис, і "Зареєструватися" для нових користувачів.

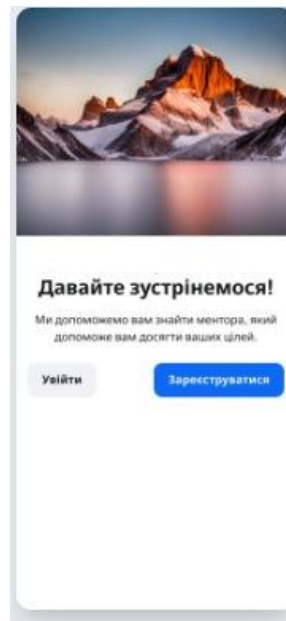


Рисунок 3.1 - Вітальна сторінка мобільного застосунку

На рисунку 3.2 показано сторінку пошуку ментора у мобільному застосунку. У верхній частині екрану розташований пошуковий рядок і фільтри для вибору спеціалізації менторів, такі як "Менеджмент" і "Продуктовий менеджмент". Нижче розташовані картки менторів із зазначенням їх імен, посад та досвіду роботи. Кожна картка включає фотографію ментора та короткий опис його кваліфікацій. У нижній частині екрану знаходиться навігаційна панель з іконками для швидкого доступу до головних розділів застосунку.

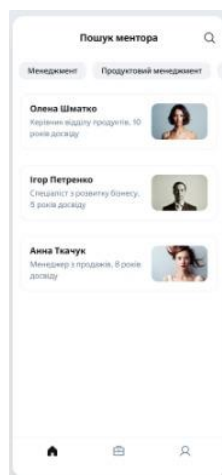


Рисунок 3.2 - Сторінка пошуку ментора

На рисунку 3.3 показано сторінку профілю ментора у мобільному застосунку. У верхній частині екрану розташований пошуковий рядок і фільтри для уточнення пошуку. На сторінці відображається детальна інформація про ментора Ірину Кузьмічову, включаючи її фотографію, посаду (Senior Product Manager at Google) та загальний рейтинг на основі 200 відгуків. Рейтинг представлений у вигляді зірок та діаграми, яка показує розподіл оцінок. Нижче розташовані дві кнопки: "Записатися на зустріч" для планування сесії з ментором та "Написати" для відправлення повідомлення.

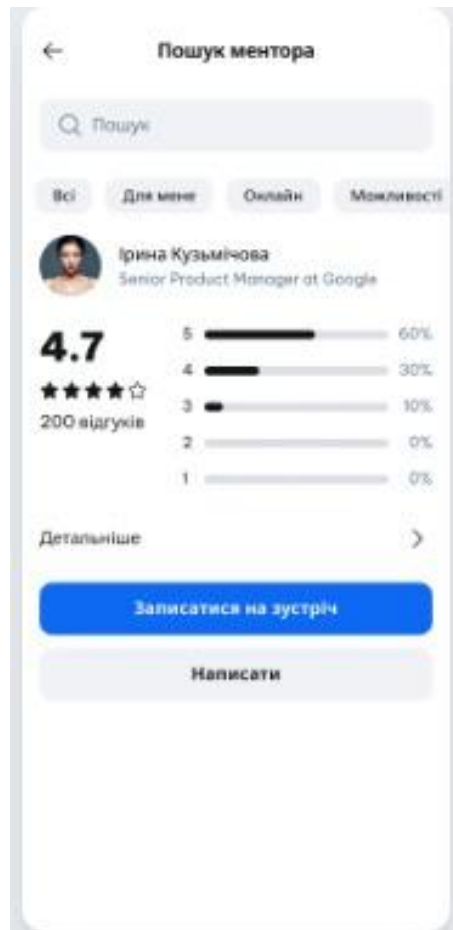


Рисунок 3.3 - Сторінка профілю ментора

На рисунку 3.4 показано деталі профілю ментора у мобільному застосунку. Сторінка містить фотографію ментора Олени Марчук, її посаду (Ментор, Інтернет-маркетинг) та короткий опис її професійного досвіду.

Вказано, що Олена працює в маркетингу вже 5 років і спеціалізується на продажах та стратегії в інтернеті. Нижче розташований рейтинг ментора (4.8 зірок на основі 120 відгуків), який представлений у вигляді зірок та діаграми розподілу оцінок. Доступні дві кнопки: "Записатися на консультацію" для планування сесії з ментором та "Написати повідомлення" для відправлення повідомлення.

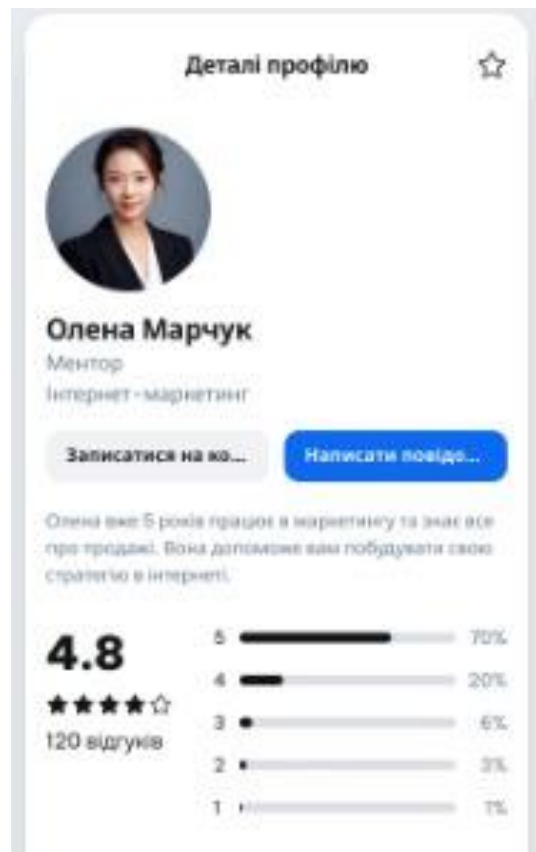


Рисунок 3.4 - Деталі профілю ментора

На рисунку 3.5 показано детальну сторінку ментора в мобільному застосунку, присвячену Людмилі Петренко, викладачу англійської мови з 15-річним досвідом. Сторінка включає її фотографію, ім'я та професійний досвід у верхній частині. Нижче розташовано "План навчання", який містить перелік запланованих тем та курсів, таких як "Тест з понеділка по літературі", "Курс за темою 'Граматика'", і "Тест з п'ятниці по понеділках". Ще нижче знаходиться розділ "Звіт про прогрес", який містить загальну

оцінку менті (8.5 з 10) та кнопку для відправлення відгуків. Ця сторінка дозволяє користувачам легко відстежувати їхній навчальний прогрес та планувати наступні кроки у навчанні.

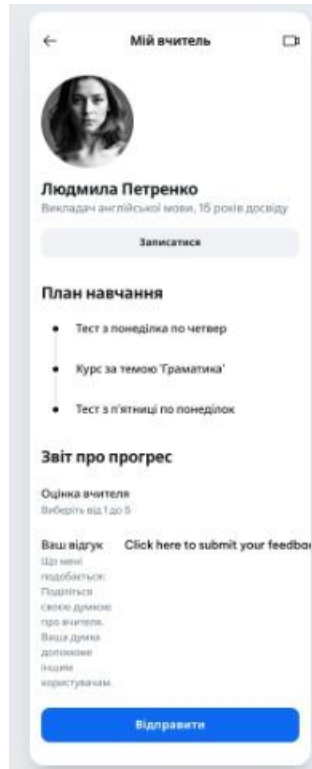


Рисунок 3.5 - Сторінка ментора з планом навчання та звітом про прогрес

На рисунку 3.6 представлено інтерфейс фільтрів пошуку менторів у мобільному застосунку. Ця сторінка дозволяє користувачам детально налаштувати параметри пошуку за допомогою кількох категорій: "Спеціалізація" з можливістю вибору між програмуванням, маркетингом та іншими; "Регіон" та "Місто" для вибору географічного розташування ментора; "Менеджмент" і "Фінанси" для уточнення сфери експертизи. Користувачі також можуть вибрати менторів згідно з їх рейтингом, наприклад, від 3.5 до 4.0 зірок, та професійним досвідом, наприклад, від 2 до 25+ років. У нижній частині сторінки розташована кнопка "Застосувати фільтри", яка дозволяє застосувати вибрані параметри пошуку.

