

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

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Інформаційна система підтримки надання психологічної допомоги.

Підсистема підтримки роботи психолога _____

Здобувача групи _____ ІТ.м-23 _____ Шапки Сергія Олександровича _____
(шифр групи) (прізвище, ім'я, по батькові)

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

(підпис)

_____ Сергій Шапка _____
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник старший викладач кафедри ІТ, к.т.н. Ольга БОЙКО
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Суми – 2023

Сумський державний університет

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

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ С. М. Ващенко
«___» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентів

Шапка Сергій Олександрович
(прізвище, ім'я, по батькові)

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

Затверджена наказом по університету від «___» _____ 2023 р. № _____

2 Термін здачі студентом закінченого проекту « 11 » грудня _____ 2023 р.

3 Вхідні дані до проекту

.кабінет психолога, кабінет користувача, онлайн консультація, надання психологічної допомоги

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

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Презентація (_ слайдів)

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	5 днів	
2	Постановка задачі та методи дослідження	5 днів	
3	Моделювання системи надання психологічної допомоги	7 днів	
4	Реалізація інформаційної підсистеми психолога	15 днів	
5	Тестування та впровадження інформаційної системи	13 днів	

Магістрант _____

Шапка С.О.

Керівник роботи _____

к.т.н., Бойко О.В.

ABSTRACT

The topic of my master's thesis is «Information System to Support the Psychological Assistance Provision. Psychologist Work Support Subsystem».

The explanatory note consists of an introduction, 4 chapters, conclusions, 43 references, and 2 appendices. The total volume of the work is 70 pages, including 44 pages of the main text, 5 pages of the list of used sources, and 21 pages of appendices.

In an era characterized by rapid technological advancement, the integration of innovative solutions into healthcare systems has become imperative. This master's paper explores the conceptualization, design, and implementation of an information system dedicated to providing psychological support. Grounded in the principles of human-computer interaction and informed by psychological theories, the system aims to bridge the gap between traditional therapeutic approaches and modern technology.

The research begins by delving into the current landscape of mental health care, identifying existing challenges and opportunities for improvement. It emphasizes the increasing prevalence of mental health issues and the need for scalable, accessible, and personalized interventions. Building upon this foundation, the study undertakes an extensive review of relevant psychological theories, ensuring a solid theoretical framework for the development of the information system.

The main goal of this work is to create an information system for providing services from the psychologist's side. This system has a high level of security, ethics and practicality.

The study incorporates a meticulous comparative analysis of existing psychological support resources within the field, evaluating systems like the proposed information system. This comparative assessment serves as a foundational step, identifying key features, strengths, and weaknesses inherent in existing solutions. To enhance clarity and facilitate a comprehensive understanding of the project's distinct contributions, a differential table is meticulously crafted. This table highlights specific innovations and differentiators, offering a clear snapshot of how the developed

information system surpasses its counterparts. This systematic comparison not only informs the design and functionality of the proposed system but also accentuates its unique advantages within the landscape of technology-driven psychological support.

In the initial stages of system design, we employed the IDEF0 (Integration Definition for Function Modeling) methodology to develop a comprehensive diagram. This IDEF0 diagram served as a visual representation of the information system's functions and interactions, mapping out the flow of data and processes. By incorporating this modeling technique, we gained a holistic understanding of the system's architecture, enabling us to identify key functionalities and their interdependencies. The IDEF0 diagram played a crucial role in shaping the conceptual framework, providing a blueprint for the subsequent phases of development, and facilitating effective communication among the multidisciplinary team involved in the project.

Subsequently, for the implementation of a robust storage solution, we opted for a relational SQL (Structured Query Language) server. The choice of a relational database management system (RDBMS) was motivated by its ability to organize and manage data with a high level of integrity and consistency. Utilizing SQL for database management allowed for efficient and structured storage of diverse user data, ensuring easy retrieval and manipulation. The relational model, with its tables, keys, and relationships, provided a coherent framework for storing and managing complex datasets associated with psychological support services. This SQL server solution not only contributed to the system's stability and scalability but also adhered to industry standards for data management and security, reinforcing the reliability of the information system in providing psychological support.

In the implementation phase of our information system dedicated to psychological support, we adopted a cutting-edge approach by leveraging REST API, React, and seamless integrations for audio, video, and payment functionalities. The utilization of REST API facilitated smooth communication and data exchange between different components of the system, enhancing its flexibility and interoperability. By

employing React, a JavaScript library for building user interfaces, we ensured a responsive and intuitive user experience, allowing individuals to interact seamlessly with the psychological support services offered. The incorporation of audio and video integrations enriched the system's capabilities, enabling users to engage in virtual counseling sessions and access multimedia resources for a more personalized support experience. Additionally, the integration of a secure payment system enhanced the accessibility and sustainability of the information system, providing a streamlined mechanism for users to access premium features or services. This amalgamation of REST API, React, and versatile integrations exemplifies our commitment to harnessing modern technologies for the effective implementation of a comprehensive and user-centric psychological support platform.

In the pursuit of creating a robust and scalable information system for psychological support, the integration of cloud service storage emerges as a pivotal strategy. Leveraging the capabilities of cloud technology offers unparalleled advantages in terms of accessibility, flexibility, and resource management. By utilizing cloud storage solutions, the developed system ensures seamless access to a diverse array of support services for users across different devices and locations. This approach not only enhances the system's responsiveness but also promotes cost-effectiveness and efficient resource allocation. Furthermore, the scalability of cloud services accommodates the dynamic nature of user data, adapting effortlessly to changing storage requirements. The centralized nature of cloud storage streamlines data management, fostering collaboration and ensuring the security and integrity of user information. Embracing cloud service storage thus becomes integral to the information system, reinforcing its capacity to deliver personalized and reliable psychological support services while adhering to contemporary standards of technological innovation and efficiency.

In conclusion, this master's paper contributes a significant step forward in leveraging technology for psychological support. By combining the expertise of psychology with the capabilities of information systems and employing a client-server

architecture, it offers a holistic and scalable solution to address the evolving landscape of mental health care. The findings have implications for practitioners, policymakers, and researchers, emphasizing the need for ongoing collaboration to refine and expand the impact of technology in promoting psychological well-being.

Keywords: Psychological Support, Information System, Online Consultation, Privacy, Client-Server Architecture, Cloud Technology.

ЗМІСТ

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Психологічні аспекти надання допомоги через інформаційні системи ...	11
1.2 Етичні та правові аспекти розробки систем психологічної підтримки....	15
1.3 Аналіз аналогів програмних продуктів.....	17
2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕНЬ.....	23
2.1. Мета та задачі дослідження	23
2.2. Вибір методів для здійснення аналізу потреб користувачів.....	24
3. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ.....	27
3.1. Структурно-функціональне моделювання процесу взаємодії психолога з клієнтом	27
3.2 Моделювання діаграми варіантів використання	28
3.3 Проектування моделі бази даних	31
4 РОЗРОБКА ПІДСИСТЕМИ ПСИХОЛОГА.....	33
4.1 Розробка архітектури інформаційної системи.....	33
4.2 Розробка веб-додатку для підсистеми психолога.....	34
4.3 Розробка серверної частини для підсистеми психолога	39
4.4 Впровадження системи в хмарне середовище.....	42
4.5 Тестування інформаційної системи	43
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТОК А. ПЛАНУВАННЯ РОБІТ	50
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ	58

ВСТУП

Актуальність теми У сучасному світі, де стрес та психологічні виклики є постійними супутниками багатьох людей, питання надання психологічної допомоги за допомогою інформаційних технологій стає все більш актуальним. Розвиток цифрових технологій відкриває нові можливості для підтримки психологічного благополуччя людей, роблячи психологічну допомогу більш доступною та ефективною. Особливо це важливо в умовах, коли прямий доступ до психологів обмежений, наприклад, через географічні чи соціальні бар'єри.

Постановка проблеми Основна проблема полягає у відсутності інформаційної системи, яка не тільки задовольняє технічні вимоги, але й враховує психологічні аспекти взаємодії з користувачами, етичні норми та правові рамки. Це вимагає глибокого розуміння як технологічних, так і психологічних аспектів надання допомоги.

Метою цього дослідження є створення інформаційної системи, яка ефективно інтегрує технічні рішення з психологічними принципами надання допомоги, забезпечуючи при цьому високу етичність та юридичну відповідність.

Об'єкт і предмет дослідження Об'єктом дослідження є процес надання психологічної допомоги за допомогою інформаційних технологій. Предметом дослідження є використання сучасних інформаційних технологій для створення інформаційної системи, її архітектура, функції та взаємодія з користувачами з психологічної точки зору.

Методи дослідження Для досягнення мети будуть використовуватися методи аналізу документації, проектування систем, а також емпіричні методи, включаючи опитування, спостереження та вивчення випадків використання.

Практична значущість. Розроблена система матиме значний практичний вплив на поліпшення якості надання психологічних послуг. Вона зможе бути використана у різних установах, включаючи медичні центри, освітні заклади та

незалежні психологічні практики, забезпечуючи більш ефективну та доступну психологічну підтримку.

Задачі дослідження. Відповідно до поставленої мети потрібно виконати наступні задачі:

- Провести літературний огляд для визначення сучасних тенденцій та підходів до існуючих інформаційних систем для психологічної підтримки. Визначити функціональні вимоги до інформаційної системи на основі аналізу галузі.

- Розробити концепцію системи, визначивши основні модулі та їх взаємодію. Сформулювати стратегію забезпечення безпеки даних та конфіденційності користувачів. Визначити технологічні стеки та інструменти, які будуть використані при розробці.

- Провести структурно-функціональне моделювання розроблюваної системи.

- Здійснити програмну реалізацію відповідної інформаційної системи. Створити дизайн та інтерфейс системи, враховуючи потреби психологів та їх клієнтів. Реалізувати необхідні інтеграції єдиної системи.

- Провести тестування розробленої системи.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Психологічні аспекти надання допомоги через інформаційні системи

Вплив візуального та аудіального сприймання на користувачів є важливим аспектом розробки систем психологічної підтримки [1]. Психологічні аспекти сприймання інформації через інформаційні системи можуть впливати на ефективність та ефективність надання допомоги [2].

Важливо враховувати, що різні типи інформаційних систем можуть використовувати різні методи подачі інформації, такі як текстові, відео та аудіо формати [3]. Вплив цих форматів може бути різним. Наприклад, відеоролики можуть бути більш емоційно збагаченими, але вимагають більше уваги від користувача, в той час як текст може бути більш структурованим і дозволяти користувачам легше зосередитися [4].

Важливим аспектом є розробка інтерфейсів, які враховують психологічні особливості користувачів [5]. Наприклад, використання кольорів, шрифтів і структури інтерфейсу може впливати на настрій та сприйняття інформації. Інтерфейс повинен бути зручним і легким у використанні, щоб забезпечити позитивний досвід користувача [6].

Для досягнення успіху у наданні психологічної підтримки через інформаційні системи, важливо також враховувати психологічний стан користувачів [7]. Моніторинг психологічного стану користувачів під час використання системи може допомогти вчасно реагувати на їхні потреби та надавати необхідну допомогу [8].

Психологічні методи підвищення мотивації користувачів також можуть грати важливу роль у наданні психологічної підтримки [9]. Сприяння активності та участі користувачів у системі може покращити їхні результати та задоволення від використання психологічних інструментів.

Загалом, вплив візуального та аудіального сприймання на користувачів є складним і многогранним аспектом розробки систем психологічної підтримки. Для досягнення успіху в цій області важливо враховувати різні аспекти сприймання інформації та забезпечувати належний психологічний підхід до користувачів.

Стратегії комунікації та психологічна підтримка через різні формати, такі як текст, відео та аудіо, важливі для надання ефективної допомоги та сприяння психологічному благополуччю користувачів [10].

Почнемо з текстової комунікації. Вона є одним із найпоширеніших способів передачі інформації та психологічної підтримки. Текст може бути корисним для надання конкретних інструкцій, подачі інформації та висловлення підтримки [11]. Важливо забезпечити зрозумілість тексту та його легку доступність для різних категорій користувачів. Додатково, текстова комунікація може бути корисною для тих, хто володіє обмеженими аудіо- або візуальними можливостями [12].

Відео формат також має великий потенціал для надання психологічної підтримки. Відеоролики можуть створювати емоційно збагачений контент, що дозволяє користувачам краще сприймати інформацію [13]. Ілюстрація слів за допомогою візуальних зображень і жестів може зробити комунікацію більш привабливою і легкозасвоюваною [14]. Важливо враховувати, що відео може бути більш часо- та ресурсномістким у виробництві, але його ефективність може бути вищою, якщо він створюється професійно [15].

Аудіо формат також може бути важливим інструментом комунікації та психологічної підтримки. Подача інформації через звук може бути корисною для тих, хто віддає перевагу аудіальному сприйманню [16]. Психологічна підтримка в аудіо форматі може включати розмови, гіпноз, медитації тощо. Для користувачів з візуальними обмеженнями аудіо може бути основним джерелом інформації та підтримки [17].

Успішна стратегія комунікації та психологічної підтримки часто полягає в поєднанні різних форматів залежно від потреб користувачів. Наприклад, текст може служити для надання деталей та інструкцій, відео - для емоційної підтримки, а аудіо - для релаксації та медитації [18].

Загалом, стратегії комунікації та психологічної підтримки через різні формати грають важливу роль у наданні допомоги користувачам.

Розробка інтерфейсів з урахуванням психологічних особливостей користувачів є важливим етапом при створенні інформаційних систем для психологічної підтримки [19].

Таблиця 1.1 – Методи розробки інтерфейсів з урахуванням психологічних особливостей користувачів.

Джерело: побудовано автором

Метод	Опис
1. Персоніфікація	Створення інтерфейсів, які можна налаштовувати з урахуванням індивідуальних потреб користувачів
2. Тестування і відгуки	Залучення користувачів до тестування інтерфейсу та збір їхніх відгуків для подальшого вдосконалення
3. Емпатія та співпраця	Співпраця з психологами та фахівцями з людських ресурсів для розробки інтерфейсів, які відповідають психологічним потребам користувачів
4. Використання технік	Використання методів психології, таких як позитивна підсилення, для створення мотиваційного інтерфейсу
5. Аналіз поведінки користувачів	Моніторинг та аналіз дій користувачів у інтерфейсі для виявлення патернів та покращення дизайну

Розробка інтерфейсів з урахуванням психологічних особливостей користувачів є важливою для створення ефективних та користувач орієнтованих інформаційних систем для психологічної підтримки. Врахування психологічних аспектів дозволяє покращити якість обслуговування та сприяти позитивному досвіду користувачів.

Моніторинг психологічного стану користувачів під час використання інформаційних систем є ключовою складовою для забезпечення якості та ефективності інтерфейсу. Зрозуміння емоцій, сприйняття та поведінки користувачів грає важливу роль у розробці та вдосконаленні інформаційних систем.

Моніторинг емоційного стану користувачів також набуває важливості. Виявлення емоцій користувачів під час взаємодії з інформаційною системою може проводитися шляхом аналізу фізіологічних показників або реакцій обличчя. Це допомагає виявити, чи сприймають користувачі інтерфейс як позитивний чи негативний, чи викликає він стрес чи релаксацію[20].

