

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ
ДЛЯ ЗАПИСУ ДО МЕДИЧНИХ ФАХІВЦІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF AN APPLICATION
FOR MAKING APPOINTMENTS WITH MEDICAL SPECIALISTS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-142м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

МОЛЧАНОВ О.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ТЕРЕЩЕНКО Е.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

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

МОЛЧАНОВА Олега Валентиновича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація застосунку для запису до медичних фахівців. Research and Software Implementation of an application for making appointments with medical specialists.

керівник проєкту (роботи) к.т.н., доцент, ФЕДОРОНЧАК Тетяна Василівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 14 ” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 15 грудня 2023 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Опис мобільного застосунку. 4. Експлуатація, тестування та експериментальне дослідження мобільного застосунку.

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ФЕДОРОНЧАК Т.В., доцент		
Нормоконтролер	ДЕЙНЕГА Л.Ю., старший викладач		

7. Дата видачі завдання “ 27 ” вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	3 тиждень	Розділ 2
4	Розробка архітектури програми.	4 тиждень	Розділ 3
5	Розробка мобільного застосунку.	5-6 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	7-8 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	9-10 тиждень	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Олег МОЛЧАНОВ
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна ФЕДОРОНЧАК
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
95 с., 3 табл., 25 рис., 3 дод., 10 джерел.

FLUTTER, МОВА ПРОГРАМУВАННЯ DART, СИСТЕМА
ЗБЕРЕЖЕННЯ ДАНИХ FIREBASE, FIRESTORE, DIO.

Об'єкт дослідження – запис на прийом до медичних фахівців.

Предмет дослідження – мобільний застосунок для запису до медичних фахівців.

Мета роботи – дослідження та розробка мобільного застосунку для зручного та ефективного запису до медичних фахівців.

Матеріали, методи та технічні засоби: фреймворк Flutter, мова програмування Dart, система збереження даних Firebase, база даних Firestore, бібліотека для HTTP-запитів Dio.

Результати. Розроблено мобільний застосунок для зручного онлайн-запису до медичних фахівців, яке використовує мову програмування Dart та фреймворк Flutter. В системі використовуються Firebase та Firestore для збереження та оновлення даних. Для взаємодії з сервером використовується бібліотека Dio. Управління станом реалізовано за допомогою патерну BLoC.

Практична цінність роботи полягає у створенні зручного інструменту для користувачів, щоб швидко та ефективно записатися до медичних фахівців.

Висновки. Виконано проєктування та реалізацію мобільного застосунку для запису до медичних фахівців. Здійснено тестування розробленого мобільного застосунку, щоб переконатися в його надійності та ефективності.

Галузь використання – медична сфера та зручне онлайн-записування до медичних фахівців.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 95 pages, 25 figures, 3 tables, 3 appendixes, 10 sources.

FLUTTER, DART PROGRAMMING LANGUAGE, FIREBASE DATA STORAGE SYSTEM, FIRESTORE, DIO.

Research Object – appointment scheduling for medical professionals..

Research Subject – software tools for online scheduling with medical professionals.

The purpose of this work is to research and develop software for convenient and efficient scheduling with medical professionals.

Materials, Methods, and Technical Means: Flutter framework, Dart programming language, Firebase data storage system, Firestore database, Dio library for HTTP requests.

Results: Software has been developed for convenient online scheduling with medical professionals, utilizing the Dart programming language and Flutter framework. The system utilizes Firebase and Firestore for data storage and updates. The Dio library is employed for server communication, and state management is implemented using the BLoC pattern.

Practical value of the work lies in creating a user-friendly tool for quick and efficient appointment scheduling with medical professionals.

Conclusions: Design and implementation of software for online scheduling with medical professionals have been accomplished. Testing of the developed application has been conducted to ensure its reliability and efficiency.

Field of application – the medical sector and convenient online scheduling with medical professionals.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про онлайн-запис до медичних фахівців	11
1.2 Дослідження мобільних застосунків для запису до медичних фахівців	15
1.2.1 Мобільний застосунок Helsi	15
1.2.2 Мобільний застосунок MeetDoc	18
1.2.3 Мобільний застосунок Doc.ua	21
1.3 Висновки за розділом 1	26
2 Матеріали і методи	27
2.1 Вибір мови програмування та інших технологій розробки	27
2.2 Вибір фреймворку для створення мобільного застосунку	32
2.3 Розробка моделі взаємодії користувача з мобільним застосунком для запису до медичних фахівців	36
2.4 Серверна частина застосунку	39
2.5 Висновки за розділом 2	41
3 Опис мобільного застосунку	42
3.1 Структура мобільного застосунку запису до медичних фахівців	42
3.2 Функціонування мобільного застосунку запису до медичних фахівців	43
3.3 Висновки за розділом 3	45
4 Експлуатація, тестування та експериментальне дослідження мобільного застосунку	46
4.1 Призначення й умови застосування мобільного застосунку	46
4.2 Характеристики мобільного застосунку	47
4.3 Опис роботи з мобільним застосунком та результати тестування	47
4.4 Висновки за розділом 4	61

Висновки	63
Перелік джерел посилань	65
Додаток А Технічне завдання	66
Додаток Б Текст програми	70
Додаток В Слайди презентації	88

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- API – Application Programming Interface;
- ПЗ – Програмне забезпечення;
- JSON – JavaScript Object Notation;
- UTF-8 – Unicode Transformation Format - 8-bit;
- HTTP – Hypertext Transfer Protocol;
- HTML – Hypertext Markup Language;
- CSS – Cascading Style Sheets;
- DOM – Document Object Model;
- ООП – Об'єктно-Орієнтоване Програмування.

ВСТУП

У сучасному світі, де стрімко розвиваються технології, важко переоцінити вплив програмного забезпечення на всі сфери нашого життя. Однією з важливих галузей, де це особливо суттєво, є сфера охорони здоров'я. Запит на швидкий та ефективний доступ до медичних послуг постійно зростає, і технології відіграють визначальну роль у вдосконаленні цього процесу.

Проблема доступу до медичних послуг і, зокрема, можливості швидкого та зручного запису до медичних фахівців визначається рядом чинників, які включають в себе зростання попиту на медичні послуги, обмеженість ресурсів та часу як самого медичного персоналу, так і самих пацієнтів. У світлі цих викликів важливо розвивати та впроваджувати сучасні технологічні рішення для оптимізації цього процесу.

Сучасні програмні засоби стають ключовим інструментом для розв'язання цих проблем. Інтерактивні інструменти, такі як онлайн-чати та системи онлайн-спілкування, відіграють роль посередника між пацієнтами та медичним персоналом, забезпечуючи швидку та зручну комунікацію. Це стає особливо важливим у сфері охорони здоров'я, де кожна мить може мати вирішальне значення.

Однією з переваг використання сучасних технологій є можливість впровадження системи запису, яка дозволяє пацієнтам ефективно керувати своїм часом та забезпечує медичному персоналу ефективний графік роботи. Використання чат-ботів та інших інтерактивних інструментів може полегшити процес заповнення необхідної інформації, роблячи його більш зручним та швидким для користувачів.

У контексті розробки мобільного застосунку для оптимізації процесу запису до медичних фахівців, сучасні технології грають вирішальну роль. Dart, Flutter, Firebase, Firestore, Dio та BLoC є інструментами, які надають унікальні можливості для створення вискоєфективних і зручних застосунків. Їхнє

використання в процесі розробки дозволяє не лише полегшити процес запису до медичних фахівців, але і підняти стандарт обслуговування в цій галузі.