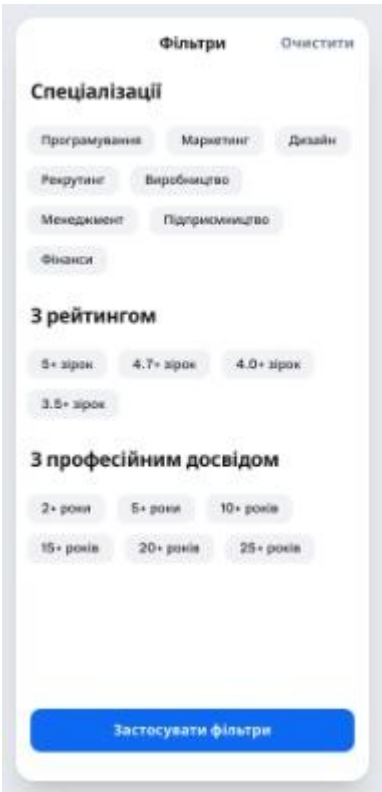


Рисунок 3.6 - Сторінка фільтрів для пошуку ментора

На рисунку 3.7 відображено віджет "Спілкування" у мобільному застосунку, який демонструє інтерфейс користувача для обміну повідомленнями. Цей інтерфейс дозволяє користувачам легко взаємодіяти та отримувати консультації або обговорювати питання, пов'язані з їхніми професійними інтересами.

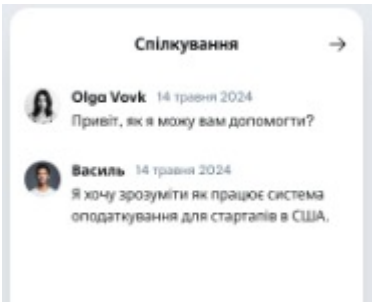


Рисунок 3.7 - Віджет спілкування в мобільному застосунку

На рисунку 3.8 показано сторінку налаштувань у мобільному застосунку, яка дозволяє користувачам керувати своїм профілем та

взаємодіями зі застосунком. Сторінка містить кілька розділів для налаштувань: "Особиста інформація", де користувачі можуть оновити свої ім'я, прізвище та стать; "Для запрошення", що містить опції налаштування професійних зв'язків та запрошення; "Професійний досвід та освіта", де можна додати або оновити інформацію про освіту та робочий досвід; розділи для управління списками команда та колеги, а також налаштування сповіщень. Користувачі також можуть вносити зміни до "Теми, що цікавлять" та керувати підписками на сповіщення, що допомагає їм краще організувати отримання інформації.

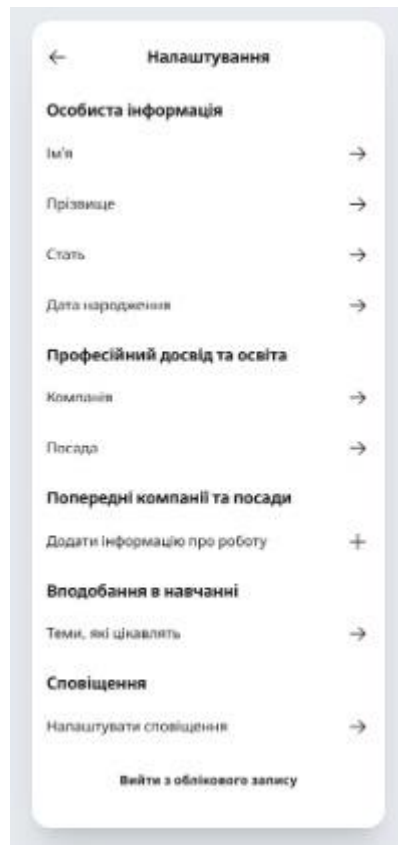


Рисунок 3.8 - Сторінка налаштувань у мобільному застосунку

На рисунку 3.9 відображено розділ "Навчальні ресурси" мобільного застосунку, де користувачі можуть знайти різноманітні курси та матеріали для саморозвитку. Ця сторінка показує кілька курсів, включаючи "Як вести переговори", "Як стати кращим менеджером" та "Курс по маркетингу", кожен

з яких супроводжується ілюстрацією та міткою "відео", що вказує на формат матеріалу. У верхній частині сторінки розміщено дропдаун меню для вибору дати, що дозволяє користувачам переглядати навчальні матеріали, опубліковані в певний день. Цей інтерфейс сприяє легкому доступу до навчальних ресурсів, спрямованих на професійний розвиток та підвищення кваліфікації.

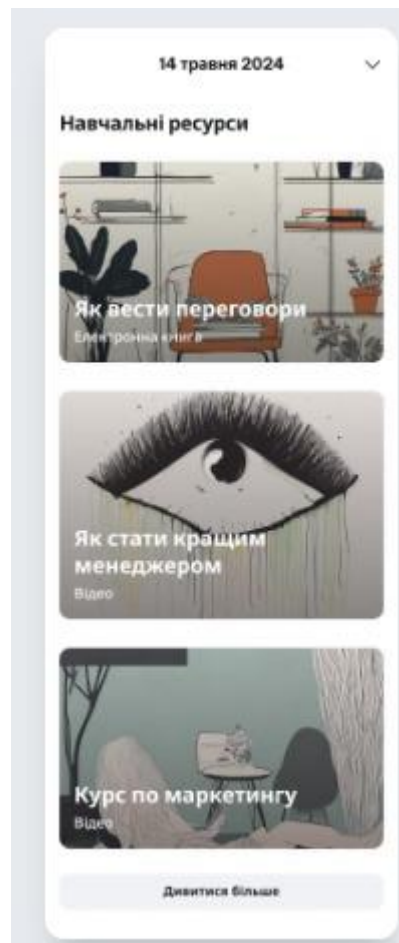


Рисунок 3.9 - Розділ навчальних ресурсів у мобільному застосунку

На рисунку 3.10 представлений інтерфейс для планування зустрічей з ментором у мобільному застосунку. Сторінка надає користувачам можливість вибрати зручну дату та час для сесії. Користувачі можуть вибрати з можливостей, таких як "Сьогодні", "Завтра", "Цього тижня", та "Власне" для введення специфічного часу. Додатково, на сторінці є опції для налаштування повторюваних нагадувань про зустрічі, такі як "Нагадати за 2

години", "Нагадати за 1 день" і "Нагадати за 1 тиждень", що дозволяє користувачам налаштувати нагадування відповідно до їхніх переваг. Цей інструмент спрощує процес координації зустрічей і допомагає утримувати навчальний графік організованим.

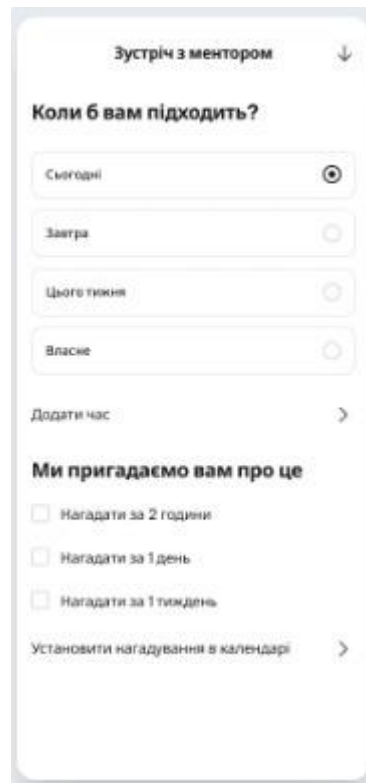


Рисунок 3.10 - Інтерфейс планування зустрічей з ментором

На рисунку 3.11 представлено інтерфейс для вибору конкретної дати та часу зустрічі в мобільному застосунку. Інтерфейс включає календар, який дозволяє користувачу обрати дату у зручному місячному форматі, де виділені доступні дні для зустрічі. Нижче календаря розташовано меню для вибору часу, з опціями такими як "07:00 AM", "08:00 PM", "09:00 PM", із вказівкою на сьогодні та завтра для додаткової зручності планування. Кнопка "Зберегти" внизу екрану дозволяє підтвердити вибір користувача, що робить цей інструмент важливим для ефективного управління часом і забезпечення легкої координації між ментором та учнем.