Фідбек від користувачів також має велике значення. Забезпечення можливості надсилати відгуки та коментарі щодо інтерфейсу дозволяє користувачам висловлювати свої побажання та виявляти проблеми, з якими вони стикаються. Такий фідбек може бути використаний для подальшого вдосконалення системи.

Нарешті, проведення тестів з реальними користувачами є ефективним способом виявлення проблем та оцінки реакції користувачів. Тестування може розкрити недоліки в інтерфейсі та допомогти вирішити їх для покращення взаємодії з системою.

Моніторинг психологічного стану користувачів є невід'ємною частиною розробки інформаційних систем. Враховуючи емоції, поведінку та сприйняття користувачів, можна створити інтерфейс, який буде не тільки ефективним, але й сприятиме психологічному комфорту та задоволенню користувачів.

1.2 Етичні та правові аспекти розробки систем психологічної підтримки

Забезпечення конфіденційності інформації користувачів є критичним аспектом розробки та використання інформаційної системи підтримки надання психологічної допомоги. Користувачі мають право на захист своєї особистої інформації та довіри до системи, щоб відчувати себе комфортно та безпечно. У цій статті розглянемо ключові аспекти забезпечення конфіденційності інформації користувачів.

Шифрування даних є однією з основних технологій, що забезпечують конфіденційність. Всі дані, які передаються між користувачем і системою, повинні бути зашифровані, щоб унеможливити несанкціонований доступ [21]. Для цього можуть використовуватися протоколи шифрування, такі як SSL або TLS.

Політика конфіденційності повинна бути ретельно розроблена і зрозуміло представлена користувачам [22]. Ця політика має включати в себе інформацію про те, як збирається, зберігається, оброблюється та захищається особиста інформація користувачів, а також про те, з якими сторонами може бути поділена ця інформація.

Ролі та доступ до інформації повинні бути чітко визначені та обмежені. Система повинна мати механізми автентифікації та авторизації, щоб дозволити доступ тільки користувачам з необхідними правами [23].

Інформування та навчання користувачів щодо методів забезпечення конфіденційності є важливим кроком. Користувачі повинні знати, як захищати свої дані, як створювати надійні паролі та як уникати фішингових атак [24].

Моніторинг та реагування на порушення конфіденційності інформації також є важливими. Система повинна мати механізми для виявлення незвичайних або підозрілих активностей та можливість реагувати на них негайно [25].

Зберігання інформації також має бути надійним і безпечним. Для цього можуть використовуватися сучасні методи зберігання даних та резервне копіювання [26].

Забезпечення конфіденційності інформації користувачів є надзвичайно важливим завданням при розробці та використанні системи психологічної підтримки. Це забезпечує довіру користувачів і дозволяє їм відчувати себе комфортно при користуванні системою.

Конфіденційність і приватність: Захист особистої інформації користувачів є обов'язковим. Система повинна забезпечувати конфіденційність та не розголошувати особисту інформацію без згоди користувача.

Закони щодо захисту особистих даних, такі як Загальний регламент про захист даних (GDPR) в Європейському Союзі, встановлюють обов'язки для організацій, які збирають та обробляють особисті дані користувачів [27]. Ці закони передбачають, що користувач повинен надавати інформовану згоду на обробку своїх даних, та має право на доступ до своїх даних та їх видалення.

Психологічні характеристики користувачів є особливо чутливою категорією даних, оскільки вони можуть розкривати інтимну інформацію про особистість та стан особи. Тому важливо дотримуватися прозорих і етичних підходів до їх збору та обробки. Фахівці та системи повинні надавати користувачам повний контроль над своєю інформацією та забезпечувати конфіденційність.

У підсумку, законодавчі обмеження та правила використання даних психологічних характеристик користувачів є важливими для забезпечення конфіденційності та етики в психологічній допомозі через інформаційні системи. Дотримання цих норм і стандартів сприяє збереженню довіри користувачів та забезпечує їхню безпеку та конфіденційність.

Важливо зауважити, що етичні аспекти збору та аналізу даних грають велику роль у цьому процесі. Користувачі повинні бути повністю інформовані про те, як їхні дані збираються та використовуються, і мають право на контроль

над своєю інформацією. Розробники та оператори системи повинні дотримуватися етичних стандартів та заходів захисту конфіденційності даних.

Збір та аналіз статистичних даних щодо потреб користувачів є важливою складовою розробки і вдосконалення інформаційних систем у сфері психологічної підтримки. Цей процес допомагає зрозуміти потреби користувачів, покращити якість послуг та забезпечити етичну обробку особистих даних.

Дослідження трендів та змін у психологічних потребах користувачів є надзвичайно важливим завданням для розробки та вдосконалення інформаційних систем у сфері психологічної підтримки. Світ постійно змінюється, і це включає в себе еволюцію психологічних потреб та трендів у сфері психологічної допомоги.

Вплив технологій: Сучасні технології, зокрема мобільні додатки та веб-платформи, стали важливим засобом надання психологічної допомоги [28]. Дослідження включають в себе аналіз популярних платформ та додатків, які використовуються для забезпечення психологічної підтримки.

1.3 Аналіз аналогів програмних продуктів

Психологам дуже важливо уважно вирішувати питання безпеки, конфіденційності та зручності для клієнтів при використанні інформаційних систем для надання психологічної допомоги[29].

Аналізуючи сучасні інформаційні системи з надання психологічних послуг, то ми можемо врахувати ряд недоліків та переваг. Для початку розглянемо переваги:

1. Гнучкість графіка: Психолог може працювати в онлайн-режимі, що дозволяє гнучко встановлювати робочий графік та працювати з клієнтами з різних часових зон [30].
2. Розширені можливості: Використання інформаційних систем дозволяє психологу використовувати різноманітні технічні інструменти, включаючи

відеоконференції, чати та інші засоби для покращення комунікації та надання допомоги.

3. Зручність для клієнтів: Клієнти можуть отримувати психологічну підтримку у зручному для них місці без необхідності фізичного переїзду [31].
4. Збереження інформації: Інформаційні системи дозволяють зберігати інформацію про клієнтів з безпекою та легкістю, що може бути корисним для подальшої роботи та аналізу [32].

Також важливо, оцінюючі явні переваги інформаційних систем, розуміти і недоліки, котрі можуть існувати при використанні цих систем, а саме:

1. Відсутність невербальної комунікації: Відсутність невербальних сигналів може ускладнити розуміння емоцій та стану клієнта, порівняно з особистою комунікацією [33].
2. Технічні труднощі: Проблеми з інтернет-з'єднанням або технічні неполадки можуть призвести до перерв у наданні послуг та створити стрес як для психолога, так і для клієнта [34].
3. Сприятливе середовище для клієнта: Важливо, щоб клієнти були у зручному та конфіденційному середовищі під час отримання психологічної допомоги, і це може бути викликано складнощами в їхній домашній обстановці.
4. Можливість перешкодження конфіденційності: При неправильному використанні або порушенні безпеки інформаційної системи можливе порушення конфіденційності особистих даних клієнтів [35].

Останнім часом ріст інформаційних технологій призвів до появи великої кількості різноманітних систем для надання психологічної допомоги [36]. Розглянемо декілька сервісів з точки зору взаємодії психолога з ними. Наприклад, TherapyNotes [37], SimplePractice [38], Mindly [39].

Розпочнемо аналіз із системи TherapyNotes.

The screenshot displays the TherapyNotes interface for a clinician. At the top, there is a navigation menu with options: To-Do, Scheduling, Patients, Staff, Billing, and Payers. Below the menu, a welcome message reads "Welcome, Clinician 1" and "ICare LLC Welcome Message" with the text "Don't Worry, Be Happy!".

The main section is titled "To-Do List: 48 Items" and contains a table of tasks:

Date	Patient	To-Do Item
9/26/16		Confirm appointment requests 2/2
9/26/16		Submit EDI claims 37
9/26/16		Create CMS-1500 forms 13
9/26/16		Mark externally submitted items 6
9/26/16		Review EDI claim rejections 4
9/1/16	Su,Su	Create a Psychotherapy Progress Note for Therapy Session (90834) on 9/1 at 10a. ×
9/4/16	Ma,A	Create a Psychotherapy Progress Note for Therapy Session (90837) on 9/4 at 12p. ×

Below the to-do list is the "Upcoming Agenda" section, which shows a calendar view for Monday (9/26), Tuesday (9/27), and Wednesday (9/28). Each day has a list of scheduled activities with time slots and assigned staff:

- Monday 9/26:**
 - 8:00a - 10:00a: Breakfast & Notes - Supervisor 1
 - 9:00a - 9:45a: In:Ed: Therapy Session (90834): Intern 1
 - 11:00a - 11:45a: In:Pa: Therapy Session (90834): Clinician 1
 - 2:00p - 3:00p: Su,Su: Therapy Session (90837): Clinician 2
- Tuesday 9/27:**
 - 9:00a - 9:30a: In:Ed: Therapy Session (90832): Clinician 1
 - 12:00p - 1:00p: Di:Di: Therapy Session: Supervisor 1
 - 2:00p - 2:45p: Ma:A: Therapy Session (90834): Clinician 1
 - 3:00p - 3:45p: In:Ed: Therapy Session (90834): Intern 1
- Wednesday 9/28:**
 - 8:00a - 10:00a: Breakfast & Notes - Supervisor 1
 - 11:00a - 11:45a: In:Pa: Therapy Session (90834): Clinician 1
 - 1:00p - 1:30p: Ma:A: Therapy Session (90832): Clinician 1
 - 5:00p - 5:45p: In:Pa: Therapy Session (90834): Clinician 2

Рисунок 1.1. – Інтерфейс кабінета психолога на TherapyNotes

Джерело: побудовано автором (знімок з екрану)

Дана інформаційна система має потужний інструмент для роботи психолога. Він включає можливість створення календарів з поточними записами, інтеграцію з іншими сервісами за допомогою API, створення заміток до консультацій, робота з платіжними системами. Явною перевагою сервісу є інтуїтивно зрозумілий інтерфейс. Можливість налаштування планів дій на певний проміжок часу та ведення заміток про клієнтів. Недоліком цієї системи є відсутність можливості спілкування та взаємодії з клієнтами та відсутність локалізації. Ресурс підтримує виключно американську локалізацію, що може спричинити певні незручності при роботі.

Далі розглянемо наступний ресурс для порівняння. Це сервіс SimplePractice.

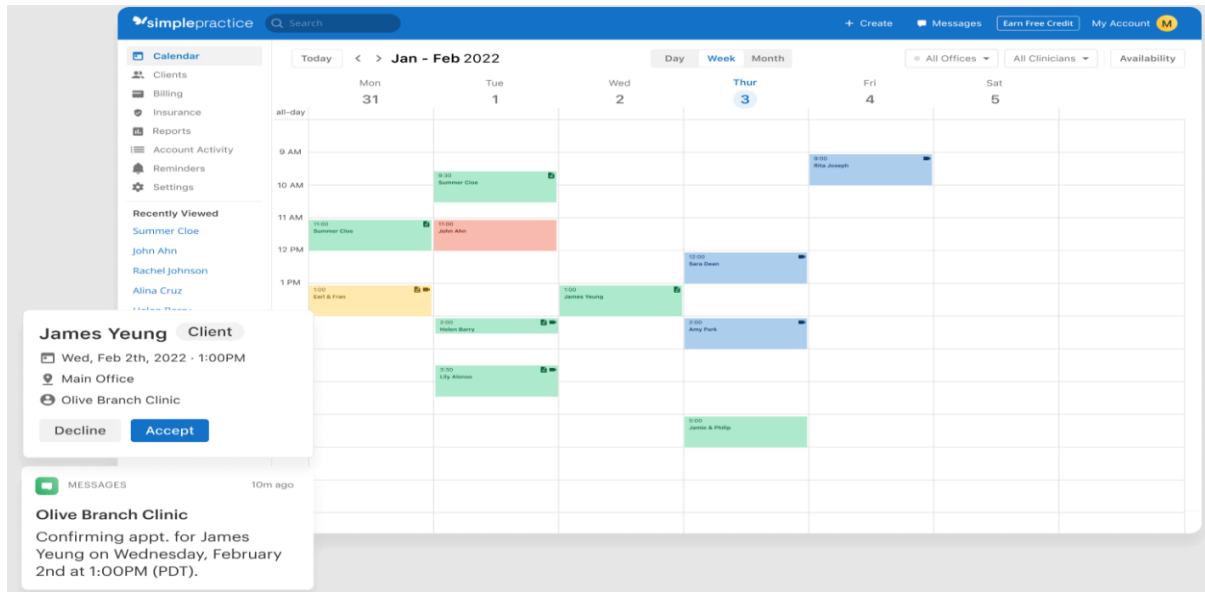


Рисунок 1.2. - Інтерфейс кабінета психолога на SimplePractice

Джерело: побудовано автором (знімок з екрану)

Явною перевагою даного сервісу є візуалізація клієнтського інтерфейсу. Даний інтерфейс має зручний та зрозумілий вигляд при відображенні календаря подій. Має безліч міток для персоніфікації та налаштувань. Беззаперечною перевагою цього ресурсу є можливість створити контакт з клієнтом, провести первинну консультацію в форматі часу, де клієнт коротко може висловити свою проблему. Сервіс має пуш-повідомлення для нагадувань та за необхідності має інтеграцію з календарями на мобільних пристроях.

Даний сервіс має декілька недоліків. По-перше це платний ресурс з доволі високою вартістю підписки. Іншим мінусом можна відмітити, аналогічний до першого ресурсу, відсутність локалізації до українського ринку.

Останнім в нашому списку порівнянь сервісів для роботи психолога з надання психологічної допомоги є український стартап Mindly.

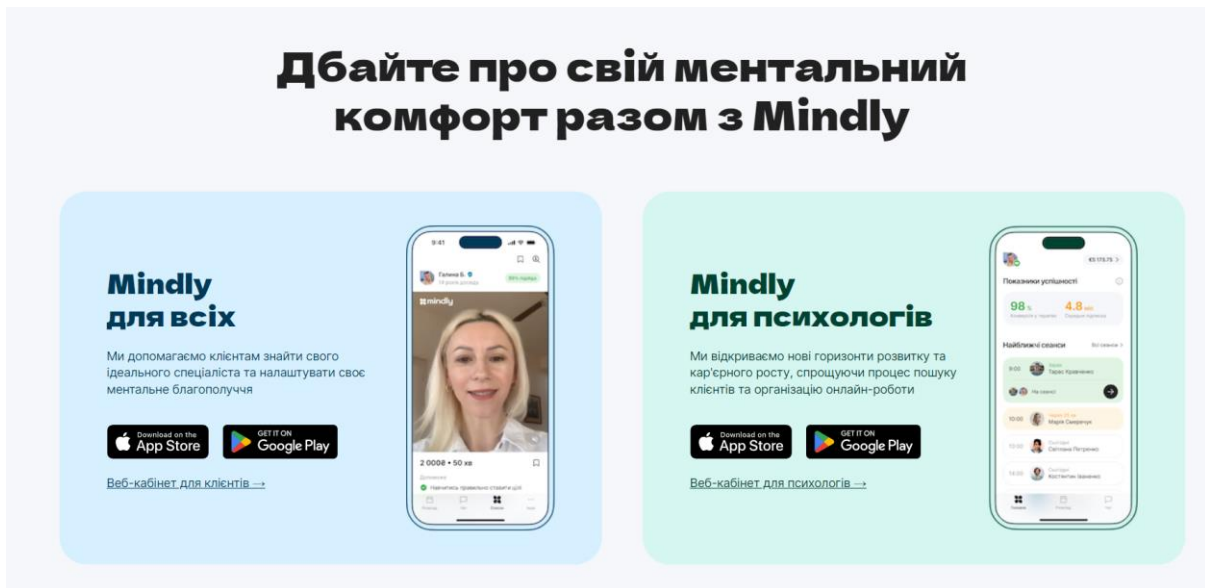


Рисунок 1.3 - Інтерфейс сервісу Mindly

Джерело: побудовано автором (знімок з екрану)

Даний продукт набирає чималу популярність останнім часом. Цей продукт орієнтований саме на українського споживача та має адаптований інтерфейс, що є явною перевагою. Mindly має велику перевагу в алгоритмах підбору клієнтів і психологів. Кожен психолог та клієнт мають створити відповідну анкету за результати якої буде підібраний спеціаліст. Проект знаходить в стадії активної розробки, тому поки не має можливості спілкуватися в додатку користувачу та психологу, але ця функція запланована на майбутні релізи. Також до недоліків можна віднести лише підтримку мобільної версії додатку.