Метою даної дипломної кваліфікаційної роботи магістра є глибоке дослідження сучасних можливостей програмного забезпечення для вдосконалення процесу запису до медичних фахівців. В рамках цього дослідження планується аналіз предметної області, огляд існуючих програмних рішень, розробка імплементації високоефективної моделі взаємодії користувача, проєктування та реалізація програмного забезпечення, а також тестування його функціональності.

Очікується, що результатом цієї роботи буде створення програмного забезпечення, яке не лише відповідає потребам користувача в ефективному записі до медичних фахівців, але також піднімає планку обслуговування в галузі медицини, роблячи її більш доступною та зручною для всіх учасників.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про онлайн-запис до медичних фахівців

У сучасному світі, де інноваційні технології диктують та трансформують різні аспекти нашого повсякденного життя, системи здоров'я та медичні послуги не залишаються осторонь цього неупинного розвитку. Швидкість, ефективність та доступність медичної допомоги стають критичними аспектами для забезпечення якісного обслуговування пацієнтів.

Одним із ключових елементів оптимізації медичних послуг та підвищення комфорту для пацієнтів є впровадження систем онлайн-запису до медичних фахівців. Це інноваційне рішення дозволяє пацієнтам здійснювати запис на консультації та процедури через Інтернет, забезпечуючи швидкий та зручний доступ до медичної допомоги.

Зараз, коли зміни в суспільстві вимагають відгуку та адаптації з боку медичних установ, онлайн-запис стає необхідністю для вдосконалення процесу надання медичних послуг. В цьому контексті виникає потреба в глибокому дослідженні та програмній реалізації мобільного застосунку для ефективного та зручного запису до медичних фахівців.

У даному контексті, розглядати поняття онлайн-запису до медичних фахівців необхідно як ключовий елемент, який не лише полегшує взаємодію між пацієнтом та медичним фахівцем, а й сприяє впровадженню прогресивних технологій у сферу охорони здоров'я. Детальний аналіз поняття онлайн-запису до медичних фахівців відкриває широкий обсяг можливостей для оптимізації та покращення систем медичної допомоги в цілому.

Онлайн-запис до медичних фахівців представляє собою інноваційний підхід до взаємодії пацієнтів із системою охорони здоров'я. Цей механізм базується на використанні сучасних інформаційних технологій для полегшення процесів запису на прийом до лікаря. Замість традиційного методу телефонних дзвінків чи особистого візиту до медичного закладу

пацієнт може використовувати онлайн-платформи чи мобільні додатки для ефективного управління своїм здоров'ям.

Системи онлайн-запису надають пацієнтам можливість переглядати профілі лікарів, ознайомлюватися з їхніми спеціалізаціями та графіками прийому, вони можуть обирати зручний для них час та дату прийому. Онлайн-системи забезпечують можливість вибору конкретної послуги або консультації, що сприяє оптимізації роботи лікарів та ефективному використанню їхнього часу.

Онлайн-запис до медичних фахівців є не тільки технологічною інновацією, але й зміною у підходах до організації медичного обслуговування, роблячи його більш доступним та зручним для всіх учасників системи охорони здоров'я.

Однією з ключових переваг є зручність та доступність для пацієнтів. Здійснення запису у будь-який час доби, незалежно від робочого графіку медичного закладу, робить цей підхід особливо привабливим для зайнятих людей. Можливість легко скасувати або перенести прийом також додає гнучкості та уникнення зайвого очікування.

Ефективне управління часом – ще одна вагома перевага. Автоматизована система онлайн-запису дозволяє лікарям оптимізувати свій робочий графік, уникати переповнення та забезпечувати раціональне розподілення часу для консультацій. Пацієнти отримують автоматичні нагадування про свої майбутні прийоми, що допомагає уникнути пропущених консультацій та сприяє більшій пунктуальності.

Спрощення адміністративних процесів – ще один важливий аспект. Зменшення обсягу телефонних дзвінків для запису та скасування прийому допомагає звільнити адміністративний персонал для виконання інших важливих завдань. Пацієнти отримують можливість самостійно керувати своїми записами, що робить взаємодію більш ефективною та зручною.

Адміністрація має можливість аналізувати ці дані для виявлення ефективності роботи лікарів та визначення популярних областей медичних

послуг. Наприклад, якщо певний лікар має велику кількість записів та позитивних відгуків, це може свідчити про його високий рівень професійної компетентності та задоволеності пацієнтів.

Загальна зручність і доступність для пацієнтів розширюється завдяки їх можливості переглядати профілі лікарів, вибирати фахівців з врахуванням їхнього досвіду, опису. Це дозволяє пацієнтам здійснювати освідчені вибори та забезпечує високий ступінь інформованості перед записом на консультацію.

Збереження медичної інформації в електронному форматі також є важливою перевагою онлайн-запису до медичних фахівців. Це підвищує безпеку та конфіденційність даних, дозволяє уникнути ризиків втрати чи несанкціонованого доступу до конфіденційної інформації. Захищена електронна система гарантує, що дані залишаються конфіденційними та безпечними.

Одні з головних функцій для запису включають:

- вибір лікаря. Пацієнти можуть переглядати профілі лікарів, включаючи їхній медичний досвід, освіту, спеціалізацію;
- графік роботи. Лікарі можуть публікувати свої розклади роботи, а пацієнти можуть обрати час прийому, який найбільше їм підходить;
- повідомлення та нагадування. Система може надсилати автоматичні повідомлення, включаючи нагадування про прийоми, інструкції перед візитом або можливі зміни у розкладі лікаря;
- ці платформи створюють не лише можливість для ефективного запису до лікаря, але й сприяють покращенню комунікації між пацієнтами та медичними фахівцями, спрощуючи адміністративні процеси та роблячи систему охорони здоров'я більш доступною та прозорою.

Видами онлайн-запису є:

- вебплатформи - представляють собою інтерактивні вебсайти, створені для забезпечення пацієнтам зручності та ефективності при записі на прийом до лікаря. Ці платформи визначаються своєрідністю інтерфейсу, який надає можливість вибрати медичного фахівця за його спеціальністю, локалізацією та іншими критеріями;

— мобільні додатки - представляють собою інноваційний підхід до організації медичних прийомів та спілкування пацієнтів із медичними фахівцями. Однією з ключових переваг таких додатків є їхня мобільність, яка дозволяє пацієнтам здійснювати запис та взаємодіяти із системою охорони здоров'я прямо зі своїх смартфонів. Ці додатки зазвичай володіють інтуїтивним інтерфейсом, який робить процес запису легким та зручним для різних користувачів. Пацієнти можуть швидко знайти лікарів за спеціальністю, переглядати їхні графіки та вибирати зручний час для прийому. Однією з ключових функцій мобільних додатків є можливість отримання сповіщень про майбутні прийоми та нагадувань. Це сприяє пунктуальності та дисциплінованості пацієнтів, а також допомагає уникнути забути прийоми. Важливою складовою є також можливість скасування або перенесення прийому через мобільний додаток, що дозволяє ефективно управляти своїм графіком та уникнути зайвих непорозумінь. Такі мобільні додатки стають важливим інструментом в сучасній медичній практиці, підвищуючи зручність та доступність медичних послуг для пацієнтів та сприяючи оптимізації роботи медичних установ.