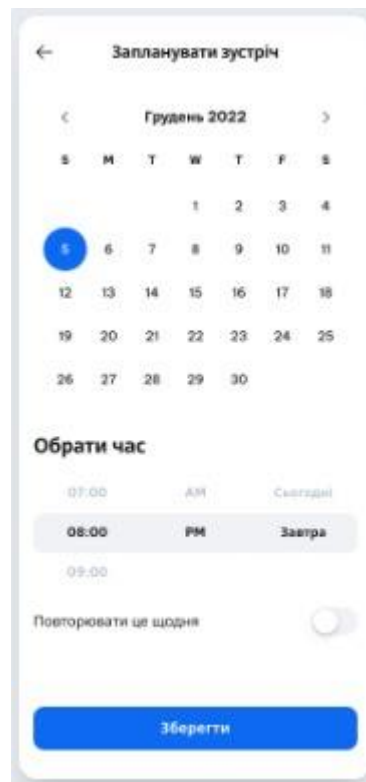


Рисунок 3.11 - Інтерфейс вибору часу для зустрічей у мобільному застосунку

3.4 Висновки за розділом

У третьому розділі було детально розглянуто процес вибору технологій та інструментів для розробки мобільного застосунку пошуку ментора, а також описано реалізацію його серверної та клієнтської частин.

При виборі технологій для розробки мобільного застосунку були розглянуті різні підходи, такі як нативна розробка, крос-платформна розробка та розробка прогресивних веб-застосунків (PWA). Для даного проєкту було обрано нативну розробку для платформи Android з використанням мов програмування Java та Kotlin. Цей вибір зумовлений перевагами нативної розробки, такими як висока продуктивність, повний доступ до можливостей платформи, широка підтримка пристроїв, зрілість та стабільність платформи, а також можливість використання сучасної мови Kotlin.

Для реалізації серверної частини мобільного застосунку було використано мову програмування Kotlin разом із фреймворком Ktor. Було

описано основні функції застосунку, такі як реєстрація та автентифікація користувачів, пошук та вибір менторів, комунікація між ментором і менті, планування та відстеження навчальних сесій, доступ до навчальних ресурсів, захист даних користувачів, підтримка користувачів та інтеграція з іншими сервісами.

Реалізація клієнтської частини мобільного застосунку була продемонстрована за допомогою серії скріншотів, які відображають основні сторінки та функціональність застосунку. Були представлені вітальна сторінка, сторінка пошуку ментора, сторінка профілю ментора, деталі профілю ментора, а також інші ключові елементи інтерфейсу користувача.

4 РЕКОМЕНДАЦІЇ ІЗ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПОШУКУ МЕНТОРА

4.1 Тестування та налагодження мобільного застосунку

Тестування та налагодження були важливими етапами в процесі розробки мобільного застосунку для пошуку ментора. Ці етапи допомогли забезпечити коректну роботу застосунку, виявити та усунути потенційні помилки, а також оптимізувати продуктивність і зручність використання.

У процесі тестування було використано кілька різних типів тестів. Модульне тестування (Unit testing) дозволило перевірити правильність роботи окремих компонентів застосунку, таких як функції та класи. Інтеграційне тестування (Integration testing) було зосереджено на взаємодії між різними модулями застосунку та перевірці їх сумісності. Системне тестування (System testing) охопило застосунок у цілому і перевірило його відповідність функціональним та нефункціональним вимогам.

Окрім цього, було проведено тестування зручності використання (Usability testing), щоб оцінити, наскільки інтуїтивно зрозумілим та простим у навігації є інтерфейс користувача. Тестування на різних пристроях (Device testing) допомогло переконатися, що застосунок коректно працює на широкому спектрі Android-пристроїв з різними розмірами екранів та версіями операційної системи.

Для автоматизації процесу тестування було використано спеціалізовані інструменти та фреймворки, такі як JUnit, Espresso та Appium. Ці інструменти дозволили створювати та запускати тестові сценарії, що імітують дії користувача та перевіряють очікувану поведінку застосунку.

Налагодження застосунку передбачало виявлення та виправлення помилок, які були знайдені під час тестування або повідомлені користувачами. Інтегроване середовище розробки (IDE) Android Studio надало потужні інструменти для налагодження, такі як точки зупинки

(breakpoints), покрокове виконання коду та перегляд значень змінних під час виконання.

Для ефективного налагодження було використано журнали (logs) та інструменти моніторингу продуктивності. Журнали дозволили відстежувати хід виконання програми та виявляти проблемні місця. Інструменти моніторингу, такі як Android Profiler, допомогли аналізувати використання ресурсів застосунку, таких як ЦП, пам'ять та мережеві з'єднання, що дозволило оптимізувати продуктивність.

Після внесення змін до коду застосунку в процесі налагодження, було проведено регресійне тестування (Regression testing), щоб переконатися, що виправлення помилок не призвело до виникнення нових проблем в інших частинах застосунку.

Тестування та налагодження були ітеративним процесом, який тривав протягом усього життєвого циклу розробки мобільного застосунку. Регулярне проведення тестів та оперативне виправлення виявлених проблем допомогло підтримувати високу якість застосунку та забезпечило позитивний досвід користувачів.

4.2 Розробка інструкцій для користувачів мобільного застосунку

Для забезпечення ефективного використання мобільного застосунку для пошуку ментора було розроблено докладні інструкції для користувачів. Ці інструкції були створені з метою допомогти користувачам швидко освоїти функціональність застосунку та отримати максимальну користь від його використання.

Інструкції для користувачів були представлені у вигляді комплексного посібника, який охоплює всі ключові аспекти застосунку (додаток Б). Посібник було структуровано за основними розділами, такими як реєстрація та вхід до системи, налаштування профілю, пошук ментора, комунікація з

ментором, планування та відстеження навчальних сесій, доступ до навчальних ресурсів та налаштування конфіденційності.

Кожен розділ інструкцій було наповнено покроковими вказівками, які детально описують дії, необхідні для виконання конкретних завдань.

Окрім покрокових інструкцій, було включено поради та рекомендації щодо ефективного використання застосунку. Наприклад, було запропоновано поради щодо оптимізації профілю користувача для привернення уваги менторів, налаштування фільтрів пошуку для знаходження найбільш відповідних менторів та використання різних методів комунікації для продуктивної взаємодії з ментором.

Для зручності користувачів інструкції було зроблено доступними безпосередньо в застосунку через розділ "Довідка" або "FAQ". Це дозволяє користувачам швидко отримати необхідну інформацію без необхідності залишати застосунок або шукати інструкції на зовнішніх ресурсах.

Крім того, було створено окремий розділ для вирішення поширених проблем та відповідей на часті запитання. Цей розділ містить інформацію про типові проблеми, з якими можуть зіткнутися користувачі, та надає рекомендації щодо їх вирішення. Наявність такого розділу допомагає користувачам самостійно впоратися з більшістю проблем і зменшує навантаження на службу підтримки.

Розробка докладних та зручних інструкцій для користувачів є важливим аспектом впровадження мобільного застосунку для пошуку ментора. Чіткі та зрозумілі інструкції допомагають користувачам швидко освоїти функціональність застосунку, отримати максимальну користь від його використання та мінімізувати потребу в додатковій підтримці.

4.3 Висновки за розділом

У четвертому розділі було розглянуто важливі аспекти впровадження програмного забезпечення пошуку ментора, а саме тестування та

налагодження мобільного застосунку, а також розробку інструкцій для користувачів.