Проаналізувавши всі три джерела можна зробити висновок, що кожна система має перелік переваг та недоліків. Серед переваг більшість систем мають зручний інструмент планувань, інтеграцію з сервісами такими як календар, методології підбору психологів та клієнтів. Серед недоліків варто зазначити відсутність української локалізації та відсутність зашифрованих каналів передачі даних для використання в аудіо- та відео- консультаціях.

Оцінка кожної системи враховує баланс всіх вказаних факторів. Важливо, щоб нова інформаційна система аналізувала як переваги, так і недоліки існуючих

рішень, і також пропонувала інновації, які відповідали б вимогам сучасного ринку психологічних послуг.

Таблиця 1.2 – Порівняльна оцінка проаналізованих систем

Джерело: побудовано автором

Критерій	TherapyNotes	SimplePractice	Mindly	Система, що розробляється
Функціональність	6	9	5	9
UI/UX	7	8	6	6
Безпека даних	9	6	6	9
Інтеграція	8	7	3	8
Локалізація	1	3	8	9
Загальна оцінка	6.2	6.8	5.9	8.2

2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕНЬ

2.1. Мета та задачі дослідження

Інформаційна система підтримки надання психологічної допомоги є важливим інструментом у сфері психологічної підтримки та здоров'я. Ця система має на меті забезпечити ефективну та доступну допомогу людям, які потребують психологічної підтримки, за допомогою інформаційних технологій. Базуючись на потребах користувачів, система повинна мати наступні функції, а саме – можливість реєстрації, авторизації та заповнення персональної інформації. Фундаментальною частиною є саме впровадження надання онлайн консультацій за допомогою аудіо- та відео- зв'язку та можливість керувати ними.[40].

Головною метою є розробка інформаційної системи для поліпшення якості та доступності психологічної допомоги. Для досягнення цієї мети можна сформулювати наступні цілі та завдання:

1. Провести літературний огляд для визначення сучасних тенденцій та підходів до існуючих інформаційних систем для психологічної підтримки. Визначити функціональні вимоги до інформаційної системи на основі аналізу галузі.
2. Розробити концепцію системи, визначивши основні модулі та їх взаємодію. Сформулювати стратегію забезпечення безпеки даних та конфіденційності користувачів. Визначити технологічні стеки та інструменти, які будуть використані при розробці.
3. Провести структурно-функціональне моделювання розроблюваної системи.
4. Здійснити програмну реалізацію відповідної інформаційної системи. Створити дизайн та інтерфейс системи, враховуючи потреби психологів та їх клієнтів. Реалізувати необхідні інтеграції єдиної системи.
5. Провести тестування розробленої системи.

Загалом, розробка інформаційної системи підтримки надання психологічної допомоги є важливим завданням для сучасного суспільства, оскільки вона дозволяє забезпечити більш доступну та ефективну психологічну допомогу для всіх, хто цього потребує. Реалізація цих цілей та завдань допоможе покращити психологічне здоров'я і якість життя багатьох людей [41].

2.2. Вибір методів для здійснення аналізу потреб користувачів

Вибір методів для здійснення аналізу потреб користувачів є важливим етапом у розробці інформаційної системи підтримки надання психологічної допомоги. Для досягнення мети розробки і впровадження такої системи, необхідно вивчити потреби та очікування користувачів, щоб створити зручний і ефективний інструмент, спрямований на поліпшення психологічного благополуччя та надання підтримки. У цьому контексті розглянемо три основні методи аналізу потреб користувачів [42].

- 1. Анкетування та опитування:** Цей метод передбачає проведення опитування серед потенційних користувачів системи. Опитування може містити запитання щодо їхніх психологічних потреб, проблем, очікувань від системи та пріоритетів у наданні психологічної допомоги. Цей метод дозволяє отримати кількісні дані та загальну уяву про потреби користувачів.
- 2. Групові обговорення:** Фокус-групи з різними групами користувачів можуть бути ефективним інструментом для глибшого розуміння їхніх потреб. Під час обговорень учасники можуть ділитися своїми думками, ідеями та специфічними вимогами до системи. Цей метод дозволяє виявити психологічні особливості різних груп користувачів і врахувати їх у розробці.
- 3. Аналіз існуючих ресурсів:** Важливо також вивчити існуючі інформаційні системи та ресурси, які надають психологічну допомогу. Аналіз сильних та слабких сторін таких систем може допомогти ідентифікувати недоліки та

можливості для покращення. Дослідження цих ресурсів може служити основою для розробки інноваційних функцій та сервісів у вашій системі.

Комбінування цих методів дозволить отримати комплексне розуміння потреб користувачів і створити інформаційну систему, яка відповідає їхнім потребам та сприяє поліпшенню психологічного благополуччя. Аналіз потреб користувачів має бути систематичним та постійним процесом, що дозволяє враховувати зміни в їхніх потребах та реагувати на них у розвитку системи [43].

Цей процес вимагає багато роботи та співпраці з багатьма фахівцями, але в результаті ми створили інформаційну систему, яка допомагає покращити психологічний стан клієнта та надання психологічної допомоги нашим користувачам.

За основу при розробці додатку була взята клієнт-серверна веб архітектура. При виборі даної архітектури головним аспектом є вибір мов програмування для реалізації програмного забезпечення. Також важливим фактором є використання баз даних, фреймворків, інструментів розробки та інфраструктури розміщення.

Для даної інформаційної системи беззаперечно важливим критерієм при виборі мови програмування є підтримка об'єктно-орієнтованого підходу, тому прийняте рішення використовувати мову C# та платформу .NET Core.

Для створення серверної частини, котра буде представляти собою RESTful Web API, платформа .NET Core надає широкий перелік вже готових рішень. ASP.NET Core Web API є найсучаснішим рішенням від компанії Microsoft. Цей фреймворк поєднує в собі останні технології щодо впровадження безпекових рішень за допомогою використання токенів авторизації, швидкодію та вбудовану документацію.

Оскільки частина серверної архітектури була вибрана із сімейства продуктів компанії Microsoft, то доцільно для кращої взаємодії компонентів використовувати повні стек технології цієї компанії.

Microsoft SQL Server – це система управління базою даних, котра впроваджує реляційний підхід в побудові бази даних. Вона надає ефективний

механізм зберігання та управління даними, що дозволяє користувачам створювати, керувати та отримувати доступ до баз даних.

Microsoft Azure є хмарною платформою, розробленою компанією Microsoft. Це великий набір хмарних послуг, які включають інфраструктуру, платформу та програмне забезпечення. Основна ідея Azure полягає в тому, щоб дати користувачам можливість будувати, розгортати та керувати додатками та послугами на інфраструктурі, яка знаходиться в хмарі. Тому розміщення інфраструктури додатка буде використане за допомогою Microsoft Azure.

Клієнтська частина додатку буде створена за допомогою фреймворку React на мові програмування Javascript. React широко використовується у веб-розробці, і його популярність зумовлена зручністю використання, швидкістю розробки та активною спільнотою розробників.

Використання цих технологій та інструментів у проекті дозволяє створити потужну, безпечну і гнучку систему, яка ефективно вирішує поставлені бізнес-завдання. Такий комплексний підхід до розробки не тільки сприяє створенню високоякісного продукту, але й істотно прискорює процес розробки та зменшує витрати на підтримку та розвиток проекту.

3. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

3.1. Структурно-функціональне моделювання процесу взаємодії психолога з клієнтом

Для цільової задачі цієї роботи чітко визначено завдання - розробити підсистему психолога для інформаційної системи, яка надає психологічну допомогу. У виконанні цього завдання планується використання вхідних та вихідних даних, де вхідні дані включають інформацію про фах, досвід та напрямок роботи психолога. Результатом роботи будуть рекомендації психолога клієнту та методики поліпшення стану. Учасниками процесу є розробник, клієнт, апаратне забезпечення, а також додаткові веб-сервіси. Відповідна контекстна діаграма в нотації IDEF0, орієнтована на психолога, представлена на рисунку 3.1.

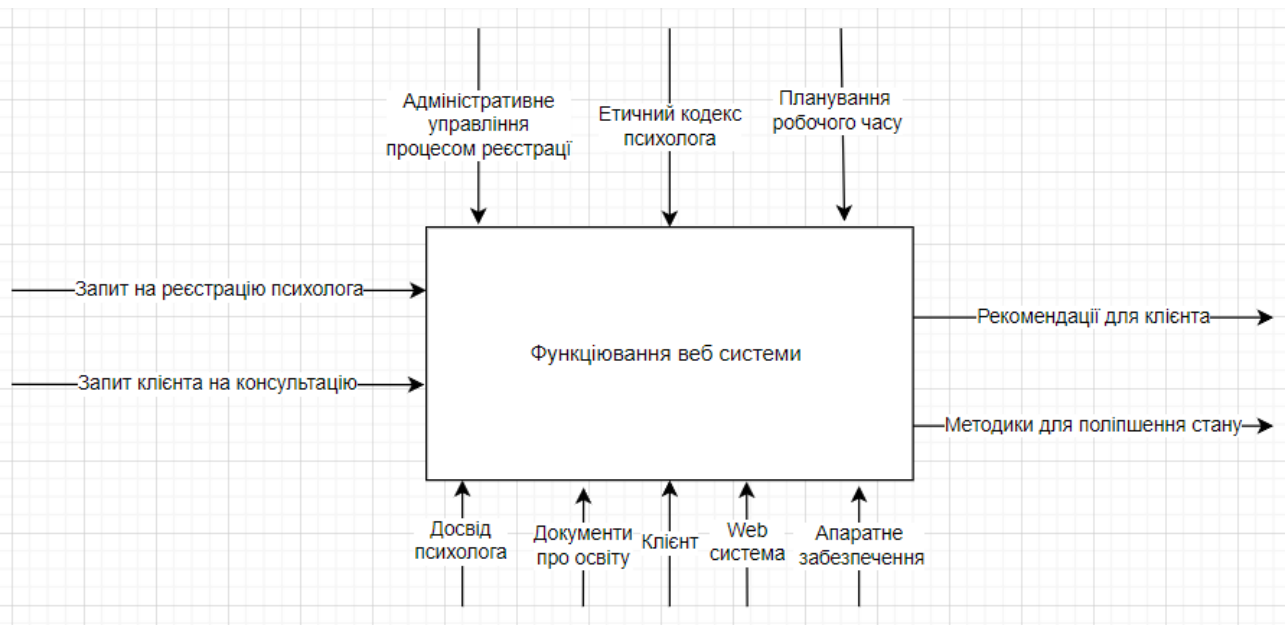


Рисунок 3.1 - Контекстна діаграма інформаційної системи по процесу надання психологічної допомоги

Джерело: побудовано автором

Розглянемо діаграму декомпозиції IDEF0 для інформаційно системи (рис. 3.2).

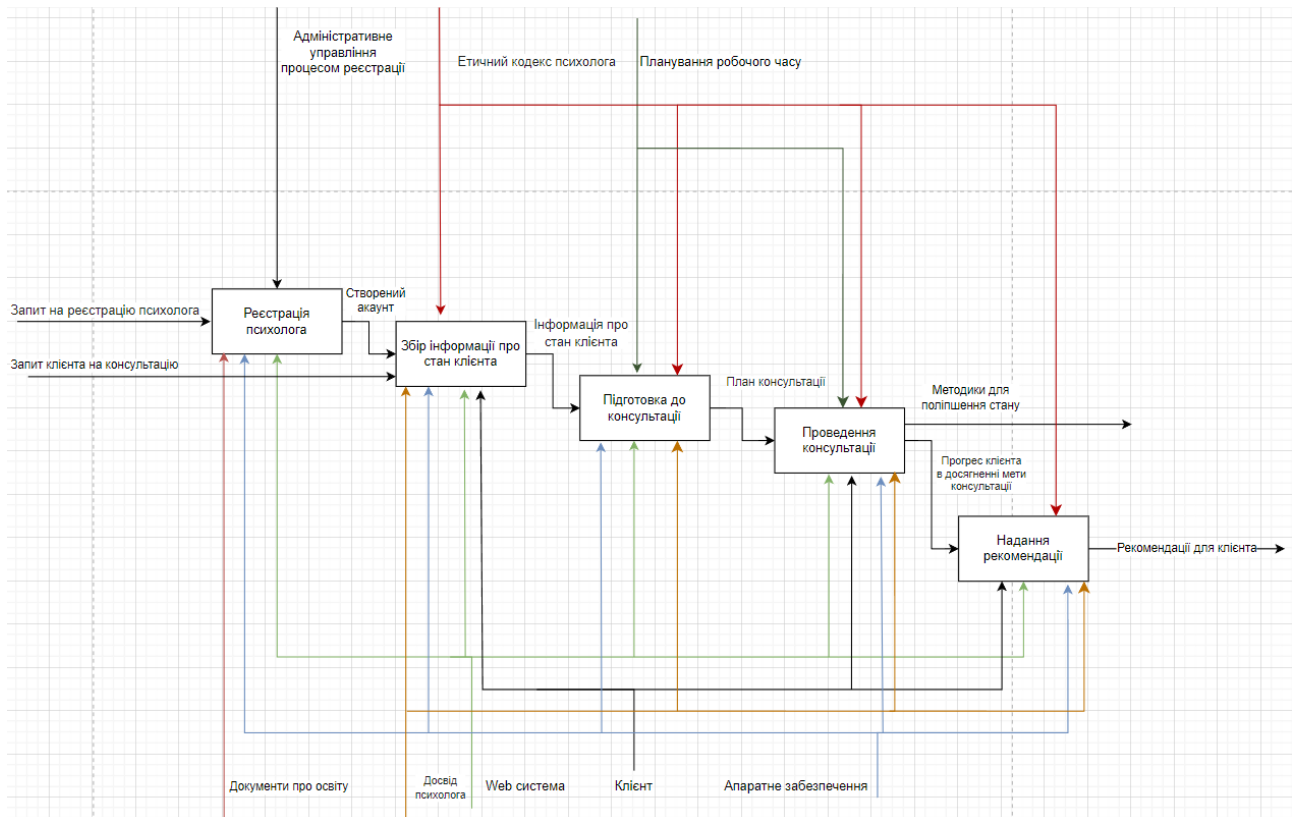


Рисунок 3.2 - Декомпозиція контекстної діаграми по процесу надання психологічної допомоги

Джерело: побудовано автором

Ця розбудова контекстної діаграми відображає ключові функції системи та їх взаємозв'язки. Всі функції представлені у вигляді блоків, а стрілки показують напрямок потоку інформації. Ця діаграма може служити базою для подальшого детального моделювання та проектування інформаційної системи.