Отже, онлайн-запис до медичних фахівців відіграє важливу роль у поліпшенні доступності та якості медичних послуг. Зручність для пацієнтів, яка випливає з можливості записатися в будь-який момент і з будь-якого місця, стає ключовим фактором у підвищенні задоволення пацієнтів. Можливість переглядати графіки лікарів і вибирати зручний час прийому додає гнучкості та сприяє оптимальному використанню робочого часу медичних працівників.

Спрощення адміністративних процесів, таких як ефективний розподіл часу для уникнення переповнення лікарських кабінетів, підвищує ефективність медичних установ. Зменшення потоку телефонних дзвінків, пов'язаних із записом та скасуванням прийому, дозволяє персоналу приділяти більше уваги медичному обслуговуванню.

Можливість відстеження записів та отримання нагадувань про майбутні прийоми через онлайн-системи дозволяє пацієнтам більш активно брати участь у керуванні своїм здоров'ям. Використання мобільних додатків

для запису сприяє технологічній модернізації медичної сфери, що відображає важливий крок у напрямку покращення обслуговування пацієнтів та створення платформи для майбутніх інновацій у галузі медичної допомоги.

У контексті глобальних тенденцій цифрової трансформації, онлайн-запис до медичних фахівців відображає загальний рух до використання технологій для поліпшення життя та забезпечення ефективної медичної допомоги. З урахуванням динаміки розвитку технологій, цей підхід стає основою для майбутнього розвитку медичної сфери, сприяючи створенню більш доступної, гнучкої та ефективної системи надання медичних послуг.

1.2 Дослідження мобільних застосунків для запису до медичних фахівців

Останнім часом розроблено багато різних мобільних застосунків для запису до медичних фахівців. Проаналізуємо застосунки для запису до медичних фахівців.

1.2.1 Мобільний застосунок Helsi

Helsi (рис. 1.1) – представляє собою інноваційну медичну інформаційну систему, яка допомагає автоматизувати роботу установ охорони здоров'я в Україні. З початку серпня 2022 року власником системи є компанія Київстар. Ця система розроблена для оптимізації роботи медичних закладів, лікарів, лабораторій та стаціонарів, а також для ефективного ведення електронних медичних карток. Зручний та інтуїтивно зрозумілий інтерфейс допомагає користувачам формувати звіти, використовувати вбудований конструктор бланків і форм для медичних документів, а також взаємодіяти з програмою медичних гарантій у співпраці з Національною службою здоров'я України (НСЗУ).

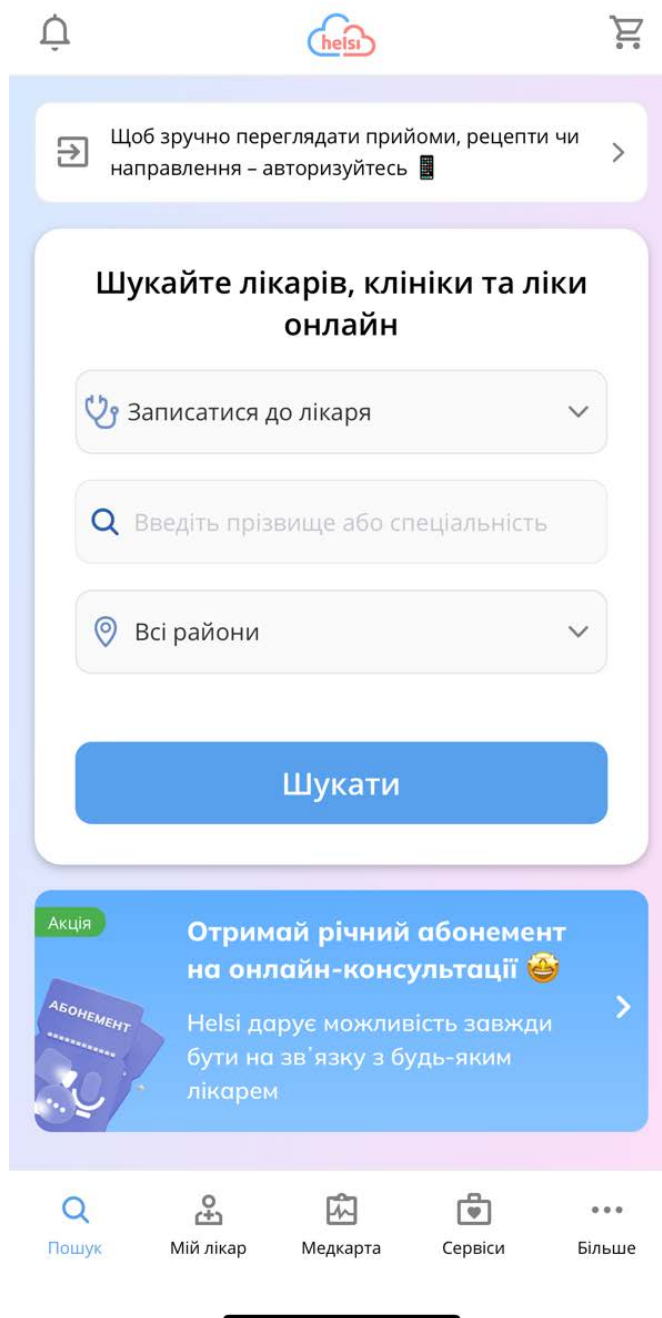


Рисунок 1.1 – Мобільний застосунок Helsi [1]

Пацієнтам надається можливість швидко та зручно знаходити лікаря, записуватися на прийом онлайн, отримувати консультації в клініці без черги та переглядати свої електронні медичні картки. Пацієнти можуть також вибрати свого сімейного лікаря для укладення декларації за медичною

реформою, просто зареєструвавшись на вебсайті Helsi чи у мобільному додатку та обираючи лікаря у своєму районі.

Реєстрація пацієнта чи лікаря в Helsi здійснюється через веб-портал або мобільний додаток. Для реєстрації пацієнта необхідно вказати такі дані:

- а) ім'я та прізвище;
- б) дату народження;
- в) номер мобільного телефону;
- г) електронну адресу;
- д) ідентифікаційний код;
- е) адресу проживання.

Для реєстрації лікаря необхідно вказати такі дані:

- а) ім'я та прізвище;
- б) дату народження;
- в) номер мобільного телефону;
- г) електронну адресу;
- д) ліцензію на медичну практику;
- е) спеціалізацію;
- ж) адресу медичного закладу, в якому працює лікар.

Для запису необхідно вказати такі дані:

- а) дату і час прийому;
- б) лікаря, до якого записується;
- в) спеціалізацію лікаря;
- г) призначення прийому.

Пацієнт може відслідковувати свій запис на прийом до лікаря. Для цього необхідно перейти в розділ «Записи» і вибрати запис, який потрібно відслідкувати. У цьому розділі відображається інформація про дату і час прийому, лікаря, до якого записується, спеціалізацію лікаря, приначення прийому.

Пацієнт може скасувати або змінити свій запис на прийом до лікаря. Для цього необхідно перейти в розділ «Записи» і вибрати запис, який потрібно скасувати або змінити. У цьому розділі є кнопка «Скасувати» або «Змінити».

Лікар може заповнити свою анкету вказавши такі дані:

- а) ім'я та прізвище;
- б) дату народження;
- в) номер мобільного телефону;
- г) електронну адресу;
- д) ліцензію на медичну практику;
- е) спеціалізацію;
- ж) звання;
- з) адресу медичного закладу, в якому працює лікар;
- и) графік роботи.