Процес тестування та налагодження відіграв ключову роль у забезпеченні коректної роботи застосунку, виявленні та усуненні потенційних помилок, оптимізації продуктивності та зручності використання. Було застосовано різні типи тестування, такі як модульне, інтеграційне, системне, тестування зручності використання та тестування на різних пристроях. Використання спеціалізованих інструментів та фреймворків дозволило автоматизувати процес тестування та ефективно імітувати дії користувача. Налагодження застосунку за допомогою вбудованих інструментів Android Studio, журналів та інструментів моніторингу продуктивності допомогло виявити та виправити проблеми. Тестування та налагодження були ітеративним процесом протягом усього життєвого циклу розробки, що забезпечило високу якість застосунку та позитивний досвід користувачів.

Для ефективного використання мобільного застосунку пошуку ментора було розроблено детальні інструкції для користувачів у вигляді комплексного посібника. Інструкції охоплюють всі ключові аспекти застосунку і містять покрокові вказівки для виконання конкретних завдань, а також поради щодо ефективного використання функцій. Доступність інструкцій безпосередньо в застосунку та на різних платформах забезпечує зручність для користувачів. Окремий розділ для вирішення поширених проблем та відповідей на часті запитання допомагає користувачам самостійно вирішувати більшість питань, зменшуючи навантаження на службу підтримки.

ВИСНОВКИ

В бакалаврській роботі проведено комплексне дослідження та розробку програмного забезпечення для пошуку ментора у вигляді мобільного застосунку.

У першому розділі було проаналізовано предметну область пошуку ментора, розглянуто існуючі мобільні застосунки та визначено потреби користувачів. На основі цього аналізу були сформовані конкретні вимоги до функціональності та інтерфейсу мобільного застосунку пошуку ментора.

У другому розділі було спроектовано архітектуру мобільного застосунку на основі клієнт-серверної моделі та структуру бази даних з використанням реляційної СКБД PostgreSQL. Обрані рішення забезпечують високу продуктивність, масштабованість, безпеку та ефективне керування даними.

У третьому розділі було детально описано процес вибору технологій та інструментів для розробки мобільного застосунку, а саме нативну розробку для платформи Android з використанням мов програмування Java та Kotlin. Також було представлено реалізацію серверної та клієнтської частин застосунку з описом основних функцій та елементів інтерфейсу користувача.

У четвертому розділі було розглянуто важливі аспекти впровадження програмного забезпечення, такі як тестування, налагодження та розробка інструкцій для користувачів. Ці заходи забезпечують коректну роботу застосунку, оптимізацію продуктивності та зручність використання, а також полегшують освоєння функціональності для кінцевих користувачів.

У результаті проведеної роботи було розроблено мобільний застосунок для пошуку ментора, який відповідає вимогам та потребам цільової аудиторії. Застосунок надає користувачам зручні інструменти для пошуку менторів, комунікації, планування та відстеження навчальних сесій, доступу до навчальних ресурсів та інших функцій, необхідних для ефективного менторства та професійного розвитку.

Розроблений мобільний застосунок має потенціал для широкого впровадження та використання, оскільки він вирішує актуальну проблему пошуку ментора та сприяє розвитку професійних навичок та особистісному зростанню користувачів. Подальші напрямки розвитку застосунку можуть включати розширення функціональності, інтеграцію з іншими платформами та сервісами, а також адаптацію до потреб специфічних галузей або професійних спільнот.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Воронкова В.Г. Веб-сайти як засіб масової інформації / В.Г. Воронкова. – Харків: ХНУ імені В.Н. Каразіна, 2012. – 186 с.
2. Лебедєв О.Л. Основи веб-дизайну: навчальний посібник / О.Л. Лебедєв. – Київ: Видавництво "Логос", 2014. – 324 с.
3. Mentor App by The Soultalk [Electronic resource] - Access mode: <https://play.google.com/store/apps/details?id=co.diy.ubdfe&hl=uk>
4. Mentora [Electronic resource] - Access mode: <https://play.google.com/store/apps/details?id=com.personalleadership.dailymentor&hl=uk&gl=US>
5. PushFar - The Mentoring App [Electronic resource] - Access mode: <https://play.google.com/store/apps/details?id=com.pushfar.app&hl=uk&gl=US>
6. Морозова О.В., Петренко А.С. Стратегії розвитку веб-ресурсів в умовах глобалізації: монографія / О.В. Морозова, А.С. Петренко. – Житомир: Видавництво ЖДУ імені І. Франка, 2016. – 148 с.
7. Пригода П.П. Розробка та оптимізація веб-сайтів / П.П. Пригода. – Київ: Наукова думка, 2018. – 192 с.
8. Чернишенко О.В. Адаптивний веб-дизайн: від простого до складного / О.В. Чернишенко. – Одеса: Астропринт, 2015. – 167 с.
9. Яковлев В.Ф. Системи управління вмістом сайту: аналіз та перспективи / В.Ф. Яковлев. – Дніпро: Видавництво ДНУ, 2019. – 210 с.
10. Семенов А.Ю., Лисенко Д.Д. Проектування інформаційних систем / А.Ю. Семенов, Д.Д. Лисенко. – Львів: Видавництво НУ "Львівська політехніка", 2020. – 134 с.
11. Ковальчук О.В. Інтернет-маркетинг: Підручник / О.В. Ковальчук. – Харків: Фоліо, 2021. – 310 с.
12. Гусєв К.Д. Веб-технології та електронний бізнес / К.Д. Гусєв. – Київ: Академія, 2017. – 295 с.

13. Бажанова Ю.О., Петренко М.П. Веб-дизайн: нові тенденції та технології / Ю.О. Бажанов, М.П. Петренко. – Львів: Видавництво "Новий Світ-2000", 2019. – 143 с.
14. Єршова О.Ю. Методи оцінювання якості веб-сайтів / О.Ю. Єршова. – Донецьк: ДонНУ, 2018. – 190 с.
15. Федоров В.А. Мобільні додатки: розробка та продвижение / В.А. Федоров. – Київ: Видавничий дім "Діло", 2022. – 256 с.
16. Міщенко А.В. Основи SEO: Повний курс / А.В. Міщенко. – Київ: Видавничий дім "Діло", 2022. – 274 с.

ДОДАТОК А
Текст програми

```

CREATE TABLE Users (
  id SERIAL PRIMARY KEY,
  username VARCHAR(50) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  profile_picture VARCHAR(255),
  role VARCHAR(20) NOT NULL CHECK (role IN ('mentor', 'mentee')),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```

CREATE TABLE Mentors (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES Users(id),
  bio TEXT,
  rating NUMERIC CHECK (rating >= 0 AND rating <= 5),
  specializations VARCHAR[]
);

```

```

CREATE TABLE MenteePreferences (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES Users(id),
  preferences JSONB
);