3.2 Моделювання діаграми варіантів використання

Після створення контекстних діаграм, необхідно розробити діаграму варіантів використання. Ця діаграма встановлює, який функціонал розробляється програмною системою доступний для кожної групи користувачів.

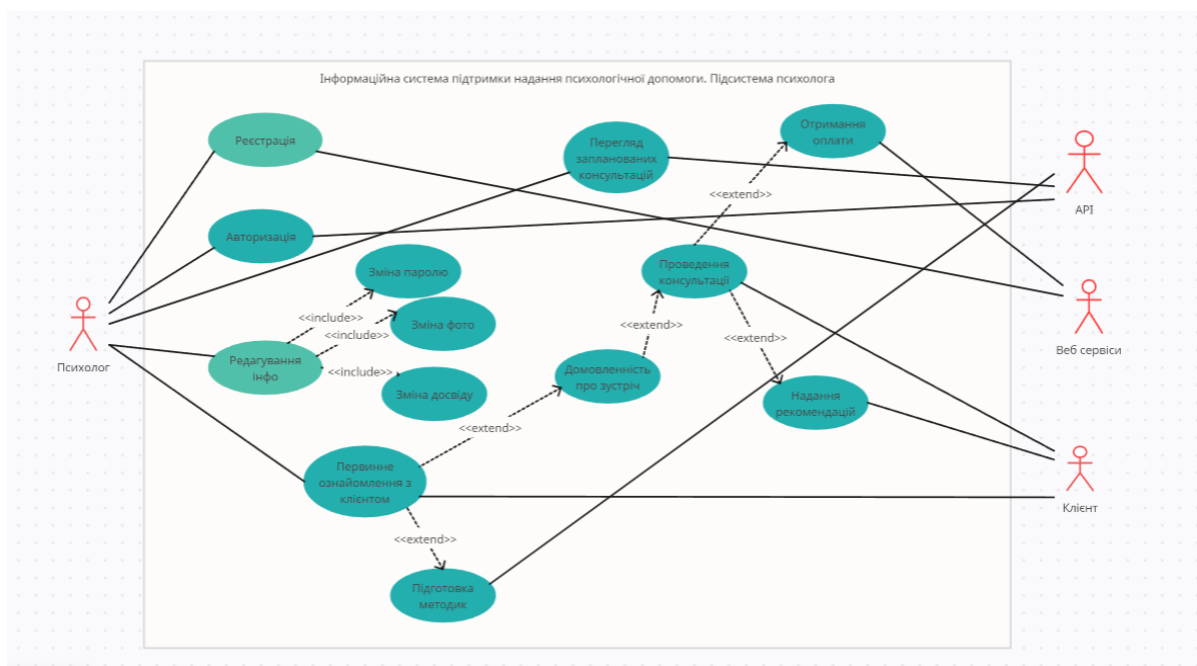


Рисунок 3.3 – Діаграма варіантів використання інформаційної системи

Джерело: побудовано автором

На цій діаграмі зображено взаємодію між психологом, клієнтом, сервісами та апаратним та програмним забезпеченням. Вона описує різноманітні сценарії для користувачів (рис. 3.3). Подробиці про акторів та сценарії використання можна знайти нижче в таблицях 3.1 та 3.2.

Таблиця 3.1. – Опис акторів

Джерело: побудовано автором

№	Іменування актора	Детальний опис
1	Психолог (П)	Ключовий компонент системи, який здійснює введення, виведення та взаємодію з іншими елементами для отримання необхідної інформації або виконання конкретних дій у межах інформаційної системи.
2	Клієнт (К)	Актор, який взаємодіє з системою з метою отримання психологічної підтримки та взаємодії з Психологом.
3	API (А)	Це актор, який зберігає та управляє інформацією, необхідною для роботи системи. Може включати дані про користувачів, психологічні аспекти та інші відомості.
4	Веб-сервіси (В)	Актор, що координує взаємодію між різними компонентами системи через Інтернет.

Таблиця 3.2. – Опис варіантів використання

Джерело: побудовано автором

№	Варіанти використання	Детальний опис	Актор
1	Реєстрація	Функція для реєстрації психолога	П/В
2	Авторизація	Функція для авторизації психолога	П/В
3	Редагування інформації	Психолог може змінити інформацію про свій профіль	П/А
4	Перегляд запланованих консультацій	Психолог має можливість бачити список запланованих консультацій	П/А
5	Первинне знайомство з клієнтом	Психолог має можливість встановлення зв'язка з клієнтом та домовленість про консультацію	П/К
6	Проведення консультації	Психолог проводить консультацію за результати якої надає клієнту рекомендації	П/К

3.3 Проектування моделі бази даних

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

ER-діаграма є графічним представленням структури даних, де сутності відображаються у вигляді таблиць, а зв'язки представлені стрілками, що вказують

на взаємозв'язки між сутностями. Використання ER-діаграми дозволяє розробникам, аналітикам та іншим учасникам проекту легко взаємодіяти та розуміти основні аспекти бази даних. На рисунку 3.4 подано ER-діаграму бази даних для інформаційної системи, визначеної в завданні.

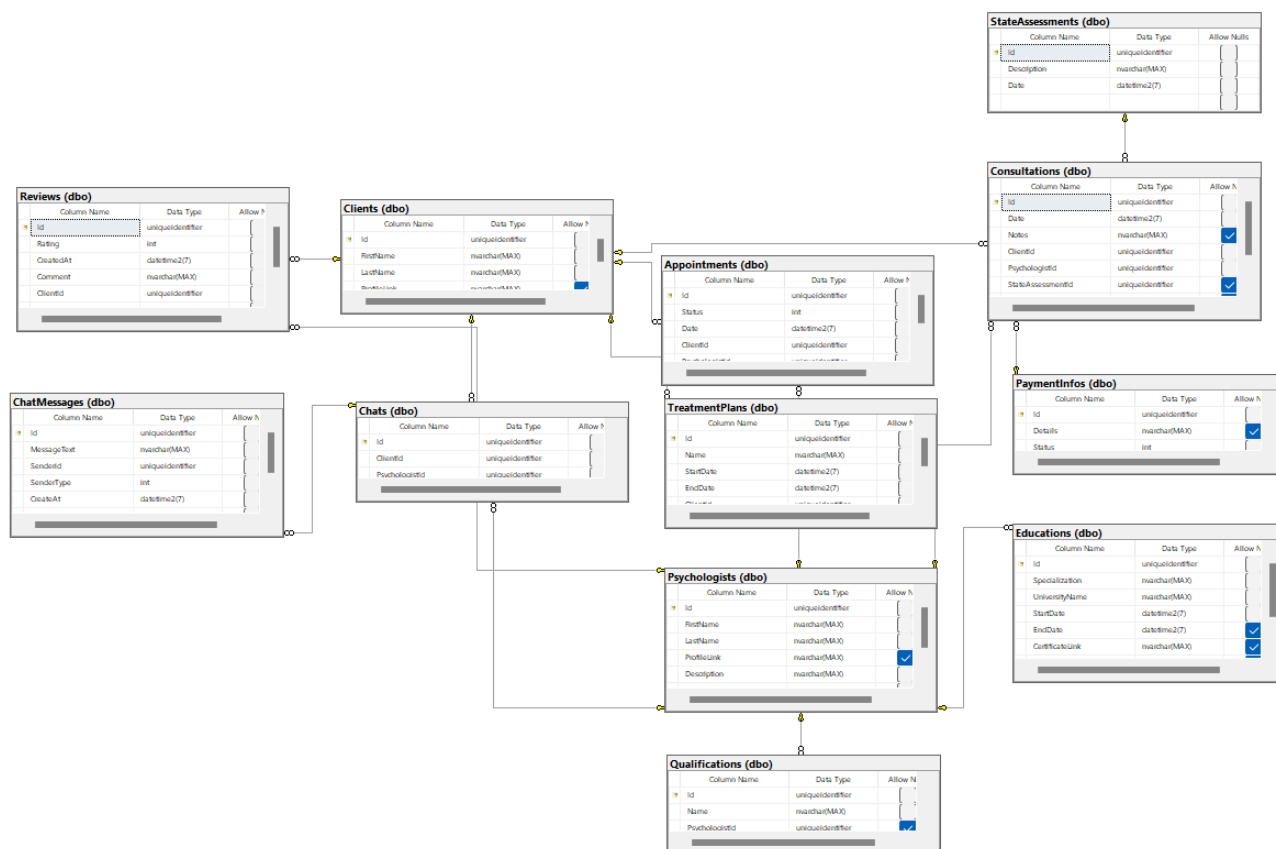


Рисунок 3.4 – ER-діаграма бази даних інформаційної системи

Джерело: побудовано автором

4 РОЗРОБКА ПІДСИСТЕМИ ПСИХОЛОГА

4.1 Розробка архітектури інформаційної системи

Створення архітектури інформаційної системи є ключовим етапом у розробці програмного продукту чи набору програм, спрямованих на обробку інформації. Архітектура визначає організацію системи, взаємозв'язки між її складовими частинами, принципи їх взаємодії та загальний механізм функціонування. Взаємодію клієнт-серверної архітектури можна описати наступним чином(рис 4.1)

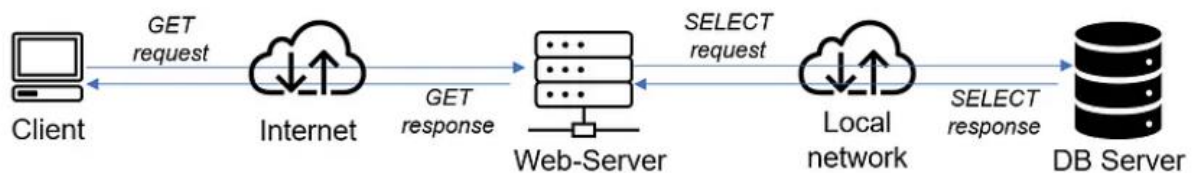


Рисунок 4.1 – Архітектура інформаційної системи

Джерело: побудовано автором

Клієнт-сервера архітектура була вибрана не випадково при розробці інформаційної системи. Серверна частина слугує надійним джерелом захищеності етичних та безпекових норм в системі. Структурно серверна частина складається з наступних компонентів : REST Web API, реляційна база даних, веб сервісів для роботи з фінансовими операціями та аудіо-, відео- потоками. Клієнтська частина в інформаційній системі складається з веб додатку. Також потрібно відокремити хмарне середовище котре буде слугувати для провадження як серверної, так і клієнтської частини.

Розроблена архітектура інформаційної системи демонструє глибоке збалансування, що базується на основних принципах клієнт-серверної архітектури. Ця архітектура сприяє високоефективній взаємодії між компонентами системи, забезпечуючи при цьому гнучкість та легкість в її функціонуванні. Описана структура системи покликана забезпечувати

оптимальне використання ресурсів та високий рівень взаємодії між користувачами та серверними компонентами.

4.2 Розробка веб-додатку для підсистеми психолога

Для розробки веб додатку було вирішено використовувати реактивний підхід, а саме React. React - це бібліотека JavaScript, яка використовується для створення користувацьких інтерфейсів. Цей фреймворк використовує віртуальний DOM, що дозволяє мінімізувати кількість операцій маніпулювання реальним DOM. Це призводить до покращення продуктивності веб-додатків, особливо при великій кількості компонентів. Також він дозволяє розбити інтерфейс на невеликі і незалежні частини. Це полегшує розробку, тестування і підтримку коду.

Процес взаємодії психолога з додатком розпочинається з етапу реєстрації, якщо обліковий запис було створено, то з аутентифікації. Проведення аутентифікації та створення нового облікового запису на стороні веб-додатку включає послідовні етапи. Під час першого етапу психолог взаємодіє з інтерфейсом додатку, ініціюючи процес реєстрації. У разі створення нового облікового запису користувач надає відповідні персональні дані про себе – логін (рис. 4.2), пароль(рис. 4.3), інформацію про досвід та освіти (рис 4.4).

The screenshot shows the first step of a three-step account creation process. The title is "Create An Account". A progress bar at the top indicates the current step. The form includes the following fields:

- * First Name: Input field containing "Петро"
- * Last Name: Input field containing "Василенко"
- * Email: Input field containing "petrovasylenko@gmail.com"
- * Date of Birthday: Input field containing "1993-12-26" with a calendar icon
- * Country: Dropdown menu showing "Ukraine"

Navigation buttons include "Next" (blue) and "Back to Login" (text link).

Рисунок 4.2 – Форма реєстрації психолога у веб-додатку

Джерело: побудовано автором

The screenshot shows the second step of the account creation process. The title is "Create An Account". The progress bar indicates the current step. The form includes the following fields:

- * Password: Input field with masked characters and a toggle icon
- * Confirm Password: Input field with masked characters and a toggle icon

Navigation buttons include "Previous" (blue), "Next" (blue), and "Back to Login" (text link).