Helsi є потужним інструментом, який може допомогти зробити медичну систему в Україні більш доступною, ефективною та прозорою.

Ось деякі з основних недоліків Helsi:

- недостатня доступність. Наразі Helsi доступний лише в Україні. Це означає, що пацієнти з інших країн не можуть користуватися цим сервісом;
- вартість послуг. Деякі послуги Helsi, наприклад, онлайн-консультації з лікарями, є платними. Це може бути проблемою для деяких пацієнтів;
- недостатня безпека. Хоча Helsi використовує сучасні технології шифрування для захисту даних пацієнтів, завжди є ризик їхнього витоку.

1.2.2. Мобільний застосунок MeetDoc

MeetDoc (рис. 1.2) – це багатофункціональний застосунок для запису на прийом до лікаря, який об'єднує лікарів та пацієнтів по всьому світу. Він служить зручною платформою для пацієнтів, щоб швидко записуватися на прийом до улюблених лікарів онлайн і отримувати найкращі медичні онлайн-

послуги. Цей застосунок також служить ідеальною платформою для лікарів, дозволяючи їм спростити процес запису на прийом. MeetDoc пропонує широкий спектр функцій, які роблять його цінним ресурсом для пацієнтів та лікарів.

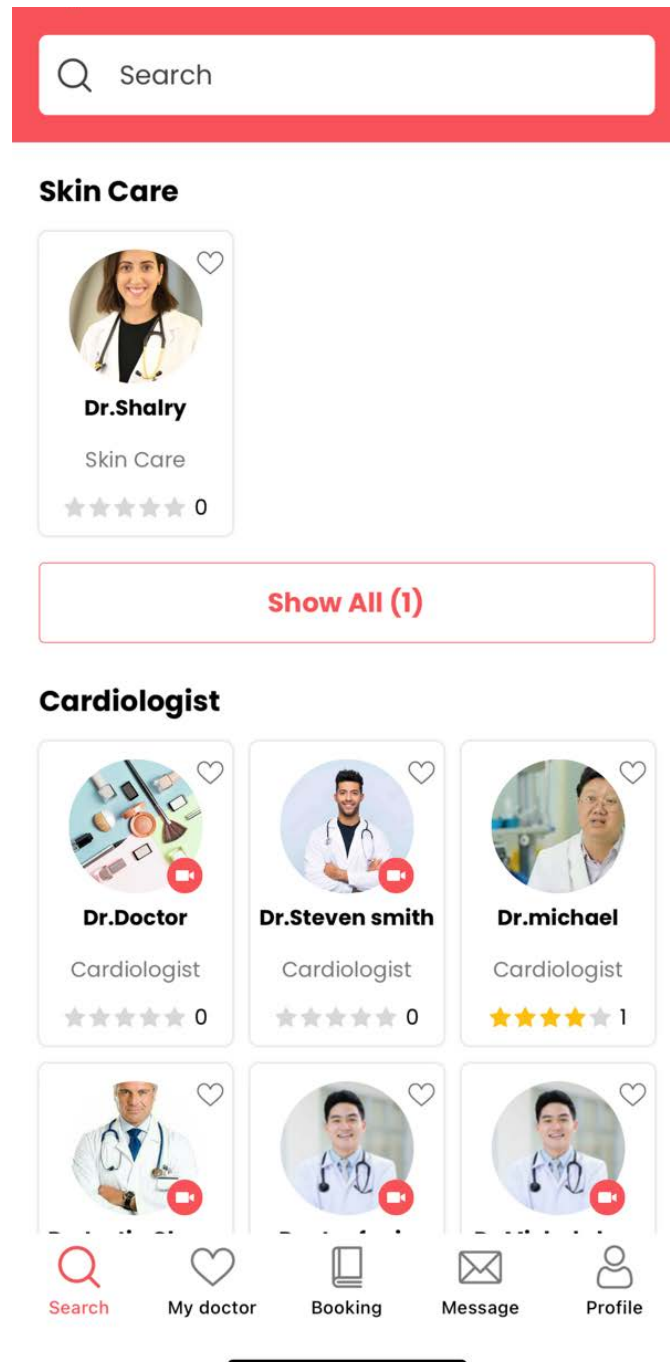


Рисунок 1.2 – Мобільний застосунок MeetDoc [2]

Пацієнти можуть шукати лікарів за різними параметрами, такими як місцезнаходження, тип спеціалізації та вибрати зручний час для візиту.

Після того, як пацієнт знайде лікаря, якого він хоче відвідати, він може вибрати зручний час для візиту. MeetDoc пропонує широкий вибір часу для візитів, включаючи ранкові, денні, вечірні та вихідні.

Ось кілька прикладів того, як пацієнти можуть використовувати MeetDoc:

- пацієнт, який відчуває симптоми застуди, може знайти педіатра або терапевта, який приймає в потрібний йому час;
- пацієнт, який має хронічне захворювання, може знайти лікаря-спеціаліста, який спеціалізується на його захворюванні;
- пацієнт, який планує вагітність, може знайти акушера-гінеколога у своєму районі.

Лікарі, у свою чергу, можуть використовувати платформу для:

- приймати онлайн-записи: MeetDoc дозволяє лікарям створювати профілі, в яких вони можуть надавати інформацію про свою практику та послуги. Ця інформація може включати спеціальності лікаря, досвід, освіту, а також страховки, які приймаються;
- управління своїм розкладом прийому: MeetDoc дозволяє лікарям легко керувати своїм розкладом прийому. Лікарі можуть створювати і редагувати графіки прийому, отримувати сповіщення про скасування або затримки.

Ці можливості можуть допомогти лікарям:

- поліпшити ефективність свого бізнесу: MeetDoc може допомогти лікарям оптимізувати свій графік прийому, зменшити кількість дзвінків від пацієнтів і підвищити рівень задоволеності пацієнтів;
- краще обслуговувати своїх пацієнтів: MeetDoc може допомогти лікарям надавати пацієнтам більш зручний і доступний спосіб отримання медичної допомоги.

Ось кілька прикладів того, як лікарі можуть використовувати MeetDoc:

- лікар, який працює в приватній практиці, може використовувати його для управління своїм розкладом прийому і приймати онлайн-записи. Це допоможе йому заощадити час і полегшить роботу його адміністратора;

– лікар, який працює в клініці, може використовувати його для спілкування з пацієнтами за допомогою електронної пошти. Це може допомогти йому відповісти на запитання пацієнтів і надати їм додаткову інформацію про лікування.

Однак у MeetDoc також є деякі недоліки, які варто враховувати:

– не завжди відображає точну інформацію про лікарів та їхні послуги. Наприклад, у деяких випадках у профілях лікарів може бути відсутня інформація про їхню спеціальність, досвід або вартість послуг;

– MeetDoc сприяє «медичному споживацтву», тобто надмірному використанню медичних послуг. Це пов'язано з тим, що платформа робить процес запису на прийом до лікаря дуже простим, що може призвести до того, що люди записуються на прийоми, які їм не потрібні.

Загалом, MeetDoc є корисним інструментом для пошуку та запису на прийом до лікаря. Однак важливо бути обізнаним про її недоліки, щоб зробити обґрунтований вибір щодо того, чи використовувати цю платформу.