```

```

CREATE TABLE Sessions (
  id SERIAL PRIMARY KEY,
  mentor_id INT REFERENCES Mentors(id),
  mentee_id INT REFERENCES Users(id),
  scheduled_time TIMESTAMP NOT NULL,
  status VARCHAR(20) NOT NULL CHECK (status IN ('planned',
'completed', 'cancelled')),
  feedback TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```

CREATE TABLE Messages (
  id SERIAL PRIMARY KEY,
  session_id INT REFERENCES Sessions(id),
  sender_id INT REFERENCES Users(id),
  content TEXT NOT NULL,
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```
CREATE TABLE Resources (
    id SERIAL PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    description TEXT,
    type VARCHAR(50) NOT NULL CHECK (type IN ('video', 'article',
'ebook', 'course')),
    url VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
INSERT INTO Users (username, password_hash, email, first_name,
last_name, role)
VALUES ('johndoe', 'hashedpassword', 'john@example.com', 'John', 'Doe',
'mentor');
```

```
INSERT INTO Mentors (user_id, bio, rating, specializations)
VALUES (1, 'Experienced software developer', 4.5, ARRAY['Software
Development', 'Career Advice']);
```

```
INSERT INTO MenteePreferences (user_id, preferences)
VALUES (2, '{"specializations": ["Software Development"], "language":
"English"}');
```

```
INSERT INTO Sessions (mentor_id, mentee_id, scheduled_time, status)
VALUES (1, 2, '2024-07-01 10:00:00', 'planned');
```

```
INSERT INTO Messages (session_id, sender_id, content)
VALUES (1, 1, 'Looking forward to our session!');
```

```
INSERT INTO Resources (title, description, type, url)
VALUES ('Intro to Python', 'A beginner\'s guide to Python programming',
'video', 'http://example.com/python-intro');
```

```
import kotlinx.serialization.Serializable
```

```
@Serializable
```

```
data class User(
    val id: Int? = null,
    val username: String,
    val passwordHash: String,
    val email: String,
    val firstName: String?,
    val lastName: String?,
    val profilePicture: String?,
```

```

        val createdAt: String? = null,
        val updatedAt: String? = null
    )

    object Users : Table() {
        val id = integer("id").autoIncrement().primaryKey()
        val username = varchar("username", 255).uniqueIndex()
        val passwordHash = varchar("password_hash", 255)
        val email = varchar("email", 255).uniqueIndex()
        val firstName = varchar("first_name", 255).nullable()
        val lastName = varchar("last_name", 255).nullable()
        val profilePicture = varchar("profile_picture", 255).nullable()
        val createdAt = datetime("created_at").defaultExpression(CurrentDateTime())
        val updatedAt = datetime("updated_at").defaultExpression(CurrentDateTime())
    }

    fun Application.registerUserModule() {
        routing {
            post("/register") {
                val newUser = call.receive<User>()

                transaction {
                    addLogger(StdOutSqlLogger)
                    SchemaUtils.create(Users)

                    val userExists = Users.select { Users.username eq
newUser.username or (Users.email eq newUser.email) }
                        .count() > 0

                    if (userExists) {
                        call.respond(HttpStatusCode.Conflict, "User already exists")
                        return@transaction
                    }

                    val userId = Users.insertAndGetId {
                        it[username] = newUser.username
                        it[passwordHash] = newUser.passwordHash
                        it[email] = newUser.email
                        it[firstName] = newUser.firstName
                        it[lastName] = newUser.last
                    }
                }
            }
        }
    }

    fun main() {
        embeddedServer(Netty, port = 8080) {

```

```

install(ContentNegotiation) {
    json()
}
install(CallLogging)
install(Authentication) {
    jwt("auth-jwt") {
        realm = "ktor.io"
        verifier(
            JWT
                .require(Algorithm.HMAC256("secret"))
                .withAudience("ktor-audience")
                .withIssuer("ktor-issuer")
                .build()
        )
        validate { credential ->
            if (credential.payload.audience.contains("ktor-audience"))
                JWTPrincipal(credential.payload) else null
        }
    }
}
routing {
    registerUserModule()
    authenticateUserModule()
}
}.start(wait = true)
}

```

```

fun Application.authenticateUserModule() {
    routing {
        post("/login") {
            val credentials = call.receive<UserCredentials>()
            val user = transaction {
                Users.select {
                    (Users.username eq credentials.username) or
                    (Users.email eq credentials.username)
                }.singleOrNull()?.let {
                    User(
                        id = it[Users.id],
                        username = it[Users.username],
                        passwordHash = it[Users.passwordHash],
                        email = it[Users.email],
                        firstName = it[Users.firstName],
                        lastName = it[Users.lastName],
                        profilePicture = it[Users.profilePicture],

```

```

        createdAt = it[Users.createdAt].toString(),
        updatedAt = it[Users.updatedAt].toString()
    )
}
}

if (user == null || !BCrypt.checkpw(credentials.password,
user.passwordHash)) {
    call.respond(HttpStatusCode.Unauthorized, "Invalid credentials")
    return@post
}

val token = JWT.create()
    .withAudience("ktor-audience")
    .withIssuer("ktor-issuer")
    .withClaim("username", user.username)
    .withExpiresAt(Date(System.currentTimeMillis() + 600000))
    .sign(Algorithm.HMAC256("secret"))

call.respond(mapOf("token" to token))
}
}
}

data class UserCredentials(val username: String, val password: String)

fun Application.updateUserProfileModule() {
    routing {
        authenticate("auth-jwt") {
            put("/profile") {
                val principal = call.principal<JWTPrincipal>()
                val username = principal!!.payload.getClaim("username").asString()
                val updatedUser = call.receive<User>()

                transaction {
                    val user = Users.select { Users.username eq username
                }.singleOrNull()

                    if (user == null) {
                        call.respond(HttpStatusCode.NotFound, "User not found")
                        return@transaction
                    }
                }
            }
        }
    }
}

```

```

        Users.update({ Users.username eq username }) {
            it[email] = updatedUser.email
            it[firstName] = updatedUser.firstName
            it[lastName] = updatedUser.lastName
            it[profilePicture] = updatedUser.profilePicture
            it[updatedAt] = org.jetbrains.exposed.sql.CurrentDateTime()
        }

        val userUpdated = Users.select { Users.username eq username
    }.single()

    call.respond(
        User(
            id = userUpdated[Users.id],
            username = userUpdated[Users.username],
            passwordHash = userUpdated[Users.passwordHash],
            email = userUpdated[Users.email],
            firstName = userUpdated[Users.firstName],
            lastName = userUpdated[Users.lastName],
            profilePicture = userUpdated[Users.profilePicture],
            createdAt = userUpdated[Users.createdAt].toString(),
            updatedAt = userUpdated[Users.updatedAt].toString()
        )
    )
}

object MenteePreferencesTable : Table() {
    val id = integer("id").autoIncrement().primaryKey()
    val userId = integer("user_id").references(Users.id)
    val preferences = jsonb("preferences")
}

fun Application.setUserPreferencesModule() {
    routing {
        authenticate("auth-jwt") {
            post("/preferences") {
                val principal = call.principal<JWTPrincipal>()
                val username = principal!!.payload.getClaim("username").asString()
            }
        }
    }
}

```

```

val preferences = call.receive<MenteePreferences>()

transaction {
  addLogger(StdOutSqlLogger)
  SchemaUtils.create(MenteePreferencesTable)

  val user = Users.select { Users.username eq username
}.singleOrNull()
  if (user == null) {
    call.respond(HttpStatusCode.NotFound, "User not found")
    return@transaction
  }

  val userId = user[Users.id]
  val existingPreferences = MenteePreferencesTable.select {
MenteePreferencesTable.userId eq userId }.singleOrNull()

  if (existingPreferences == null) {
    MenteePreferencesTable.insert {
      it[MenteePreferencesTable.userId] = userId
      it[MenteePreferencesTable.preferences] =
preferences.preferences
    }
  } else {
    MenteePreferencesTable.update({
MenteePreferencesTable.userId eq userId }) {
      it[MenteePreferencesTable.preferences] =
preferences.preferences
    }
  }

  call.respond(HttpStatusCode.OK, "Preferences updated")
}

get("/preferences") {
  val principal = call.principal<JWTPrincipal>()
  val username = principal!!.payload.getClaim("username").asString()

  transaction {
    val user = Users.select { Users.username eq username
}.singleOrNull()
    if (user == null) {
      call.respond(HttpStatusCode.NotFound, "User not found")

```

```

        return@transaction
    }

    val userId = user[Users.id]
    val preferences = MenteePreferencesTable.select {
MenteePreferencesTable.userId eq userId }.singleOrNull()

    if (preferences == null) {
        call.respond(HttpStatusCode.NotFound, "Preferences not
found")
    } else {
        call.respond(
            MenteePreferences(
                id = preferences[MenteePreferencesTable.id],
                userId = preferences[MenteePreferencesTable.userId],
                preferences = preferences[MenteePreferencesTable.preferences]
            )
        )
    }
}
}
}
}
}
```

```
object Mentors : Table() {
  val id = integer("id").autoIncrement().primaryKey()
  val userId = integer("user_id").references(Users.id)
  val bio = text("bio")
  val specializations = varchar("specializations", 255)
}
```

```
fun Application.registerMentorModule() {
    routing {
        authenticate("auth-jwt") {
            post("/mentor/register") {
                val principal = call.principal<JWTPrincipal>()
                val username = principal!!.payload.getClaim("username").asString()
                val mentorDetails = call.receive<Mentor>()

                transaction {
                    addLogger(StdOutSqlLogger)
                }
            }
        }
    }
}
```

```

SchemaUtils.create(Mentors)

val user = Users.select { Users.username eq username
}.singleOrNull()

if (user == null) {
    call.respond(HttpStatusCode.NotFound, "User not found")
    return@transaction
}

val userId = user[Users.id]

Mentors.insert {
    it[Mentors.userId] = userId
    it[bio] = mentorDetails.bio
    it[specializations] = mentorDetails.specializations
}

call.respond(HttpStatusCode.Created, "Mentor registered
successfully")
}
}
}
}
}
}
}

```

ДОДАТОК Б
Інструкції користувача

Розділ 1: Реєстрація та вхід до системи

Для початку роботи з мобільним застосунком для пошуку ментора, вам необхідно створити обліковий запис або увійти до системи, якщо ви вже маєте обліковий запис.