Рисунок 4.3 – Форма вводу та підтвердження паролю

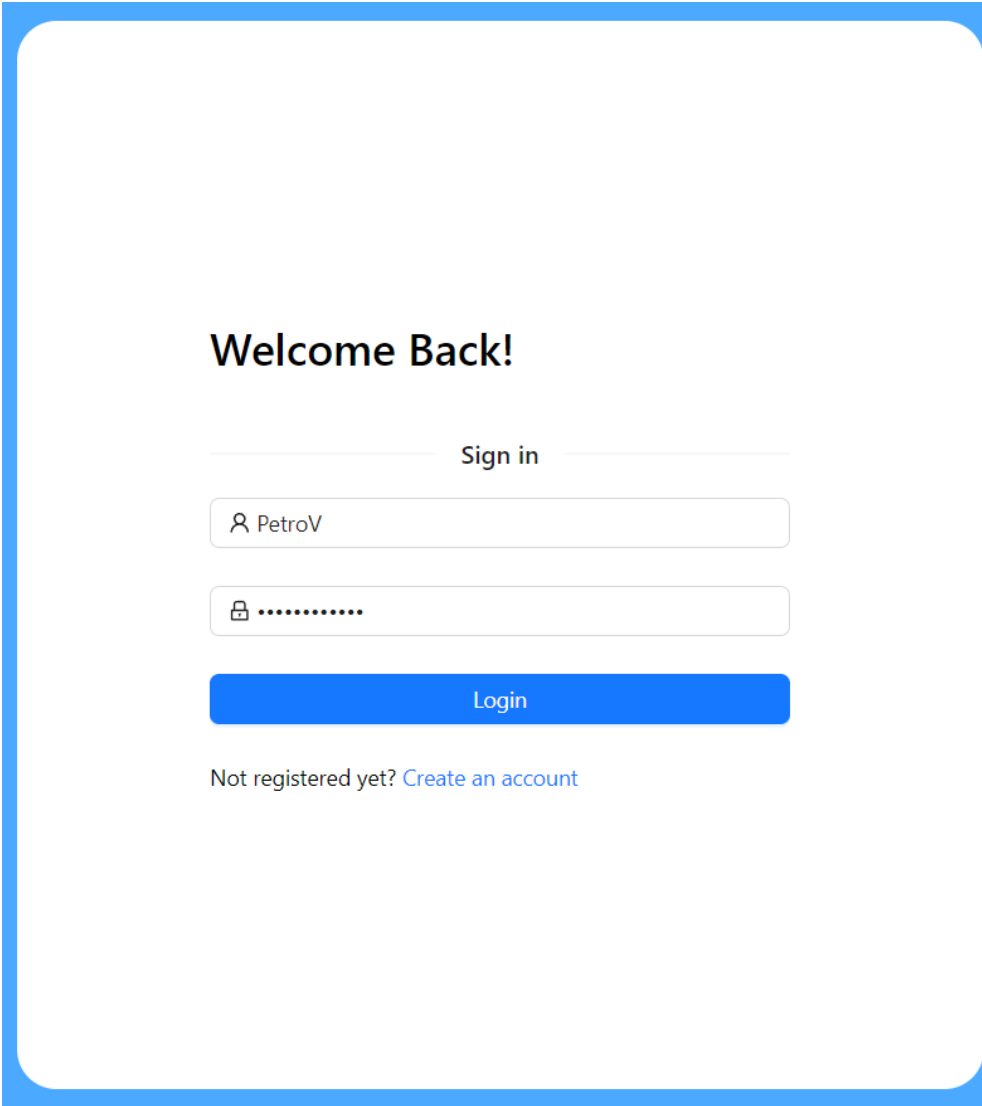
Джерело: побудовано автором

The image shows a web form titled "Create An Account" with a progress indicator showing three steps: "Personal Information", "Confirm Password", and "Questions". The "Personal Information" step is active. It contains two input fields: "Enter your experience" with the text "I very expiriened psychologist" and "Enter your education" with the text "Master degree. SumDU". At the bottom, there are "Previous" and "Submit" buttons, and a "Back to Login" link.

Рисунок 4.4 – Форма вводу персональних даних

Джерело: побудовано автором

Після перевірки сервером на валідність даних реєстрації, в системі буде створений обліковий запис. Використовуючи форму авторизації (рис. 4.5) користувач може заповнити поля логіну та паролю. У випадку, якщо дані були введені вірно, то психолог потрапляє на головку сторінку власного облікового запису.



Welcome Back!

Sign in

Login

Not registered yet? [Create an account](#)

Рисунок 4.5 – Форма авторизації у веб-додатку

Джерело: побудовано автором

Центральним елементом головної сторінки психолога (рис 4.6) є календар подій. В цьому календарі відображається поточні та майбутні заплановані консультації з клієнтами. Також цей календар дає змогу запланувати для психолога зустрічі з потенційними клієнтами відкривши відповідний запис.

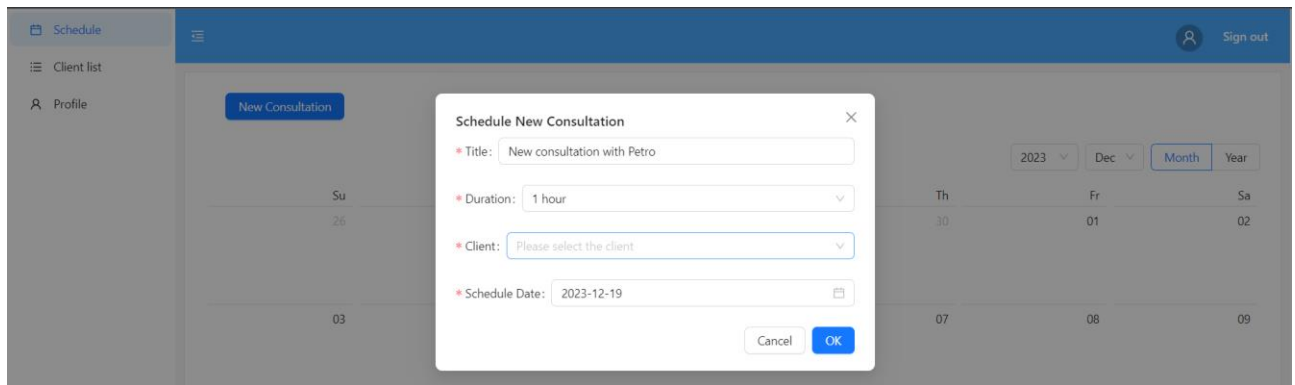


Рисунок 4.6 – Екран планування консультацій у веб-додатку

Джерело: побудовано автором

Психологу також надається можливість редагувати свій особистий профіль для актуального відображення персональних даних про себе, свій фах та освіту (рис. 4.7)

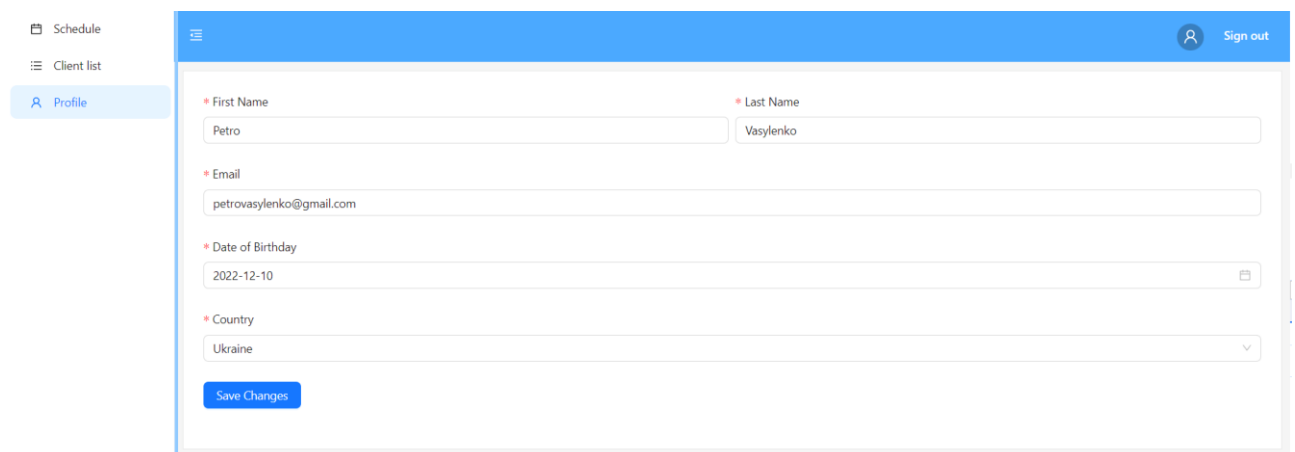


Рисунок 4.7 – Екран редагування профіля у веб-додатку

Джерело: побудовано автором

Основною складовою інформаційної системи є консультація, яка реалізується в рамках даного веб-додатку і полягає у взаємодії між психотерапевтом та клієнтом у віртуальному середовищі. На рисунку 4.8 представлено екран для проведення відео- та аудіо-консультацій, використовуючи сторонній сервіс.

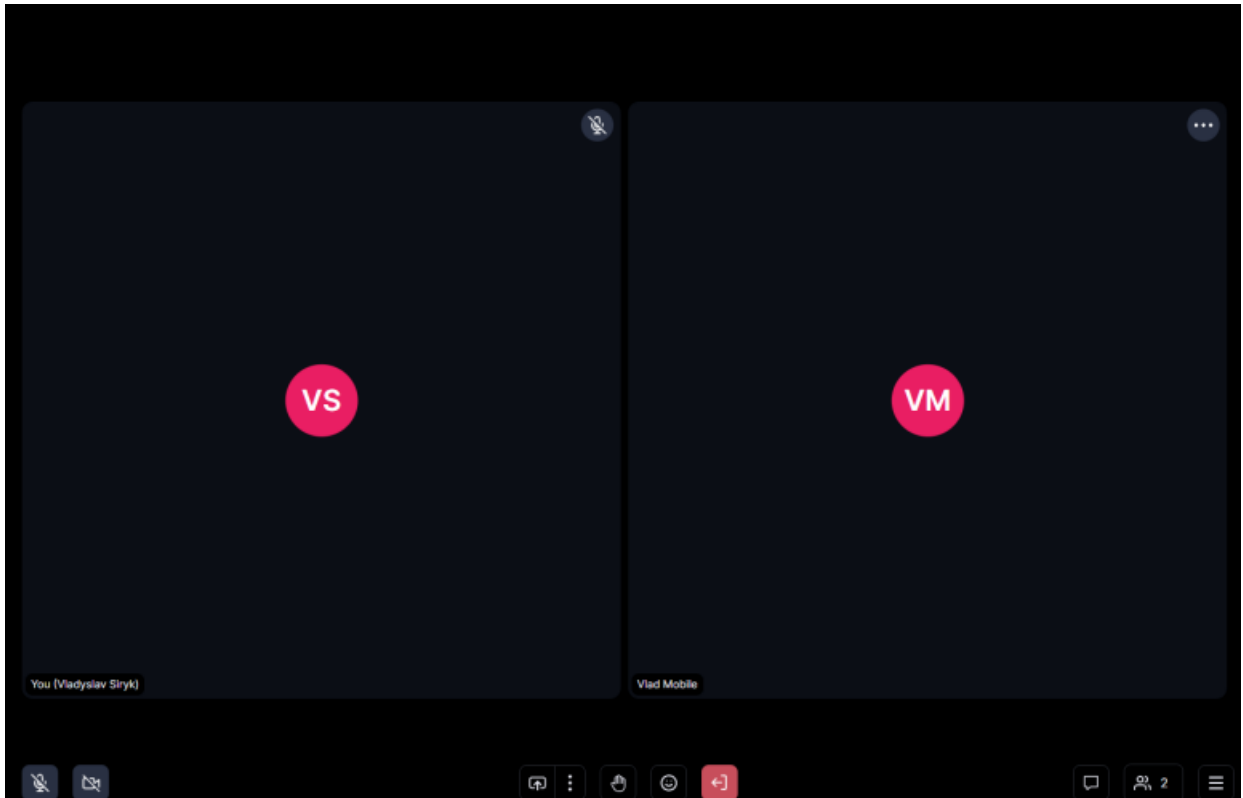


Рисунок 4.8 – Екран відео- та аудіо- консультації у веб-додатку

Джерело: побудовано автором

Надані зображення та опис функціональних можливостей веб-додатку демонструють, як взаємодіють всі компоненти для створення єдиної системи для психолога. Такий підхід сприяє оптимізації використання додатка, роблячи його більш зручним та ефективним для всіх користувачів.

4.3 Розробка серверної частини для підсистеми психолога

З боку серверної частини, то перевагу було надано еко-системі Microsoft, котра має збалансовану інтеграції між всіма своїми сервісами. Для реалізації REST API було використано останню версію ASP.NET Core Web API. Цей фреймворк має потужний інструмент створення контролерів. Клієнтська частина має змогу взаємодіяти за допомогою HTTP запитів і виконувати методи описані в цих контролерах. Зручна система візуалізації методів контролеру, їх вхідних та вихідних параметрів надається за допомогою бібліотеки Swagger.

Приклад відображення методів контролера зображена на рисунку 4.9

The screenshot shows a REST client interface for the 'Psychologists' API. The selected method is POST /api/Psychologists/Register. The parameters section shows two query parameters: 'login' (string) and 'password' (string). The request body is set to 'application/json' and contains the following JSON example value:

```
{
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "firstName": "string",
  "lastName": "string",
  "profileLink": "string",
  "qualifications": [

```

Рисунок 4.9 – POST метод реєстрації психолога

Джерело: побудовано автором

Для підсистеми психолога були використані наступні контролери:

- **AppointmentsController:** Дає можливість психологу створити/редагувати запрошення клієнту на консультацію
- **AuthController :** Створений для авторизації користувачів в системі
- **ChatsController.** – Надає можливість створення аудіо- відео- з'єднання між клієнтом та психологом
- **ConsultationsController.** – Використовується для проведення записів проведення консультацій. Наданні методики, технік та рекомендацій.
- **PsychologistsController.** Потрібен для реєстрації психолога, зміни персональних даних психолога.

REST API реалізує імплементацію ORM (Object-Relational Mapping) системи Entity Framework. ORM системи представляють собою технологію, яка забезпечує взаємодію між об'єктно-орієнтованим програмуванням (ООП) і

реляційними базами даних. Ці системи дозволяють програмістам використовувати об'єктно-орієнтований підхід при роботі з даними в базах даних.

На рисунку 4.10 зображені доменні моделі, котрі були створені для виконання взаємодії зі сторони ООП. За допомогою інструментів міграції на основі цих класів була побудована база даних MS SQL для зберігання та обробки даних різних типів.

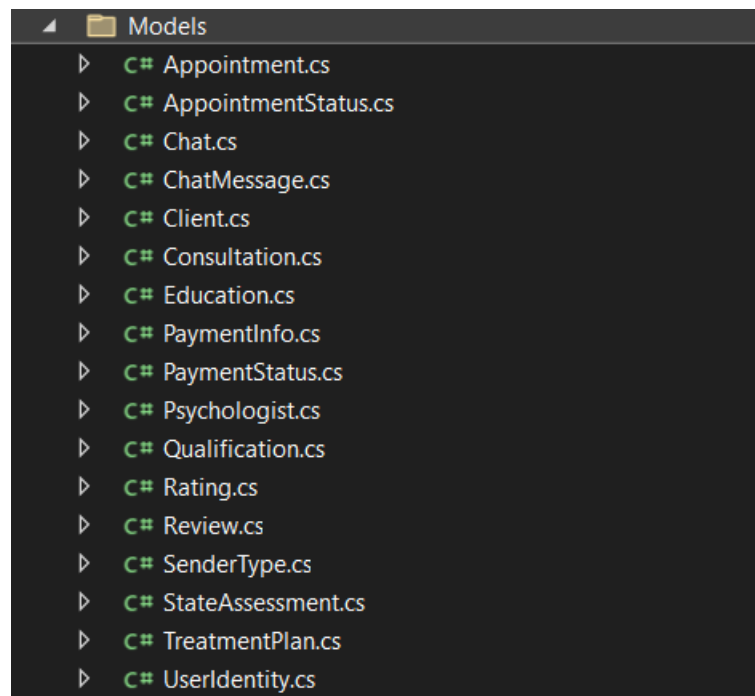


Рисунок 4.10 – Перелік доменних моделей серверної частини

Джерело: побудовано автором

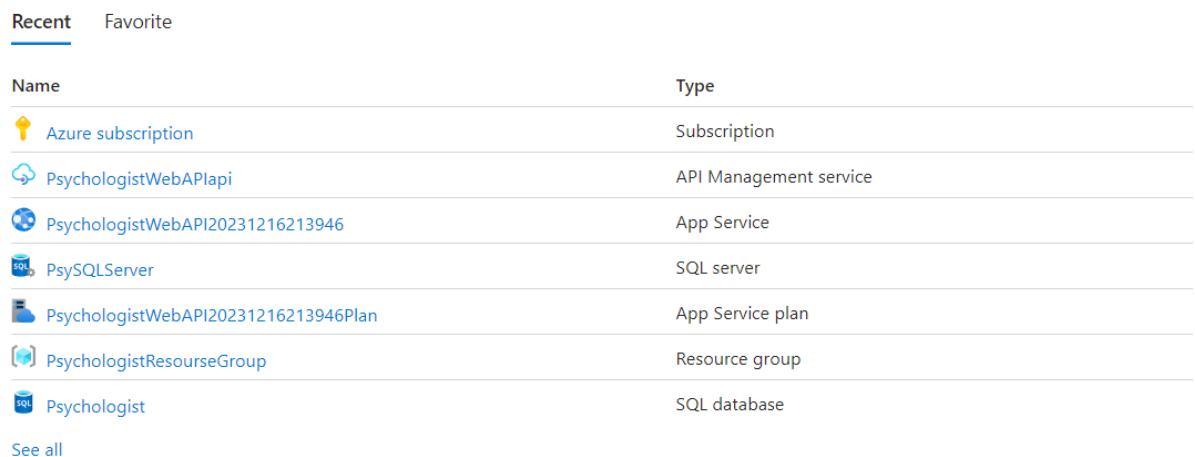
Такий підхід до створення серверної архітектури сприяє зрозумілості коду, забезпечує простоту управління та можливість розширення функціоналу додатку. Завдяки цій архітектурі розробники можуть легко ідентифікувати та змінювати різні компоненти системи, що полегшує процес розробки, тестування та супроводження додатку, роблячи його більш зручним та ефективним.