1.2.3 Мобільний застосунок Doc.ua

Doc.ua (рис. 1.3) – це медична платформа, яка дозволяє записатися до лікаря в будь-якому місті України. Була заснована в 2012 році і з того часу стала однією з найстаріших та найпопулярніших платформ у країні. На платформі зареєстровано понад 100 тисяч лікарів і понад 1 мільйон пацієнтів. Doc.ua відрізняється величезним каталогом діагностичних центрів в будь-якому місті України - Київ, Одеса, Харків, Львів, Дніпро та ін. Діагностичні центри пропонують широкий спектр досліджень: МРТ, КТ, УЗІ, рентген, ендоскопічні методи дослідження та інші.

Довідник лікарів на Doc.ua дозволяє пацієнтам знайти лікаря за спеціальністю, містом та іншими параметрами. У довіднику представлені лікарі з усієї України, включаючи їхні контактні дані та інформацію про досвід роботи.

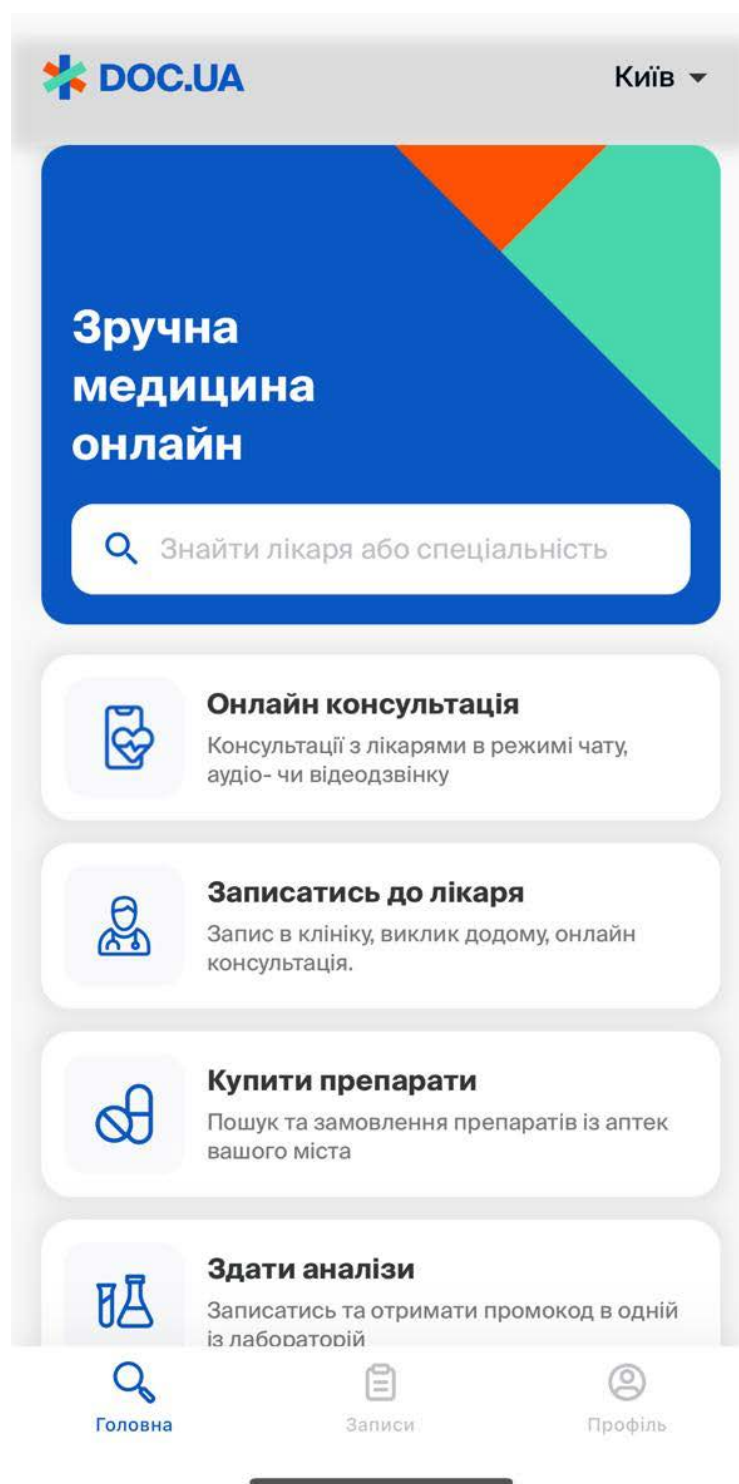


Рисунок 1.3 - Медична платформа Doc.ua [3]

Реєстрація пацієнта на Doc.ua здійснюється за допомогою електронної пошти або номера телефону. Для реєстрації лікаря необхідно надати інформацію про себе, свою освіту, досвід роботи та спеціалізацію.

Запис на прийом можна здійснити за допомогою фільтрів пошуку, які дозволяють вибрати лікаря за спеціальністю, містом, районом, ціною

консультації та іншими параметрами. Після вибору лікаря можна ознайомитися з його анкетой та записатися на прийом.

Пацієнт може відслідковувати свій запис на прийом на Doc.ua в особистому кабінеті. У особистому кабінеті також можна знайти інформацію про лікаря, до якого призначений прийом, дату, час і місце прийому.

Пацієнт може скасувати або змінити свій запис на прийом на Doc.ua в особистому кабінеті. Для скасування або зміни запису необхідно натиснути відповідну кнопку в особистому кабінеті.

Лікар може заповнити свою анкету на Doc.ua, додавши інформацію про себе, свою освіту, досвід роботи, спеціалізацію та графік його роботи.

У особистому кабінеті пацієнта на Doc.ua міститься інформація про всі його актуальні записи, а також можливість скасування або зміни запису.

Новини та статті про здоров'я на Doc.ua охоплюють широкий спектр тем, пов'язаних із медициною, здоров'ям та здоровим способом життя. Серед них:

- нові дослідження та відкриття у сфері медицини;
- актуальні медичні проблеми та тенденції;
- поради щодо профілактики захворювань та підтримки здоров'я;
- рекомендації щодо здорового харчування, фізичної активності та інших аспектів здорового способу життя.

Новини та статті на Doc.ua написані фахівцями у галузі медицини та здоров'я, тому вони є достовірними та актуальними, а також написані доступною мовою, що дозволяє будь-кому зрозуміти їх. Вони є цінним ресурсом для пацієнтів, які хочуть бути в курсі останніх медичних новин, отримати корисну інформацію про здоров'я та здоровий спосіб життя, а також краще зрозуміти своє захворювання.

Загальні недоліки сервісу медицини онлайн Doc.ua:

- недостатня кількість лікарів у деяких регіонах. У деяких містах України, особливо в маленьких, може бути обмежений вибір лікарів у Doc.ua;
- висока вартість послуг. Ціни на послуги лікарів у Doc.ua можуть бути вищими, ніж у традиційних медичних закладах.

Порівняльний аналіз мобільних застосунків для запису до медичних фахівців наведено у табл. 1.1.

Таблиця 1.1 - Порівняльний аналіз мобільних застосунків для запису до медичних фахівців

Критерій, за яким порівнюють	Helsi	MeetDoc	DOC.ua
Наявність в Україні	+	-	+
Мова інтерфейсу	українська, російська, англійська	англійська, французька	українська, російська, англійська
Наявність медичних фахівців	+	+	+
Процес запису	+	+	+
Ціна	базовий тариф безкоштовний, платні тарифи від 129 грн/місяць	базовий тариф безкоштовний, платні тарифи від 15 USD/місяць	базовий тариф безкоштовний, платні тарифи від 99 грн/місяць
Критика	+-	+-	+-

Розшифровка символів:

а) + - перевага;

б) - - недолік;

в) +- - перевага або недолік, залежно від потреб користувача.