Реєстрація нового облікового запису:

- завантажте та встановіть мобільний застосунок для пошуку ментора з App Store або Google Play Store;
- відкрийте застосунок та натисніть кнопку "Реєстрація" на початковому екрані;
- введіть свою електронну адресу, ім'я та прізвище у відповідні поля;
- створіть надійний пароль, дотримуючись вимог щодо його складності (наприклад, мінімум 8 символів, включаючи великі та малі літери, цифри та спеціальні символи);
- підтвердіть пароль, ввівши його ще раз у відповідне поле;
- ознайомтеся з Умовами використання та Політикою конфіденційності, поставивши відмітку у відповідному чекбоксі;
- натисніть кнопку "Зареєструватися" для створення облікового запису;
- після успішної реєстрації, вам буде надіслано електронний лист з посиланням для підтвердження вашої електронної адреси. перейдіть за цим посиланням для активації облікового запису.

Вхід до системи з існуючим обліковим записом:

- на початковому екрані застосунку натисніть кнопку "Вхід";
- введіть свою електронну адресу та пароль у відповідні поля;
- натисніть кнопку "Увійти" для входу до системи;
- Якщо ви забули пароль:
- на екрані входу натисніть опцію "Забули пароль?";

- введіть свою електронну адресу, пов'язану з обліковим записом, і натисніть кнопку "Надіслати";
- вам буде надіслано електронний лист з інструкціями щодо скидання пароля. дотримуйтесь цих інструкцій для створення нового пароля;
 - Вхід через облікові записи соціальних мереж:
- на екрані входу натисніть кнопку соціальної мережі, через яку ви бажаєте увійти (наприклад, Facebook або Google);
- дотримуйтесь інструкцій для входу та надання дозволів застосунку на доступ до вашої інформації профілю;
- після успішної автентифікації, ваш обліковий запис соціальної мережі буде прив'язано до профілю в застосунку.

Після успішної реєстрації або входу до системи, ви отримаєте доступ до всіх функцій мобільного застосунку для пошуку ментора та зможете розпочати пошук ментора відповідно до ваших потреб та уподобань.

Розділ 2: Налаштування профілю

Після успішної реєстрації або входу до системи, важливо налаштувати свій профіль, щоб потенційні ментори могли отримати більше інформації про вас і ваші цілі. Ось покрокова інструкція для налаштування профілю:

- на головному екрані застосунку натисніть на зображення профілю у верхньому правому куті, щоб перейти до розділу "Профіль";
- натисніть кнопку "Редагувати профіль" для внесення змін;
- завантажте своє фото профілю, натиснувши на область зображення та вибравши фото з галереї вашого пристрою або зробивши нове фото за допомогою камери;
- введіть своє повне ім'я та прізвище у відповідні поля;
- у полі "Про мене" напишіть короткий опис себе, своїх інтересів, досвіду та цілей. це допоможе менторам краще зрозуміти, чим ви займаєтесь і що прагнете досягти;
- у розділі "Навички та досвід" додайте інформацію про свої професійні навички, освіту, досвід роботи та інші відповідні досягнення. для

додавання нової навички або досвіду натисніть кнопку "+" та заповніть необхідні поля;

- у розділі "Цілі та очікування" вкажіть свої основні цілі, які ви прагнете досягти за допомогою менторства. це може включати розвиток конкретних навичок, отримання порад щодо кар'єрного росту, або вирішення специфічних проблем;

- якщо ви волієте спілкуватися певною мовою з ментором, вкажіть її у відповідному полі;

- у розділі "Налаштування конфіденційності" виберіть, яка інформація з вашого профілю буде доступна для інших користувачів застосунку;

- після завершення налаштувань профілю натисніть кнопку "Зберегти".

Додаткові поради для створення ефективного профілю:

- використовуйте професійне та привабливе фото профілю, яке чітко показує ваше обличчя;

- будьте лаконічними та інформативними в описі себе та своїх цілей;

- виділіть свої найсильніші сторони, досягнення та унікальні характеристики, які можуть зацікавити потенційних менторів;

- регулярно оновлюйте свій профіль, додаючи нові навички, досвід та досягнення.

Пам'ятайте, що добре оформлений та інформативний профіль збільшує ваші шанси знайти відповідного ментора та побудувати продуктивні менторські відносини.

Розділ 3: Пошук ментора

Після налаштування профілю ви готові розпочати пошук ментора, який відповідає вашим потребам та цілям. Ось покрокова інструкція для ефективного пошуку ментора:

- на головному екрані застосунку натисніть на вкладку "Пошук";

- використовуйте рядок пошуку для введення ключових слів, пов'язаних з вашими інтересами, навичками або цілями. наприклад, "маркетинг", "розробка програмного забезпечення" або "дизайн";
- використовуйте фільтри пошуку для звуження результатів:
- спеціалізація: виберіть конкретну галузь або сферу діяльності ментора;
- досвід: вкажіть бажаний рівень досвіду ментора (наприклад, початківець, середній рівень або експерт);
- розташування: виберіть географічний регіон, якщо ви віддаєте перевагу ментору з певного місця;
- мова: вкажіть мову, якою ви хотіли б спілкуватися з ментором;
- натисніть кнопку "Застосувати фільтри" для перегляду відфільтрованих результатів пошуку;
- перегляньте список потенційних менторів, які відповідають вашим критеріям. кожен запис у списку міститиме фото ментора, ім'я, спеціалізацію та короткий опис;
- натисніть на профіль ментора, щоб переглянути детальну інформацію, включаючи повний опис, навички, досвід та відгуки інших менті;
- якщо ви знайшли ментора, який вас зацікавив, натисніть кнопку "Зв'язатися" на його профілі, щоб надіслати запит на менторство;
- у запиті вкажіть свої цілі, очікування та чому ви вважаєте, що цей ментор може вам допомогти. будьте ввічливими та професійними у своєму зверненні;
- якщо ментор приймає ваш запит, ви отримаєте сповіщення в застосунку та зможете розпочати спілкування;
- Додаткові поради для пошуку ментора:
- будьте відкритими до менторів з різних галузей та сфер діяльності, оскільки вони можуть запропонувати цінні знання та досвід;

- перегляньте відгуки та рейтинги менторів, щоб отримати уявлення про їхню ефективність та стиль роботи;
- не обмежуйтеся одним ментором - ви можете мати декількох менторів для різних аспектів вашого професійного розвитку;
- будьте готові до відмови або відсутності відповіді від деяких менторів. не сприймайте це особисто та продовжуйте пошук.

Пам'ятайте, що знаходження відповідного ментора може потребувати часу та зусиль, але результати того варті. Гарний ментор може допомогти вам розвинути навички, досягти цілей та досягти успіху у вашій кар'єрі.