4.4 Впровадження системи в хмарне середовище

Microsoft Azure - це хмарна платформа, розроблена компанією Microsoft, яка надає широкий спектр хмарних послуг. Azure дозволяє користувачам будувати, розгорнути та керувати додатками та послугами через глобальну мережу даних Microsoft.

Серверна частина додатку була розгорнута в хмарному середовищі Azure, що дозволяє зробити інформаційну систему максимально захищеною, відказостійкою та керувати всіма налаштуваннями система в одному порталі.

Для цього була створена підписка і відповідна група ресурсів для того щоб розмістити всі необхідні сервіси. Перелік всіх створених ресурсів зображений на рисунку 4.11



Name	Type
Azure subscription	Subscription
PsychologistWebAPIapi	API Management service
PsychologistWebAPI20231216213946	App Service
PsySQLServer	SQL server
PsychologistWebAPI20231216213946Plan	App Service plan
PsychologistResourceGroup	Resource group
Psychologist	SQL database

Рисунок 4.11 – Перелік служб розміщених в хмарі

Джерело: побудовано автором

Для впровадження захисту інформації в хмарині було використані налаштування фаєрвола доступу до сервера бази даних. Доступ до бази даних можливий лише для ресурсів розміщених в хмарі (рис 4.12)

Firewall rules

Allow certain public internet IP addresses to access your resource. [Learn more](#)

+ Add your client IPv4 address (91.222.18.97) + Add a firewall rule

Rule name	Start IPv4 address	End IPv4 address
-----------	--------------------	------------------

Exceptions

Allow Azure services and resources to access this server ⓘ

Рисунок 4.12 – Налаштування безпеки доступу до бази даних

Джерело: побудовано автором

4.5 Тестування інформаційної системи

Тестування інформаційної системи є важливою частиною розробки програмного забезпечення. Цей процес дозволяє виявити помилки, визначити відповідність вимогам та забезпечити якість роботи системи перед її впровадженням. Існує кілька видів тестування інформаційних систем.

Результатом тестування інформаційної системи була перевірка виконання функціональних вимог. Всі функціональні вимоги були виконані в повному обсязі, помилок чи невідповідності заявленим вимогам не було знайдено. Також проведено інтеграційне тестування в якому перевірено взаємодію клієнтської частини з серверної. Всі оголошені контракти були реалізовані та інтегровані. Останнім етапом було виконане тестування безпеки. Результатом є перевірка, що всі паролі зберігається в загешованому вигляді. Та доступ до бази мають лише сервіси хмарини.

Загалом виконання тестування системи має відмінну оцінку, що задовольняє оголошеним вимогам до системи.

ВИСНОВКИ

В результаті проведеного дослідження та розробки була створена інформаційна система підтримки надання психологічної допомоги, яка відображає важливі аспекти психологічної допомоги та інформаційної підтримки. Робота охопила широкий спектр аспектів, починаючи від аналізу психологічних потреб користувачів і закінчуючи програмною реалізацією підсистеми психолога.

Важливим результатом є розуміння важливості інформаційних систем у психологічній допомозі та їхньої можливості забезпечувати підтримку користувачів у складних ситуаціях. Розроблена система має потенціал поліпшити доступ до психологічної допомоги та зробити її більш доступною та ефективною.

Основні етапи дослідження та розробки, включаючи аналіз потреб користувачів, етичні та правові аспекти, проектування інтерфейсу та програмну реалізацію, дозволили створити цілісний продукт, який може служити користувачам у покращенні їхнього психологічного стану та наданні підтримки.

Важливо підкреслити, що дана робота не лише дозволила розробити інформаційну систему, але й вивчити психологічні та етичні аспекти цієї сфери, що має велике значення для подальших досліджень і розвитку психологічної допомоги через інформаційні технології.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Використання інформаційних систем у психотерапії / Г. М. Гончарова, Н. С. Петрова // Психотерапія і консультування. - 2019. - № 1. - С. 67-79.
2. Використання інформаційних систем у психотерапії / Г. М. Гончарова, Н. С. Петрова // Психотерапія і консультування. - 2020. - № 1. - С. 67-79.
3. Використання інформаційних систем у психотерапії / Г. М. Гончарова, Н. С. Петрова // Психотерапія і консультування. - 2022. - № 4. - С. 56-68.
4. Використання інформаційних технологій у психологічній підтримці населення / О. М. Кравченко, М. Г. Гончарова // Психологічні дослідження і практика. - 2019. - № 6. - С. 23-36.
5. Використання інформаційних технологій у психологічній підтримці населення / О. М. Кравченко, М. Г. Гончарова // Психологічні дослідження і практика. - 2020. - № 6. - С. 23-36.
6. Ефективність використання інформаційних технологій у роботі психолога / М. М. Ткаченко, Г. М. Лисенко // Психологічні науки. - 2021. - № 5. - С. 34-47.
7. Ефективність інформаційних систем у психологічній діяльності / М. О. Полякова, Т. О. Гончарова // Психологічна інформатика. - 2022. - № 3. - С. 34-47.
8. Ефективність інформаційних систем у психологічній діяльності / М. О. Полякова, Т. О. Гончарова // Психологічна інформатика. - 2017. - № 5. - С. 34-47.
9. Ефективність інформаційних систем у психологічній діяльності / М. О. Полякова, Т. О. Гончарова // Психологічна інформатика. - 2019. - № 4. - С. 23-36.

10. Інноваційні підходи до використання інформаційних систем у психологічній діяльності / Г. М. Поляков, О. О. Гончарова // Психологічні дослідження і практика. - 2018. - № 5. - С. 12-24.
11. Інноваційні підходи до використання інформаційних систем у психологічній діяльності / Г. М. Поляков, О. О. Гончарова // Психологічні дослідження і практика. - 2018. - № 3. - С. 12-24.
12. Інноваційні підходи до використання інформаційних систем у психологічній діяльності / Г. М. Поляков, О. О. Гончарова // Психологічні дослідження і практика. - 2019. - № 3. - С. 12-24.
13. Інтерактивність інформаційних систем для психологічної допомоги / В. О. Гончарова, Г. О. Петренко // Інновації в психології. - 2020. - № 6. - С. 23-36.
14. Інтерактивність інформаційних систем для психологічної допомоги / В. Г. Гончарова, Г. О. Петренко // Інновації в психології. - 2020. - № 1. - С. 56-68.
15. Інтерактивність інформаційних систем у роботі психолога / І. М. Сидоренко, Н. О. Шевченко // Інноваційні підходи в психологічній допомозі. - 2021. - № 4. - С. 12-24.
16. Інтерактивність інформаційних систем у роботі психолога / І. М. Сидоренко, Н. О. Шевченко // Інноваційні підходи в психологічній допомозі. - 2022. - № 1. - С. 12-24.
17. Інтерактивність інформаційних систем у роботі психолога / І. М. Сидоренко, Н. О. Шевченко // Інноваційні підходи в психологічній допомозі. - 2020. - № 5. - С. 12-24.
18. Інформаційна система психологічної підтримки населення: аналіз функціональних можливостей / І. М. Гончаров, Т. Г. Соколова // Психологічний інструментарій. - 2022. - № 1. - С. 45-58.
19. Інформаційні системи в психологічній діяльності: досягнення і виклики / М. С. Кравченко, О. В. Полякова // Психологічні аспекти розвитку. - 2017. - № 4. - С. 45-58.

20. Інформаційні системи в психологічній діяльності: досягнення і виклики / М. С. Кравченко, О. В. Полякова // Психологічні аспекти розвитку. - 2017. - № 2. - С. 45-58.

21. Інформаційні системи в психологічній діяльності: досягнення і виклики / М. С. Кравченко, О. В. Полякова // Психологічні аспекти розвитку. - 2018. - № 1. - С. 45-58.

22. Інформаційні системи для надання психологічної допомоги: досягнення і перспективи / Г. М. Гончаров, О. Г. Ткаченко // Психологічні аспекти розвитку. - 2020. - № 6. - С. 45-58.

23. Інформаційні системи для надання психологічної допомоги: досягнення і перспективи / Г. М. Гончаров, О. Г. Ткаченко // Психологічні аспекти розвитку. - 2021. - № 5. - С. 45-58.

24. Інформаційні системи для психологічного консультування: принципи створення і використання / В. С. Кравченко, О. М. Іванова // Психологічні дослідження. - 2021. - № 2. - С. 45-58.

25. Інформаційні системи для психологічного консультування: принципи створення і використання / В. С. Кравченко, О. М. Іванова // Психологічні дослідження. - 2022. - № 4. - С. 45-58.

26. Інформаційні системи для психологічного консультування: принципи створення і використання / В. С. Кравченко, О. М. Іванова // Психологічні дослідження. - 2021. - № 3. - С. 45-58.

27. Інформаційні технології в психологічному консультуванні / І. О. Ткаченко, В. Г. Гончаров // Психологічний прогрес. - 2020. - № 5. - С. 23-36.

28. Інформаційні технології в психологічному консультуванні / І. О. Ткаченко, В. Г. Гончаров // Психологічний прогрес. - 2021. - № 3. - С. 23-36.

29. Інформаційні технології в психологічному консультуванні / І. О. Ткаченко, В. Г. Гончаров // Психологічний прогрес. - 2020. - № 2. - С. 34-47.

30. Інформаційні технології в роботі психолога: актуальні питання інтеграції / М. С. Ткаченко, Г. В. Лисенко // Психологічний інструментарій. - 2018. - № 2. - С. 45-58.
31. Інформаційні технології в роботі психолога: проблеми та перспективи / О. О. Петрова, М. Г. Гончарова // Психологічні науки. - 2021. - № 6. - С. 23-36.
32. Інформаційні технології у роботі психолога: досвід та перспективи / О. О. Петрова, М. Г. Гончарова // Психологічні дослідження і практика. - 2021. - № 4. - С. 45-58.
33. Психологічні аспекти впровадження інформаційних технологій у психологічну практику / І. М. Гончарова, Т. Г. Соколова // Психологічна допомога і консультування. - 2019. - № 6. - С. 56-68.
34. Психологічні аспекти впровадження інформаційних технологій у психологічну практику / І. В. Гончарова, Т. О. Соколова // Психологічна допомога і консультування. - 2019. - № 2. - С. 45-58.
35. Психологічні аспекти впровадження інформаційних технологій у психологічну практику / І. В. Гончарова, Т. О. Соколова // Психологічна допомога і консультування. - 2020. - № 4. - С. 23-36.
36. Розвиток інформаційних технологій для психологічної допомоги / О. Г. Шевченко, М. О. Ткаченко // Психологічний інструментарій. - 2018. - № 3. - С. 23-36.
37. Behavioral Health EHR/EMR | TherapyNotes [Електронний ресурс] – <https://www.therapynotes.com/>
38. EHR Software for Health & Wellness Professionals | SimplePractice [Електронний ресурс] - <https://www.simplepractice.com/>
39. Mindly | професійна психологічна терапія онлайн [Електронний ресурс] - <https://mindlyspace.com/>

40. Розвиток інформаційних технологій для психологічної допомоги / О. Г. Шевченко, М. О. Ткаченко // Психологічний інструментарій. - 2019. - № 4. - С. 23-36.
41. Розвиток інформаційних технологій для психологічної допомоги / О. Г. Шевченко, М. О. Ткаченко // Психологічний інструментарій. - 2021. - № 2. - С. 67-79.
42. Розробка інформаційних систем для психологічного консультування: досвід і перспективи / Г. О. Поляков, О. О. Гончарова // Психологічні дослідження і практика. - 2021. - № 5. - С. 34-47.
43. Роль інформаційних систем у психологічній реабілітації / Г. С. Сидоренко, О. М. Шевченко // Психологічна допомога і консультування. - 2022. - № 3. - С. 67-79.

ДОДАТОК А. ПЛАНУВАННЯ РОБІТ

А.1 Ідентифікація мети ІТ-проекту

Кваліфікаційна робота магістра націлена на створення підсистеми психолога в інформаційній системі, що забезпечує психологічну підтримку та базується на принципах SMART.

Ідентифікація мети ІТ-проекту за методом SMART разом з таблицею:

Таблиця А. 1 – Деталізація мети методом SMART

Джерело: побудовано автором

Компонент	Опис
Специфічна (Specific)	Розробка інформаційної системи підтримки надання психологічної допомоги.
Міряна (Measurable)	Забезпечення збільшення доступності та якості психологічної допомоги через інформаційну систему.
Досяжна (Achievable)	Реалізація проекту за наявності необхідних ресурсів та фахівців.
Релевантна (Relevant)	Відповідає потребам та вимогам користувачів, а також актуальним тенденціям у сфері психологічної підтримки.
Часово обмежена (Time-bound)	Розробка та впровадження системи повинні бути завершені протягом визначеного графіку, наприклад, протягом 12 місяців.

Проект інформаційної системи для психологічної допомоги дотримується принципів SMART, установлюючи чіткі цілі, вимірювані показники успіху та реалістичні завдання. Проект зосереджений на створенні ефективного інтерфейсу для покращення психічного здоров'я користувачів, при цьому

враховуючи обмеження ресурсів та дотримання стандартів безпеки. Чіткі терміни виконання завдань гарантують своєчасне впровадження проекту. Загалом, проект відзначається систематичним та вимірюваним підходом до досягнення успіху.

А.2 Планування змісту структури робіт інформаційної системи

WBS (рис. А.2) представляє собою ієрархічну структуру, яка може відображати майбутні завдання та цілі, такі як програма, проект або договір. Його призначенням є планування, оцінка, а також визначення та розподіл відповідальності між учасниками.