Як видно з таблиці, всі три платформи пропонують схожі функції та можливості, але мають свої особливості. Helsi та MeetDoc мають більш широкий вибір медичних фахівців, тоді як DOC.ua має додаткові функції.

При розробці мобільного застосунку для запису до медичних фахівців необхідно забезпечити такі функціональні вимоги:

- пошук лікаря. Користувач повинен мати можливість швидко і легко знайти лікаря, відповідного його потребам. Пошук повинен проводитися за такими параметрами, як спеціальність, місце розташування, рейтинг, наявність страховки тощо; бронювання прийому. Користувач повинен мати можливість швидко і легко забронювати прийом до лікаря. Процес бронювання повинен бути простим і зрозумілим;
- відслідковування запису. Після запису на прийом пацієнти можуть відстежувати свій запис у своєму обліковому записі. Вони можуть переглянути час та дату прийому;
- скасування або зміна запису. Пацієнти можуть скасувати або змінити свій запис у будь-який час до початку прийому;
- підтвердження або скасування заявки на запис. Лікар може підтвердити або скасувати заявку на запис від пацієнта;
- переглядання актуальних статей. Статті є цінним ресурсом для пацієнтів, які бажають бути інформованими щодо останніх медичних новин, отримати корисні відомості про здоров'я та здоровий спосіб життя, а також отримати більше знань про своє захворювання. Вони будуть відображатись по запиту до API.

Крім цих функціональних вимог, при розробці мобільного застосунку для запису до медичних фахівців необхідно також врахувати такі фактори:

- безпека. Застосунок повинен забезпечувати захист даних користувачів, у тому числі персональних даних та медичної інформації;
- доступність. Застосунок повинен бути доступним для користувачів з різними типами пристроїв, у тому числі з мобільних телефонів та планшетів.

1.3 Висновки за розділом 1

Досліджено поняття та потреби в мобільному застосунку для запису до медичних фахівців. Визначено, що існує значна необхідність в зручному та ефективному інструменті для пацієнтів, який дозволить їм записатися на консультацію або прийом до лікаря шляхом використання спеціалізованого програмного забезпечення.

Проаналізовано переваги використання такого застосунку для запису до медичних фахівців. Визначено, що це може спростити процес отримання медичної допомоги для пацієнтів, зменшити час очікування та полегшити роботу медичного персоналу. Описано основні функції, які повинний включати мобільний застосунок, такі як запис до лікаря, відслідковування запису, скасування його.

Досліджено існуючі програмні рішення для запису до медичних фахівців. Визначено, що деякі існуючі рішення можуть бути вартісними, але часто вони вимагають плати за їх використання. Тому обґрунтовано потребу у розробці безкоштовного мобільного застосунку для запису до медичних фахівців з метою забезпечення доступу до цих сервісів для широкого кола користувачів.

Визначено основні функціональні вимоги до розробки мобільного застосунку для запису до медичних фахівців, такі як інтуїтивний інтерфейс, можливість персоналізації акаунтів пацієнтів та лікарів, а також ефективна система підтримки та нагляду за записами. Отже, розробка такого застосунку може значно поліпшити доступність та якість медичної допомоги для пацієнтів та сприяти оптимізації роботи медичних установ.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування та інших технологій розробки

Dart - це клієнтська мова програмування для розробки швидких додатків на будь-якій платформі. Її мета полягає в тому, щоб запропонувати найпродуктивнішу мову програмування для мультиплатформної розробки, поєднану з гнучкою платформою виконання для фреймворків додатків.

Dart призначений для технічного середовища, яке особливо підходить для розробки клієнтських застосунків, надаючи пріоритет як розробці (підзавантаження збереженого стану за менше одної секунди), так і високоякісному виробництву для широкого спектру цілей компіляції (веб, мобільні та настільні комп'ютери). Dart також є основою Flutter. Dart надає мову і середовище виконання, які забезпечують роботу додатків Flutter, але Dart також підтримує багато основних завдань розробника, таких як форматування, аналіз і тестування коду.

Мова Dart є типобезпечною; вона використовує статичну перевірку типів, щоб гарантувати, що значення змінної завжди відповідає її статичному типу. Іноді це називають звуковою типізацією. Хоча типи є обов'язковими, анотації типів є необов'язковими через виведення типів. Система типізації Dart також є гнучкою, дозволяючи використовувати динамічний тип у поєднанні з перевіркою під час виконання, що може бути корисним під час експериментів або для коду, який має бути особливо динамічним.

Це високорівнева мова: вона оперує абстрактними поняттями, віддаленими від «заліза», але близькими до людини. Тому код на Dart лаконічний і зрозумілий, а синтаксис легко освоюється. Особливо зручно вважається використання Dart для людей, які вже вивчали інші мови програмування.

Творці Dart - компанія Google, яка також вигнала мову Go. Dart має відкритий вихідний код, мова є повністю безкоштовною та вільно поширюється.

Dart має багатий набір основних бібліотек, які забезпечують необхідність для багатьох повсякденних програмних завдань:

- вбудовані типи, колекції та інші основні функції для кожної програми Dart (`dart:core`);
- розширені типи колекцій, такі як черги, зв'язні списки, хеш-карти та бінарні дерева (`dart:collection`);
- кодери та декодери для конвертації між різними представленнями даних, включаючи JSON та UTF-8 (`dart:convert`);
- математичні константи та функції, а також генерація випадкових чисел (`dart:math`);
- підтримка введення/виведення файлів, сокетів, HTTP та інших для застосунків, що не є веб-додатками (`dart:io`);
- підтримка асинхронного програмування за допомогою класів, таких як `Future` та `Stream` (`dart:async`). Списки, які ефективно обробляють дані фіксованого розміру (наприклад, беззнакові 8-байтові цілі числа) та числові типи SIMD (`dart:typed_data`);
- інтерфейси зовнішніх функцій для взаємодії з іншим кодом, який має інтерфейс у стилі C (`dart:ffi`);
- конкурентне програмування за допомогою ізолятів – незалежних робітників, схожих на потоки, але не діляться пам'яттю, взаємодіючи лише через повідомлення (`dart:isolate`);
- HTML-елементи та інші ресурси для веб-додатків, які повинні взаємодіяти з браузером та об'єктно-орієнтованим об'єктним моделлю (DOM) (`dart:html`).

Поза основними бібліотеками, багато API надаються через широкий набір пакетів. Команда Dart публікує багато корисних допоміжних пакетів, таких як:

- `dio`;
- `intl`;
- `http`;
- `crypto`;
- `markdown`.