Розділ 4: Комунікація з ментором

Після того, як ментор прийняв ваш запит, важливо налагодити ефективну комунікацію для побудови продуктивних менторських відносин. Ось покрокова інструкція для комунікації з ментором:

- на головному екрані застосунку перейдіть до вкладки "Повідомлення";
- знайдіть ментора, з яким ви хочете зв'язатися, у списку ваших контактів;
- натисніть на ім'я ментора, щоб відкрити вікно чату;
- розпочніть розмову, привітавшись та подякувавши ментору за прийняття вашого запиту;
- чітко поясніть свої цілі та очікування від менторства. будьте конкретними щодо того, які навички ви хочете розвинути або які проблеми прагнете вирішити;
- запитайте ментора про його досвід, думки та поради щодо ваших цілей;
- домовтеся про зручний для обох сторін графік та формат спілкування (наприклад, щотижневі відеодзвінки, обмін повідомленнями в застосунку або особисті зустрічі);
- будьте активним слухачем під час розмов з ментором. ставте уточнюючі запитання, робіть нотатки та показуйте свою зацікавленість;

- регулярно інформуйте ментора про свій прогрес, досягнення та виклики. це допоможе ментору надавати більш персоналізовані поради та підтримку;
- якщо вам потрібна додаткова допомога або ресурси, не соромтеся просити ментора про рекомендації або контакти;
- виявляйте вдячність за час та зусилля, які ментор витрачає на вашу підтримку. повідомляйте про свої успіхи та те, як поради ментора допомогли вам.

Додаткові можливості комунікації в застосунку:

- голосові та відеодзвінки: для більш глибокого обговорення ваших цілей, прогресу та викликів;
- обмін файлами: можливість надсилати документи, презентації або інші релевантні матеріали безпосередньо в чаті;
- нагадування про зустрічі: застосунок може автоматично надсилати нагадування про заплановані дзвінки або зустрічі з ментором;
- Поради для ефективної комунікації:
- будьте ввічливими, пунктуальними та надійними у спілкуванні з ментором;
- підготуйте питання або теми для обговорення перед зустрічами, щоб максимально використати час;
- будьте відкритими до зворотного зв'язку та конструктивної критики від ментора;
- повідомляйте ментора, якщо у вас виникають труднощі з виконанням завдань або досягненням цілей;
- після завершення формальної менторської програми підтримуйте зв'язок з ментором для подальшої підтримки та networking;

Ефективна комунікація з ментором є ключовою для побудови міцних та взаємовигідних менторських відносин. Застосовуючи ці кроки та поради, ви зможете максимально використати можливості, які надає менторство для вашого особистого та професійного розвитку.

Розділ 5: Планування та відстеження навчальних сесій

Для ефективного менторства важливо мати структурований підхід до планування та відстеження навчальних сесій. Мобільний застосунок для пошуку ментора надає зручні інструменти для організації та моніторингу вашого прогресу. Ось покрокова інструкція для планування та відстеження навчальних сесій:

- на головному екрані застосунку перейдіть до вкладки "Сесії";
- натисніть кнопку "+" для створення нової навчальної сесії;
- виберіть ментора зі списку ваших контактів, з яким ви хочете запланувати сесію;
- вкажіть дату, час та тривалість сесії. застосунок автоматично врахує часові пояси для зручності планування;
- додайте назву або короткий опис сесії, щоб визначити її основну мету або тему;
- за необхідності, додайте нотатки або питання, які ви хочете обговорити під час сесії;
- натисніть кнопку "Створити сесію" для збереження запланованої сесії;
- ментор отримає сповіщення про запит на сесію і зможе підтвердити, запропонувати інший час або відхилити запит;
- після підтвердження сесії ментором, вона з'явиться у вашому календарі в застосунку;
- напередодні сесії ви отримаєте нагадування, щоб підготуватися та не пропустити зустріч;
- під час сесії робіть нотатки в застосунку, щоб зафіксувати ключові моменти, поради та завдання;
- після завершення сесії позначте її як виконану та додайте свої враження, рейтинг або відгук;

- у розділі "Прогрес" ви можете відстежувати свої completed sessions, досягнуті цілі та отримувати візуальне представлення свого розвитку з часом.

Додаткові можливості для планування та відстеження сесій:

- інтеграція з календарем: синхронізуйте заплановані сесії з вашим календарем на пристрої для зручності планування та отримання нагадувань;
- встановлення цілей: встановлюйте конкретні, вимірювані цілі для кожної сесії та відстежуйте прогрес у їх досягненні;
- ресурси та завдання: прикріплюйте релевантні ресурси, статті або завдання до сесії для подальшого вивчення та практики;
- історія сесій: переглядайте минулі сесії, нотатки та відгуки, щоб відстежити свій прогрес та визначити області для подальшого розвитку.

Поради для ефективного планування та відстеження сесій:

- будьте проактивними у плануванні сесій та встановленні чітких цілей для кожної зустрічі;
- регулярно переглядайте свій прогрес та обговорюйте його з ментором для отримання зворотного зв'язку та коригування курсу за необхідності;
- використовуйте нотатки та враження після сесій, щоб рефлексувати над отриманими знаннями та визначати наступні кроки;
- святкуйте свої досягнення та діліться успіхами з ментором для підтримання мотивації та залученості.

Регулярне планування та відстеження навчальних сесій допоможе вам залишатися організованими, цілеспрямованими та відповідальними на шляху до досягнення ваших цілей за підтримки ментора.

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Розробка програмного забезпечення пошуку ментора

Виконав студент групи, КНТ-211сп
Керівник проєкту, к.т.н., доцент

СВІДЕНКО В.О.
СКРУПСЬКИЙ С.Ю.

Рисунок В.1 – Слайд 1

Мета, задачі роботи

Метою даної дипломної роботи є розробка мобільного застосунку для пошуку ментора, який дозволить користувачам знаходити досвідчених професіоналів у різних галузях, спілкуватися з ними та отримувати цінні знання і поради.

Об'єктом дослідження є процес пошуку ментора за допомогою мобільних технологій.

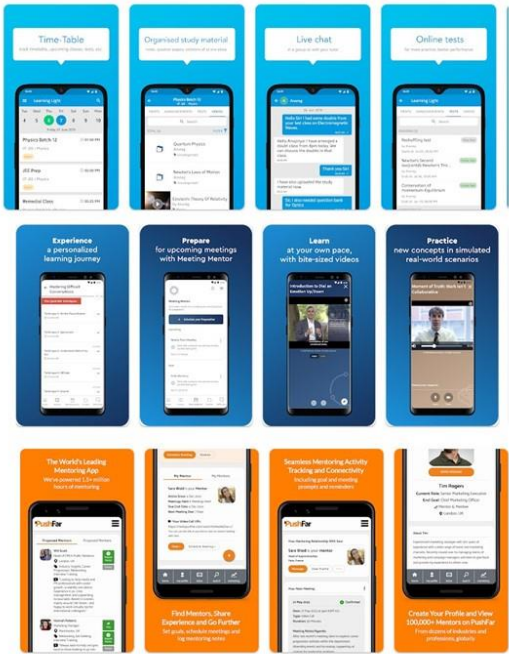
Предметом дослідження є методи та технології розробки мобільних застосунків для пошуку ментора.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- проаналізувати предметну область пошуку ментора та існуючі рішення;
- сформулювати вимоги до мобільного застосунку пошуку ментора;
- спроектувати архітектуру та базу даних мобільного застосунку;
- розробити серверну та клієнтську частини мобільного застосунку;
- провести тестування та налагодження мобільного застосунку;
- розробити інструкції для користувачів мобільного застосунку.