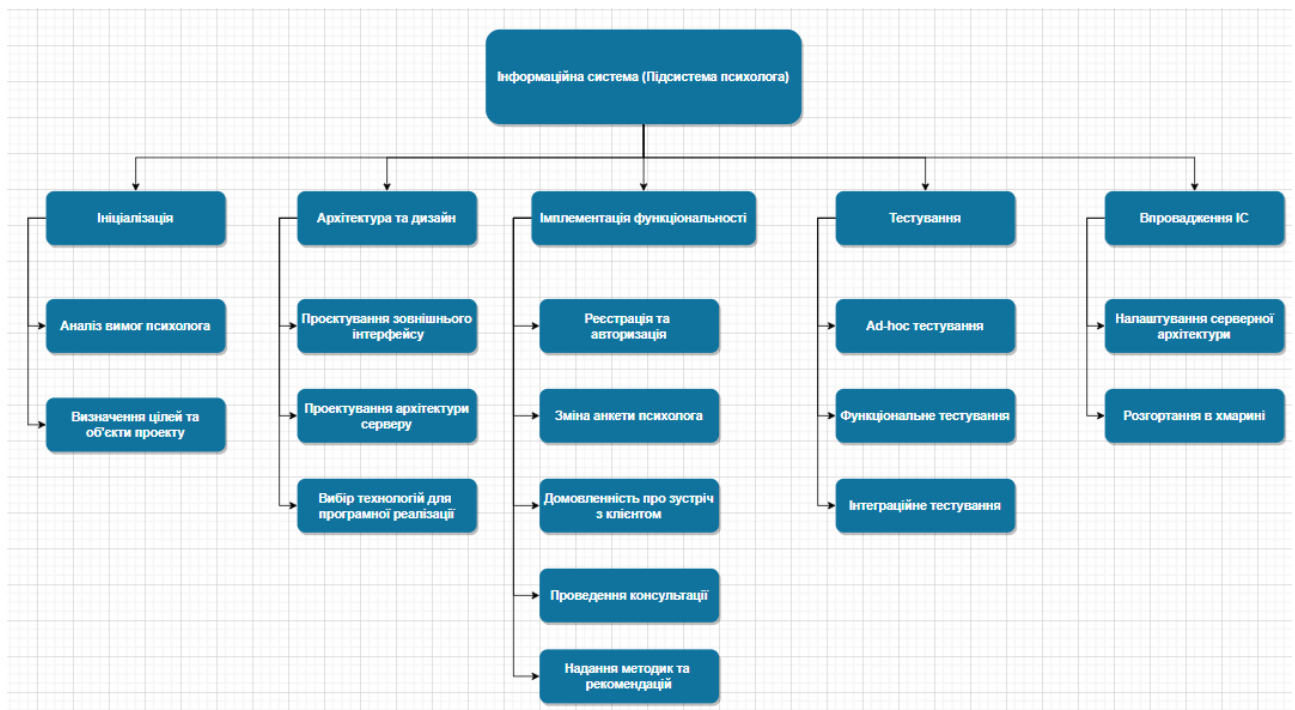


Рисунок А.2 – Діаграма декомпозиції робіт проекту

Джерело: побудовано автором

Після створення WBS наступним етапом є розробка організаційної структури виконавців (OBS), що представляє собою графічне відображення осіб або відповідальних учасників, які беруть участь у реалізації проекту.

Організаційна структура проекту (OBS) показана на рисунку А.3. Таблиця А.2 містить перелік учасників, що беруть участь у виконанні проекту.

Таблиця А.2 – Виконавці проекту

Джерело: побудовано автором

Роль	ПІБ	Проектна роль
Менеджер	Шапка С.О.	Встановлює стратегію та організовує взаємодію між учасниками проекту.
Розробник	Шапка С.О.	Відповідає за розробку програмного забезпечення.
Тестувальник	Шапка С.О.	Виявляє помилки в програмному та апаратному забезпеченні.
Архітектор	Шапка С.О., Сірик В.Ю., Бойко О.В.	Визначає архітектурні принципи та контролює дотримання стандартів.
Аналітик	Шапка С.О.	Визначає вимоги до програмного забезпечення.
Дизайнер	Шапка С.О.	Розробляє інтерфейс користувача, забезпечує відповідність дизайну.
Серверний інженер	Шапка С.О.	Відповідає за встановлення, налаштування та технічне обслуговування серверного обладнання та програмного забезпечення.

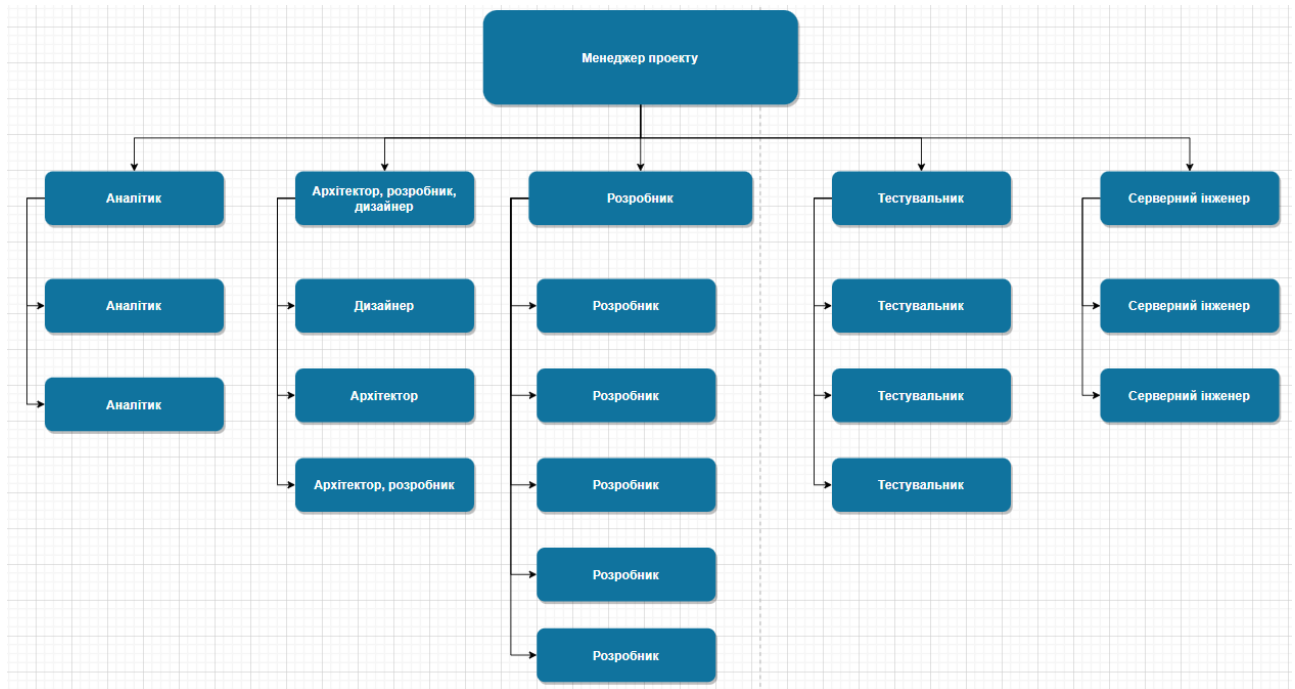


Рисунок А.3 – OBS-структура робіт проекту

Джерело: побудовано автором

А.3 Побудова календарного графіку виконання інформаційної системи

У даному розділі представлено діаграму Ганта для нашого проекту, яка є важливим інструментом для візуалізації стратегії впровадження та структури проекту. Цей графічний інструмент допомагає не лише в ілюстрації послідовності та тривалості завдань, але й у визначенні критичних шляхів, розподілі ресурсів та уникненні можливих затримок у виконанні проекту.

Кожен сегмент діаграми відображає конкретне завдання чи підзадачу і розташований вертикально. Початок, кінець та тривалість кожного сегмента на шкалі часу відповідають відповідно початку, закінченню та тривалості відповідного завдання. Деякі діаграми також демонструють взаємозв'язок між різними завданнями.

Рисунок А.5 із діаграмою Ганта надає інформацію про початок проекту 2 жовтня 2023 року. Протягом виконання проекту передбачалося витрачання по 3 години на день, включаючи вихідні. З діаграми видно, що робота над різними етапами відбувалася послідовно, за винятком етапу контролю та моніторингу, де процеси відбувалися паралельно.

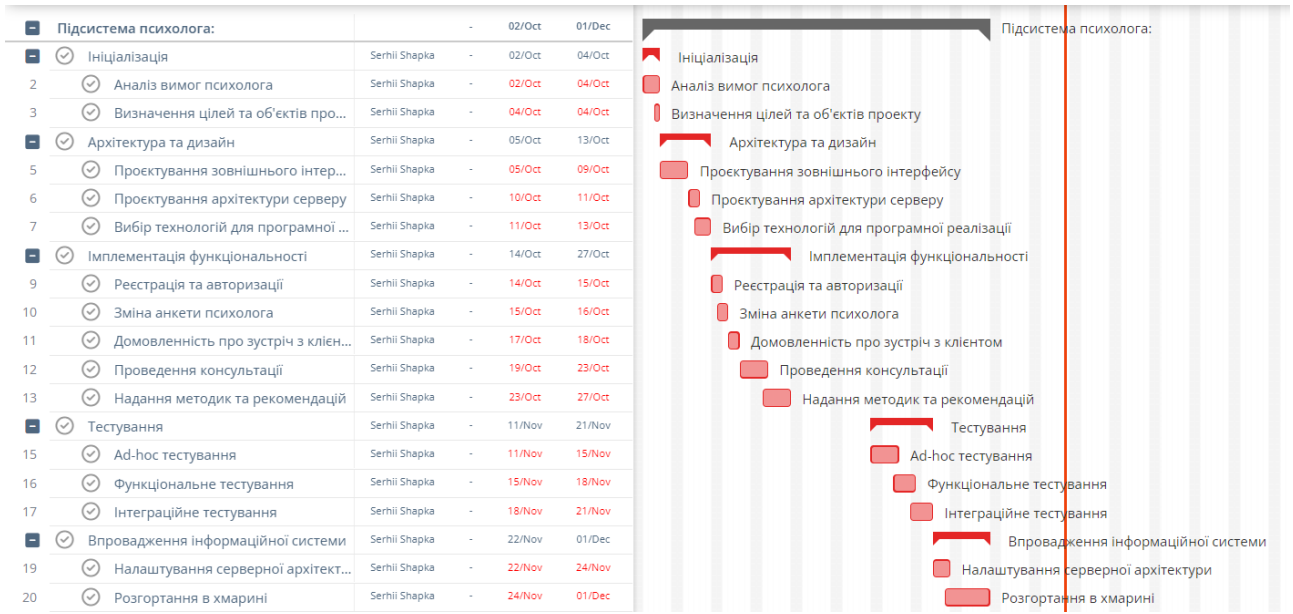


Рис А.5 – Діаграма Ганта

Джерело: побудовано автором

А.4 Планування ризиків проекту

Проведення якісної оцінки ризиків допомагає визначити ті, які є найбільш критичними та потребують негайних заходів для їх контролю. Оцінка важливості ризику визначає, які конкретні заходи реагування слід прийняти. Подальший етап включає кількісне оцінювання ризиків, яке може проводитися паралельно з якісною оцінкою чи окремо, залежно від вимог проекту.

Для визначення значень можливих ризиків була використана методика експертної оцінки. За допомогою цих оцінок можна визначити ранг ризику за формулою $R = P * L$, де:

- R – ранг ризику;
- P – ймовірність виникнення;

- L – ступінь впливу.

Шкала оцінки ризику відповідає емпіричній шкалі оцінювання ризику:

- 5 балів - критичний ризик (0,81 - 1);
- 4 бали - максимальний ризик (0,61 - 0,8);
- 3 бали - високий ризик (0,41 - 0,6);
- 2 бали - нормальний ризик (0,31 - 0,4);
- 1 бал - малий ризик (0 - 0,3).

У таблиці А.3 міститься перелік ризиків, ідентифікованих для нашого проекту.

Таблиця А.3 – Ризики проекту

Джерело: побудовано автором

№ ризику	Назва та опис ризику
1	Небезпека втрати конфіденційності. Даний ризик існує, особливо у випадку психологічної інформації про користувачів, яка є надзвичайно чутливою. Без забезпечення високого рівня безпеки інформація може потрапити у руки третіх осіб, що може призвести до порушення конфіденційності.
2	Невірне та нечітке розуміння вимог користувачів. Може призвести до того, що підсистема не відповідатиме потребам користувачів, або буде занадто складною та витратною у розробці та обслуговуванні.
3	Етичне питання. Збір та обробка психологічної інформації може викликати етичні питання. Недостатня увага до етичних аспектів може мати негативні наслідки для користувачів.
4	Проблема із доступом до сторонніх сервісів. В системі планується використання зовнішніх сервісів, які не можемо контролювати.

Результати оцінки ризиків представлені в таблиці А.4

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

Джерело: побудовано автором

№ ризику	Назва ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Небезпека втрати конфіденційності	0,1	0,8	0,08
2	Невірне та нечітке розуміння вимог користувачів	0,6	0,4	0,24
3	Етичне питання	0,2	0,9	0,18
4	Проблема із доступом до сторонніх сервісів	0,2	0,7	0,14

Отримані результати оцінки ризиків використовуються для формулювання планів реагування на ці ризики. У нашій інформаційній системі розроблені плани реагування на ризики, спрямовані на зменшення ймовірності виникнення ризику або зменшення його впливу, які наведено нижче (таблиця А.5).

Таблиця А.5 - Стратегії реагування на ризики

Джерело: побудовано автором

Назва ризику	Ранг ризику	План реагування
Небезпека втрати конфіденційності	0,08	1 Розробити політику конфіденційності 2 Регулювання надійності паролю 3 Двофакторна аутентифікація
Невірне та нечітке розуміння вимог користувачів	0,24	1 Створити механізм зворотного зв'язку з користувачами
Етичне питання	0,18	1 Впровадження методів анонімності та підтримки кодексу психолога
Проблема із доступом до сторонніх сервісів	0,14	1 Звернення до технічної підтримки сторонніх сервісів.