Основними особливостями Dart є:

- орієнтований на створення інтерфейсів. Dart був створений як мова для розробки інтерфейсів, тому в ньому є багато функцій для їх створення. Зараз ця особливість знайшла застосування в програмуванні мобільних додатків: створювати візуальний інтерфейс за допомогою Dart досить зручно. Ідея розробників полягала в тому, щоб зменшити кількість перемикачів між макетом і кодом, щоб розробник не втрачав час і не відволікався. Деякі функції мови спрямовані саме на це. Наприклад, при компіляції в JavaScript автоматично створюється шаблонна HTML-сторінка з макетом. Її можна відразу відкрити у браузері і подивитися, що вийшло;
- наслідок від C і C++. Dart сильно відчув вплив системних мов C і C++, і це знайшло відображення у його синтаксисі. Він схожий на C: це означає, що правила написання коду та його структура подібні до C. Хоча Dart простіший у вивченні та орієнтований на інші цілі, він успадкував деякі важливі елементи, наприклад, основний блок програми розміщується у функції `main()`;
- факультативна типізація. Типізація в мовах програмування буває статичною та динамічною. У першому випадку розробник повинен самостійно вказувати, до якого типу буде відноситися змінна: число, рядок, масив чи щось інше. У другому випадку мова вирішує це сама на основі даних, які розробник записав у змінну. У Dart об'єднали ці підходи — типізація там факультативна. Це означає, що розробник може вказати тип при створенні змінної, або не вказувати — обидва підходи припустимі. Який з них краще використовувати, залежить від задачі;
- класичний ООП. В Dart реалізований класичний підхід до об'єктно-орієнтованого програмування, який наразі вважається однією з найпопулярніших парадигм. Тут є класи, включаючи абстрактні, реалізовано наслідування — все це без нововведень і експериментів, так, як це передбачає парадигма. Тому писати ООП-код в Dart досить зручно: не потрібно вигадувати «костилі» і боротися з особливостями мови;
- ізолятори. Dart — одно потокова мова. Це означає, що всі програми на ній виконуються в єдиному потоці подій, а не в кількох. Однопоточність

також робить його схожим з JavaScript. Але іноді процеси потрібно паралелізувати, щоб кілька дій могли виконуватися одночасно. Для цього в Dart реалізований механізм ізоляторів (isolate). Іноді їх ще перекладають як ізоляти. Ізолятор — це сутність, яка виконує які-то дії. У кожного ізолятора своя область пам'яті, він може виконуватися одночасно з іншими і обмінюватися з ними даними. Це своєрідні віртуальні потоки, створені на більш високому рівні абстракції, ніж реальні. Ними управляє Event Loop — процес, який працює, поки відкрита програма, і послідовно виконує код всередині різних ізоляторів. Він працює з чергою, яка діє за принципом FIFO: «перший увійшов, перший вийшов». Тобто той процес, який «прийшов» в чергу раніше, і виконається раніше інших. Є дві черги: для простих коротких дій і для блоків коду, які повинні виконатися відповідно до якої-небудь зовнішньої події;

- гаряче перезавантаження. Зручна особливість Dart — гаряча перезавантаження, або hot reload. Термін означає, що код програми можна виправити, поки вона запущена, і зміни внесуться «на льоту». Для цього не потрібно відкривати і знову закривати програму. В JavaScript і подібних мовах теж є інструменти для hot reload, але там доводиться використовувати зовнішні рішення, а в Dart ця можливість вбудована з самого початку. Тому гаряча перезавантаження в цій мові працює швидше і ефективніше;

- компільованість. Технологія компіляції в Dart надає гнучкість в виконанні коду залежно від цільової платформи. Для мобільних та настільних пристроїв Dart може використовувати віртуальну машину Dart з JIT-компіляцією або AOT-компілятор для генерації машинного коду. Це забезпечує оптимальну продуктивність та ефективність виконання для цих пристроїв. Для веб-платформи Dart може компілювати код як для розробки, так і для виробництва. У випадку розробки веб-додатків, його веб-компілятор перетворює Dart-код у JavaScript, що полегшує інтеграцію з екосистемою веб-технологій.

Ілюстрація підтримуваних цілей Dart представлена на рис. 1.4.

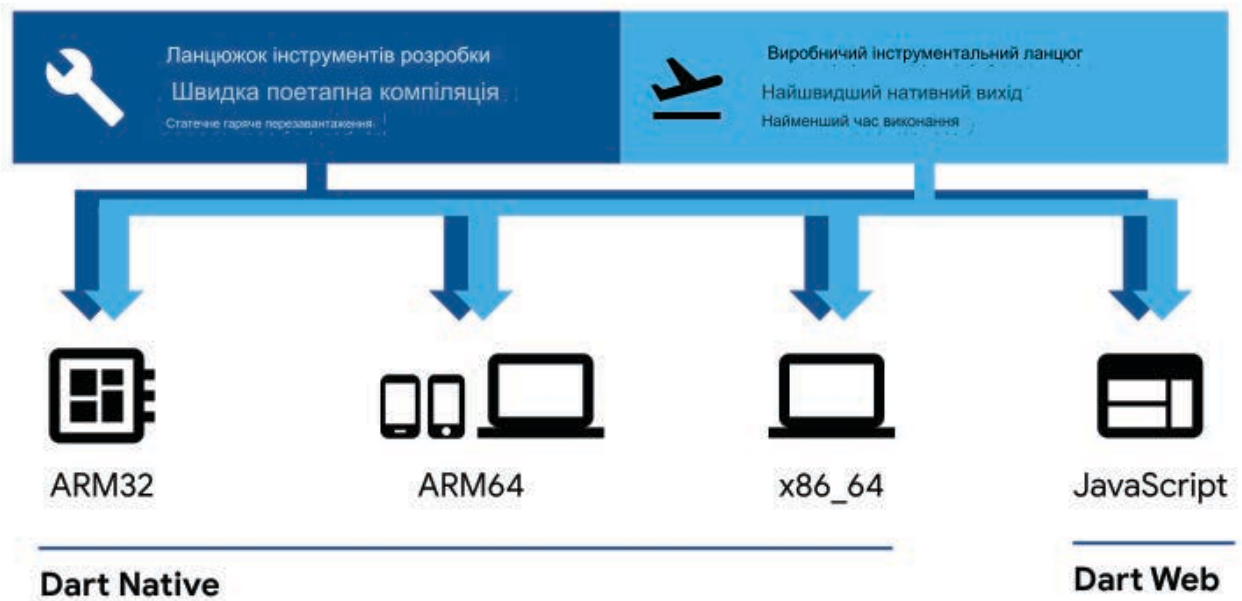


Рисунок 1.4 - Ілюстрація підтримуваних цілей Dart

Фреймворк Flutter є популярним багатоплатформенним набором інструментів для створення користувацького інтерфейсу, який працює на платформах iOS, Android, macOS, Windows, Linux і в Інтернеті, і працює на платформі Dart, надаючи інструменти та бібліотеки інтерфейсу користувача.

Обґрунтування вибору мови програмування для розробки мобільного застосунку для запису до медичних фахівців наведено у табл. 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки мобільного застосунку для запису до медичних фахівців

Критерій порівняння мов програмування	Мова програмування		
	Dart	Java	Rust
Зручність використання для створення мобільного застосунку для запису до медичних фахівців	+	+ -	+ -

Продовження таблиці 2.1

Критерій порівняння мов програмування	Мова програмування		
	Dart	Java	Rust
Продуктивність	+	-	-
Легкість вивчення	+	+	+
Розробка мобільних додатків	+	+	-
Екосистема та фреймворки	+	+	+
Спільнота та підтримка	+	+	-+
Переносимість коду	+	-	+

Розшифровка символів:

- а) + - перевага;
- б) - - недолік;
- в) +- - перевага або недолік, залежно від потреб користувача.

Таким чином, для розробки застосунку для запису до медичних фахівців обрано мову програмування Dart.

2.2 Вибір фреймворку для створення мобільного застосунку

В якості фреймворку для створення мобільного застосунку для запису до медичних фахівців було використано Flutter.

Flutter – це кроссплатформова платформа для розробки програмного забезпечення з відкритим кодом, розроблена Google. Він використовується для створення програм для Android, iOS, веб-браузерів, Windows, macOS і Linux.