Рисунок В.2 – Слайд 2

Огляд існуючих мобільних застосунків для пошуку ментора



3

Рисунок В.3 – Слайд 3

Порівняння мобільних застосунків

Функціонал / Застосунок	Mentor	Mentora	PushFar
Персоналізований навчальний план	✓	✓	
Підготовка до зустрічей з ментором		✓	✓
Онлайн навчання через відео	✓	✓	
Інтерактивні симуляції для практики	✓	✓	
Відстеження менторських сесій			✓
Обмін досвідом із менторами		✓	✓
Профіль користувача і доступ до менторів	✓	✓	✓
Онлайн-чат з менторами			✓

4

Рисунок В.4 – Слайд 4

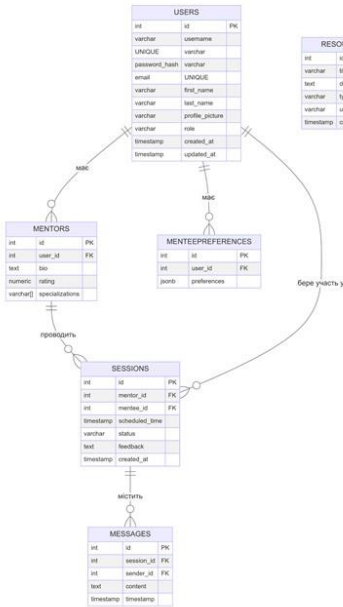
Аналіз потреб користувачів

Категорія	Опис
Легкість використання	Інтуїтивно зрозумілий інтерфейс, легкість навігації
Персоналізація	Індивідуалізація налаштувань профілю та пошуку
Доступність менторів	Широкий вибір менторів з різних галузей
Комунікація	Засоби для легкої комунікації між ментором та менті
Відстеження прогресу	Інструменти для оцінки та відстеження навчального прогресу
Навчальні ресурси	Доступ до різноманітних навчальних матеріалів
Гнучкість у виборі ментора	Можливість вибору менторів за різними параметрами
Безпека та конфіденційність	Заходи для захисту особистих даних та приватності комунікацій
Підтримка користувачів	Ефективна служба підтримки для вирішення запитів та проблем
Інтеграція з іншими сервісами	Сумісність з іншими платформами для забезпечення більшої зручності

5

Рисунок В.5 – Слайд 5

Схема бази даних



6

Рисунок В.6 – Слайд 6

Порівняння технологій розробки мобільних застосунків

Технологія	Переваги	Недоліки
Native (нативна)	Висока продуктивність; Повний доступ до можливостей платформи; Широка підтримка пристроїв; Зрілість та стабільність платформи	Високі витрати на розробку для кожної платформи окремо; Складність в підтримці кількох кодових баз
Cross-platform (крос-платформна)	Економія часу і ресурсів; Єдина кодова база для обох платформ	Можливі обмеження у продуктивності; Обмежений доступ до специфічних функцій платформи
Progressive Web Apps (PWA)	Легка розповсюджуваність; Працюють офлайн; Хороша продуктивність	Обмежений доступ до апаратних можливостей пристрою; Менша продуктивність у порівнянні з нативними застосунками

7

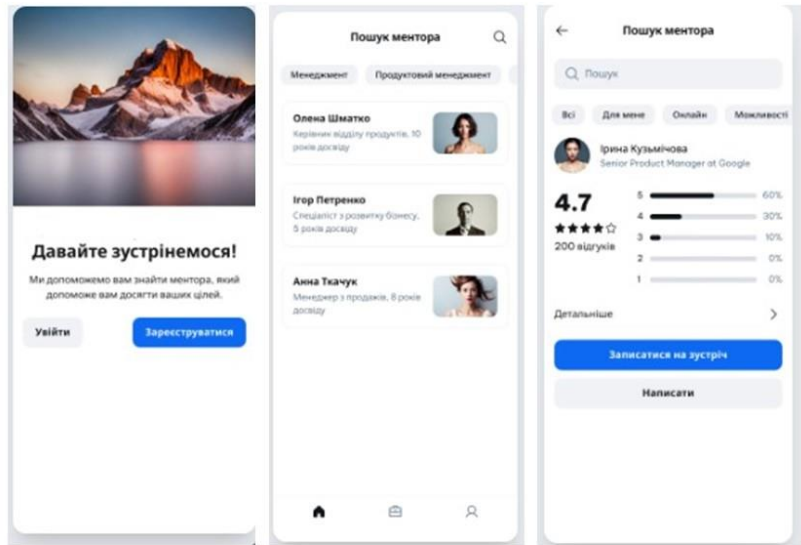
Рисунок В.7 – Слайд 7

Таблиця функцій мобільного застосунку

Назва функції	Опис
registerUser	Реєстрація нового користувача в системі з основною інформацією.
authenticateUser	Вхід користувача в систему за допомогою логіну та пароля.
updateUserProfile	Зміна особистої та професійної інформації користувача.
setUserPreferences	Налаштування фільтрів пошуку, нотифікацій та інших особистих параметрів.
registerMentor	Реєстрація нового ментора в системі з основною інформацією та професійним досвідом.
updateMentorProfile	Зміна професійної інформації ментора.
addMentorSpecializations	Додавання та редагування спеціалізацій та навичок ментора.
searchMentors	Пошук менторів за спеціалізаціями, досвідом, рейтингом та іншими параметрами.
viewMentorProfile	Перегляд детальної інформації про ментора.
selectMentor	Вибір ментора для початку співпраці.
textChat	Обмін текстовими повідомленнями між ментором і менті.
voiceCall	Здійснення голосових дзвінків між ментором і менті.
videoCall	Здійснення відео дзвінків між ментором і менті.
createSession	Планування нової навчальної сесії.
trackGoals	Відстеження досягнення навчальних цілей.
rateSession	Оцінка проведених сесій і залишення відгуків.
uploadResources	Завантаження навчальних матеріалів, таких як відео, статті, електронні книги.
viewResources	Перегляд і використання доступних навчальних матеріалів.
jwtAuthentication	Захист автентифікації користувачів за допомогою JWT токенів.
dataProtection	Захист особистих даних користувачів.
contactSupport	Надсилання запитів до служби підтримки.
resolveIssues	Вирішення технічних проблем користувачів.
calendarIntegration	Синхронізація з календарями для планування сесій.
socialMediaIntegration	Інтеграція з соціальними мережами для зручного використання застосунку.

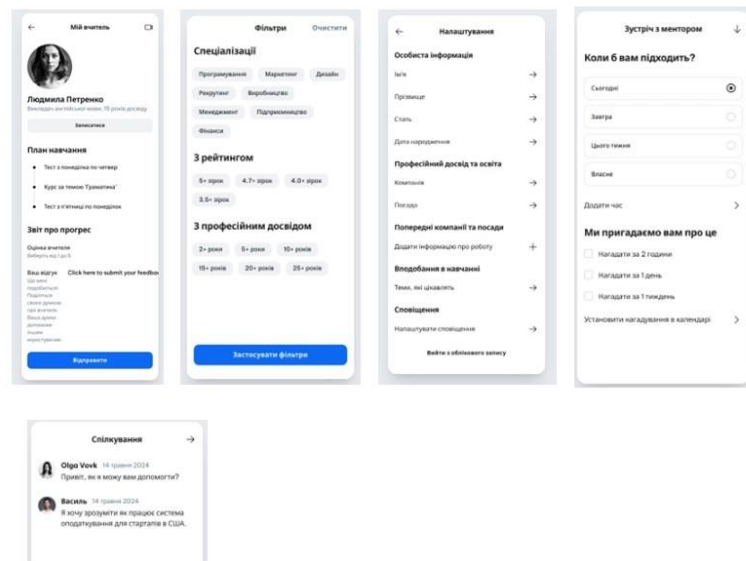
8

Рисунок В.8 – Слайд 8



9

Рисунок В.9 – Слайд 9



10

Рисунок В.10 – Слайд 10

Висновки

- Мета дипломної роботи, яка полягала в розробці мобільного застосунку для пошуку ментора, була успішно досягнута. Розроблений застосунок надає користувачам можливість знаходити досвідчених професіоналів у різних галузях, спілкуватися з ними та отримувати цінні знання і поради, що повністю відповідає поставленій меті.
- Для досягнення мети були успішно виконані всі поставлені задачі. Проведений аналіз предметної області пошуку ментора та існуючих рішень дозволив сформулювати чіткі вимоги до функціональності та інтерфейсу мобільного застосунку. На основі цих вимог була спроектована ефективна архітектура та структура бази даних, які забезпечують високу продуктивність, масштабованість та безпеку застосунку.
- Розробка серверної та клієнтської частин мобільного застосунку була успішно завершена з використанням сучасних технологій та інструментів, таких як Kotlin, Java, Ktor та PostgreSQL. Проведене тестування та налагодження застосунку гарантує його коректну роботу, оптимальну продуктивність та зручність використання.