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ

Лістинг файлу Program.cs

```

using DB_Core;
using Microsoft.EntityFrameworkCore;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Authentication;
using System.Text;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.IdentityModel.Tokens;
using Microsoft.Extensions.DependencyInjection;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
string connection = builder.Configuration.GetConnectionString("DefaultConnection");
builder.Services.AddDbContext<PsyDBContext>(options =>
options.UseSqlServer(connection));

var keyValue = "";
var key = Encoding.ASCII.GetBytes(keyValue);

builder.Services.AddAuthentication(options =>
{
    options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
    options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
})
.AddJwtBearer(options =>
{
    options.RequireHttpsMetadata = false;
    options.SaveToken = true;
    options.TokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(key),
        ValidateIssuer = false,
        ValidateAudience = false
    };
});

builder.Services.AddScoped<TokenService>(provider =>
{
    return new TokenService(keyValue);
});

var app = builder.Build();

// Configure the HTTP request pipeline.

app.UseSwagger();
app.UseSwaggerUI();
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();

```

Лістинг файлу PsyDBContext.cs

```

using DB_Core.Models;
using Microsoft.EntityFrameworkCore;

namespace DB_Core
{
    public class PsyDBContext : DbContext
    {
        public PsyDBContext(DbContextOptions<PsyDBContext> options) : base(options)
        { }

        public DbSet<Psychologist> Psychologists { get; set; }
        public DbSet<Consultation> Consultations { get; set; }
        public DbSet<StateAssessment> StateAssessments { get; set; }
        public DbSet<Appointment> Appointments { get; set; }
        public DbSet<TreatmentPlan> TreatmentPlans { get; set; }
        public DbSet<PaymentInfo> PaymentInfos { get; set; }
        public DbSet<Client> Clients { get; set; }
        public DbSet<Chat> Chats { get; set; }
        public DbSet<ChatMessage> ChatMessages { get; set; }
        public DbSet<Review> Reviews { get; set; }
        public DbSet<Qualification> Qualifications { get; set; }
        public DbSet<Education> Educations { get; set; }
        public DbSet<UserIdentity> UserIdentities { get; set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Appointment>().
                HasOne(x => x.Client).
                WithMany(y => y.Appointments).
                HasForeignKey(x => x.ClientId).
                OnDelete(DeleteBehavior.NoAction);

            modelBuilder.Entity<Appointment>().
                HasOne(x => x.Psychologist).
                WithMany(y => y.Appointments).
                HasForeignKey(x => x.PsychologistId).
                OnDelete(DeleteBehavior.NoAction);

            modelBuilder.Entity<Chat>().
                HasOne(x => x.Client).
                WithMany(y => y.Chats).
                HasForeignKey(x => x.ClientId).
                OnDelete(DeleteBehavior.NoAction);

            modelBuilder.Entity<Chat>().
                HasOne(x => x.Psychologist).
                WithMany(y => y.Chats).
                HasForeignKey(x => x.PsychologistId).
                OnDelete(DeleteBehavior.NoAction);

            modelBuilder.Entity<Review>().
                HasOne(x => x.Client).
                WithMany(y => y.Reviews).
                HasForeignKey(x => x.ClientId).
                OnDelete(DeleteBehavior.NoAction);

            modelBuilder.Entity<Review>().
                HasOne(x => x.Psychologist).
                WithMany(y => y.Reviews).
                HasForeignKey(x => x.PsychologistId).
        }
    }
}

```

```

        OnDelete(DeleteBehavior.NoAction);

        modelBuilder.Entity<Consultation>().
            HasOne(x => x.Client).
            WithMany(y => y.Consultations).
            HasForeignKey(x => x.ClientId).
            OnDelete(DeleteBehavior.NoAction);

        modelBuilder.Entity<Consultation>().
            HasOne(x => x.Psychologist).
            WithMany(y => y.Consultations).
            HasForeignKey(x => x.PsychologistId).
            OnDelete(DeleteBehavior.NoAction);
    }
}
}

```

Лістинг файлу LoginModel.cs

```

public class LoginModel
{
    public string Login { get; set; }
    public string Password { get; set; }
} public class LoginModel
{
    public string Login { get; set; }
    public string Password { get; set; }
}

```

Лістинг файлу StringExtension.cs

```

using System.Security.Cryptography;
using System.Text;

namespace DB_Core.Helper
{
    public static class StringExtension
    {
        public static string GetMd5Hash(this string input)
        {
            using MD5 md5Hash = MD5.Create();
            byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(input));
            StringBuilder sBuilder = new StringBuilder();

            for (int i = 0; i < data.Length; i++)
            {
                sBuilder.Append(data[i].ToString("x2"));
            }
            return sBuilder.ToString();
        }
    }
}

```

Лістинг файлу TokenService.cs

```

using Microsoft.IdentityModel.Tokens;
using System;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;

```

```

public class TokenService
{
    private readonly string _secretKey;

    public TokenService(string secretKey)
    {
        _secretKey = secretKey;
    }

    public string GenerateToken(string username, Guid id, bool isPsychologist)
    {
        var tokenHandler = new JwtSecurityTokenHandler();
        var key = Encoding.ASCII.GetBytes(_secretKey);
        var tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = new ClaimsIdentity(new Claim[]
            {
                new Claim(ClaimTypes.Name, username),
                new Claim("Id", id.ToString()),
                new Claim(ClaimTypes.Actor, isPsychologist.ToString())
            }),
            Expires = DateTime.UtcNow.AddDays(1),
            SigningCredentials = new SigningCredentials(new
SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256Signature)
        };
        var token = tokenHandler.CreateToken(tokenDescriptor);
        return tokenHandler.WriteToken(token);
    }
}

```

Лістинг файлу Psychologist.cs

```

using System.ComponentModel.DataAnnotations;

namespace DB_Core.Models
{
    public class Psychologist
    {
        public Guid Id { get; init; }

        public required string FirstName { get; set; }

        public required string LastName { get; set; }

        public string? ProfileLink { get; set; }

        public required IList<Qualification> Qualifications { get; set; }

        public required string Description { get; set; }

        public required IList<Education> Educations { get; set; }

        public bool IsVerified { get; set; }

        public virtual IList<Consultation> Consultations { get; set; } = new
List<Consultation>();

        public virtual IList<Appointment> Appointments { get; set; } = new
List<Appointment>();

        public virtual IList<Chat> Chats { get; set; } = new List<Chat>();
    }
}

```

```

        public virtual IList<Review> Reviews { get; set; } = new List<Review>();
    }
}

```

Лістинг файлу Appointment.cs

```

namespace DB_Core.Models
{
    public class Appointment
    {
        public Guid Id { get; init; }

        public AppointmentStatus Status { get; set; }

        public DateTime Date { get; set; }

        public Guid ClientId { get; set; }
        public Guid PsychologistId { get; set; }

        public required virtual Client Client { get; set; }

        public required virtual Psychologist Psychologist { get; set; }
    }
}

```

Лістинг файлу Consultation.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DB_Core.Models
{
    public class Consultation
    {
        public Guid Id { get; set; }

        public DateTime Date { get; set; }

        public string? Notes { get; set; }

        public Guid ClientId { get; set; }
        public Guid PsychologistId { get; set; }

        public virtual required Client Client { get; set; }

        public virtual required Psychologist Psychologist { get; set; }

        public virtual StateAssessment? StateAssessment { get; set; }

        public virtual PaymentInfo? PaymentInfo { get; set; }
    }
}

```

Лістинг файлу Education.cs

```

namespace DB_Core.Models
{
    public class Education
    {
        public Guid Id { get; set; }

        public required string Specialization { get; set; }

        public required string UniversityName { get; set; }

        public DateTime StartDate { get; set; }

        public DateTime? EndDate { get; set; }

        public string? CertificateLink { get; set; }
    }
}

```

Лістинг файлу Review.cs

```

namespace DB_Core.Models
{
    public class Review
    {
        public Guid Id { get; set; }

        public Rating Rating { get; set; }

        public DateTime CreatedAt { get; set; }

        public required string Comment { get; set; }

        public Guid ClientId { get; set; }

        public Guid PsychologistId { get; set; }

        public required Psychologist Psychologist { get; set; }

        public required Client Client { get; set; }
    }
}

```

Лістинг файлу AppointmentsController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using DB_Core.Models;
using Microsoft.AspNetCore.Authorization;
using DB_Core;

namespace Psychologist_Web_API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]

```

```

[Authorize]
public class AppointmentsController : ControllerBase
{
    private readonly PsyDBContext _context;

    public AppointmentsController(PsyDBContext context)
    {
        _context = context;
    }

    // GET: api/Appointments
    [HttpGet]
    public async Task<ActionResult<IEnumerable<Appointment>>> GetAppointment()
    {
        string login = User.Identity.Name;
        var user = await _context.UserIdentities.FirstOrDefaultAsync(x =>
x.Login == login);
        if (user == null)
        {
            return BadRequest("User not found");
        }

        if (user.IsPsychologist)
        {
            return _context.Appointments.Where(x => x.PsychologistId ==
user.UserId).ToList();
        }

        return _context.Appointments.Where(x => x.ClientId ==
user.UserId).ToList();
    }

    // GET: api/Appointments/5
    [HttpGet("{id}")]
    public async Task<ActionResult<Appointment>> GetAppointment(Guid id)
    {
        var appointment = await _context.Appointments.FindAsync(id);

        if (appointment == null)
        {
            return NotFound();
        }

        return appointment;
    }

    // PUT: api/Appointments/5
    // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPut("{id}")]
    public async Task<IActionResult> PutAppointment(Guid id, Appointment
appointment)
    {
        if (id != appointment.Id)
        {
            return BadRequest();
        }

        _context.Entry(appointment).State = EntityState.Modified;

        try
        {
            await _context.SaveChangesAsync();
        }
    }
}

```



```

    }
    catch (DbUpdateConcurrencyException)
    {
        if (!AppointmentExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return NoContent();
}

// POST: api/Appointments
// To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPost]
public async Task<ActionResult<Appointment>> PostAppointment(Appointment
appointment)
{
    _context.Appointments.Add(appointment);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetAppointment", new { id = appointment.Id },
appointment);
}

// DELETE: api/Appointments/5
[HttpDelete("{id}")]
public async Task<IActionResult> DeleteAppointment(Guid id)
{
    var appointment = await _context.Appointments.FindAsync(id);
    if (appointment == null)
    {
        return NotFound();
    }

    _context.Appointments.Remove(appointment);
    await _context.SaveChangesAsync();

    return NoContent();
}

private bool AppointmentExists(Guid id)
{
    return _context.Appointments.Any(e => e.Id == id);
}
}
}

```

Лістинг файлу AuthController .cs

```

using DB_Core;
using DB_Core.Helper;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;

[ApiController]
[Route("api/[controller]")]
[AllowAnonymous]

```

```

public class AuthController : ControllerBase
{
    private readonly TokenService _tokenService;
    private readonly PsyDBContext _dbContext;

    public AuthController(TokenService tokenService, PsyDBContext dbContext)
    {
        _tokenService = tokenService;
        _dbContext = dbContext;
    }

    [HttpPost("Login")]
    public IActionResult Login([FromBody] LoginModel loginModel)
    {
        var user = _dbContext.UserIdentities.FirstOrDefault(x => x.Login == loginModel.Login); ;
        if (user == null)
        {
            return Unauthorized("Login not found");
        }

        if (loginModel.Password.GetMd5Hash() != user.Password)
        {
            return Unauthorized("Password is not correct");
        }

        // Generate a token
        var token = _tokenService.GenerateToken(loginModel.Login, user.UserId, user.IsPsychologist);

        return Ok(new { Token = token, expiresIn = 1440});
    }
}

```

Лістинг файлу ConsultationsController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using DB_Core.Models;
using DB_Core;

namespace Psychologist_Web_API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ConsultationsController : ControllerBase
    {
        private readonly PsyDBContext _context;

        public ConsultationsController(PsyDBContext context)
        {
            _context = context;
        }

        // GET: api/Consultations
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Consultation>>> GetConsultation()

```

```

    {
        string login = User.Identity.Name;
        var user = await _context.UserIdentities.FirstOrDefaultAsync(x =>
x.Login == login);
        if (user == null)
        {
            return BadRequest("User not found");
        }

        if (user.IsPsychologist)
        {
            return _context.Consultations.Where(x => x.PsychologistId ==
user.UserId).ToList();
        }

        return _context.Consultations.Where(x => x.ClientId ==
user.UserId).ToList();
    }

    // GET: api/Consultations/5
    [HttpGet("{id}")]
    public async Task<ActionResult<Consultation>> GetConsultation(Guid id)
    {
        var consultation = await _context.Consultations.FindAsync(id);

        if (consultation == null)
        {
            return NotFound();
        }

        return consultation;
    }

    // PUT: api/Consultations/5
    // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPut("{id}")]
    public async Task<IActionResult> PutConsultation(Guid id, Consultation
consultation)
    {
        if (id != consultation.Id)
        {
            return BadRequest();
        }

        _context.Entry(consultation).State = EntityState.Modified;

        try
        {
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!ConsultationExists(id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
    }

```

```

        return NoContent();
    }

    // POST: api/Consultations
    // To protect from overposting attacks, see
    https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPost]
    public async Task<ActionResult<Consultation>> PostConsultation(Consultation
consultation)
    {
        _context.Consultations.Add(consultation);
        await _context.SaveChangesAsync();

        return CreatedAtAction("GetConsultation", new { id = consultation.Id },
consultation);
    }

    // DELETE: api/Consultations/5
    [HttpDelete("{id}")]
    public async Task<IAActionResult> DeleteConsultation(Guid id)
    {
        var consultation = await _context.Consultations.FindAsync(id);
        if (consultation == null)
        {
            return NotFound();
        }

        _context.Consultations.Remove(consultation);
        await _context.SaveChangesAsync();

        return NoContent();
    }

    private bool ConsultationExists(Guid id)
    {
        return _context.Consultations.Any(e => e.Id == id);
    }
}

```

Лістинг файлу PsychologistsController.cs

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using DB_Core.Models;
using Microsoft.AspNetCore.Authorization;
using DB_Core.Helper;

namespace DB_Core.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    [Authorize]
    public class PsychologistsController : ControllerBase
    {
        private readonly PsyDBContext _context;

        public PsychologistsController(PsyDBContext context)
        {
            _context = context;
        }

        // GET: api/Psychologists

```

```

[HttpGet]
public async Task<ActionResult<IEnumerable<Psychologist>>>
GetPsychologists()
{
    return await _context.Psychologists.ToListAsync();
}

// GET: api/Psychologists/5
[HttpGet("{id}")]
public async Task<ActionResult<Psychologist>> GetPsychologist(Guid id)
{
    var psychologist = await _context.Psychologists.FindAsync(id);

    if (psychologist == null)
    {
        return NotFound();
    }

    return psychologist;
}

// PUT: api/Psychologists/5
// To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPut("{id}")]
public async Task<IActionResult> PutPsychologist(Guid id, Psychologist
psychologist)
{
    if (id != psychologist.Id)
    {
        return BadRequest();
    }

    _context.Entry(psychologist).State = EntityState.Modified;

    try
    {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!PsychologistExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return NoContent();
}

// POST: api/Psychologists
// To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPost("Register")]
[AllowAnonymous]
public async Task<ActionResult<Psychologist>> Register([FromBody]
Psychologist psychologist, string login, string password)
{
    if (ModelState.IsValid)
    {

```

```

        if (_context.UserIdentities.Any(x => x.Login == login))
        {
            return BadRequest($"User with login {login} already exists");
        }

        _context.Psychologists.Add(psychologist);

        UserIdentity userIdentity = new UserIdentity
        {
            Id = Guid.NewGuid(),
            IsPsychologist = true,
            Login = login,
            Password = password.GetMd5Hash(),
            UserId = psychologist.Id
        };

        _context.UserIdentities.Add(userIdentity);

        await _context.SaveChangesAsync();

        return CreatedAtAction("GetPsychologist", new { id = psychologist.Id
    }, psychologist);
    }
    else
    {
        return BadRequest(ModelState);
    }
}

private bool PsychologistExists(Guid id)
{
    return _context.Psychologists.Any(e => e.Id == id);
}
}
}

```