Flutter використовує мову програмування Dart і компілюється в машинний код, що забезпечує високу продуктивність і швидкість.

Архітектура Flutter відрізняється від інших програмних каркасів (React, Apache Cordova) тим, що він не використовує для побудови інтерфейсу мови HTML, CSS та Javascript, відповідно і вбудований рушій WebView. [4].

Flutter використовує тільки одну мову програмування Dart.

Базовим компонентом Flutter-програми є «віджет», який, у свою чергу, може складатися з інших віджетів. Віджет описує логіку, взаємодію та дизайн елемента інтерфейсу користувача з реалізацією, подібною до React. На відміну від інших крос-платформних інструментів, таких як React Native та Xamarin, які малюють віджети за допомогою нативних компонентів платформи, Flutter рендерить віджети самостійно на попиксельній основі [5].

Існує два основних типи віджетів: віджети без стану та віджети зі станом. Віджети без стану оновлюються лише при зміні вхідних даних, що робить їх дуже ефективними, в той час як віджети зі станом можуть викликати метод `setState()` для оновлення внутрішнього стану та перемальовування [5].

Архітектура фреймворку Flutter представлена на рис. 1.4.

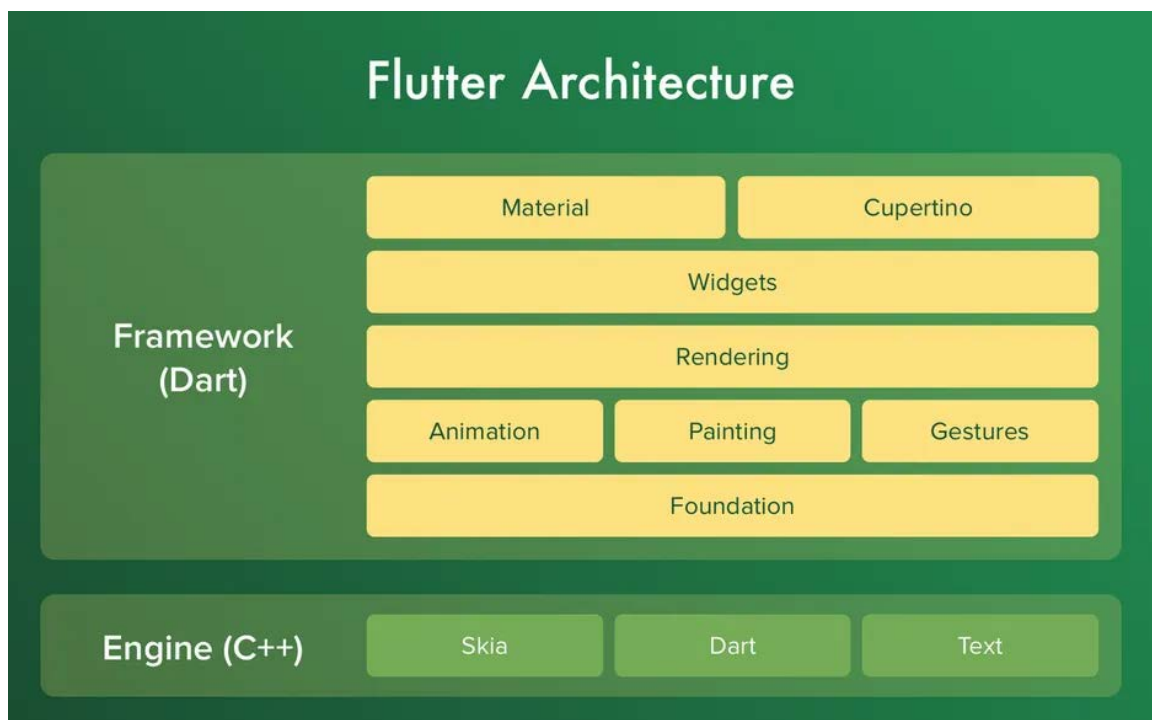


Рисунок 2.1 – Архітектура фреймворку Flutter

Додатки, розроблені за допомогою Flutter, відповідають потребам як власників бізнесу, пропонуючи хорошу якість за прийнятну ціну, так і спеціалістам з ІТ, забезпечуючи високу зручність використання та швидкість.

Переваги розробки додатків на Flutter:

- додатки Flutter є бюджетними. Додатки, розроблені в Flutter, не займають багато часу на розробку, і тому вони не такі дорогі, як нативні додатки. Це ідеально підходить для стартапів MVP, коли вам потрібно швидко протестувати свою бізнес-модель: вартість розробки додатків Flutter буде нижчою, ніж розробка для двох платформ. Крім того, вам не потрібно утримувати великий штат розробників Android та iOS: вам знадобиться не більше кількох інженерів у команді;

- продуктивність, як у нативних додатків. З точки зору продуктивності Flutter - нативний додаток реального часу. Фреймворк прагне забезпечити продуктивність 60 кадрів в секунду або 120 кадрів в секунду на пристроях, що підтримують оновлення 120 Гц. Додаткам, створеним у Flutter, не потрібен міст для взаємодії з нативними компонентами (що впливає на продуктивність), як це роблять інші фреймворки;

- розробка відбувається швидше. Щоб отримати вигляд і продуктивність нативних додатків, мобільні розробники раніше створювали два окремі додатки, використовуючи різні мови – Swift/Objective-C для iOS та Java/Kotlin для Android. Flutter дозволяє програмістам використовувати єдину базу коду, об'єднувати команди, зменшувати ризики та скорочувати час виходу на ринок. І все це при отриманні переваг нативного вигляду та продуктивності;

- функція Hot Reload. Серед переваг Flutter є функція під назвою Hot Reload. Коли розробник натискає кнопку перезавантаження, всі зміни коду миттєво відображаються на гаджетах, емуляторах і симуляторах. Якщо програмісти змінюють деякий код у додатках, створених за допомогою Flutter, після перекомпіляції, їм не потрібно повертатися назад або вручну відновлювати стан, щоб побачити, що змінилося. Hot Reload дає розробникам Flutter можливість надавати більше функцій за менший час. Плюс робить

легшим додавання нових функцій частіше, щоб підтримувати активність користувачів.

Обґрунтування вибору фреймворку Flutter для створення мобільного застосунку для запису до медичних фахівців наведено у табл. 2.2.

Таблиця 2.2 – Обґрунтування вибору фреймворку Flutter для створення мобільного застосунку для запису до медичних фахівців

Критерій	Перевага
Нативність	Flutter забезпечує нативну продуктивність на всіх платформах, що означає, що застосунки, створені за допомогою Flutter, будуть працювати так само швидко і плавно, як і застосунки, створені за допомогою native-фреймворків. Це важливо для застосунків, які вимагають швидкої і плавної роботи, таких як застосунки для запису до медичних фахівців.
Розробка для кількох платформ	Flutter дозволяє розробляти застосунки для кількох платформ за допомогою одного коду. Це може значно заощадити час і витрати на розробку. Для застосунків для запису до медичних фахівців, які повинні бути доступні на різних платформах, це може бути важливим фактором.
Компонентний дизайн	Flutter використовує компонентний дизайн, який робить застосунки простішими в розробці та підтримці. Компонентний дизайн також дозволяє створювати застосунки, які легко адаптуються до різних розмірів екранів і пристроїв.

Продовження таблиці 2.2

Критерій	Перевага
Швидка розробка	Flutter використовує байт-код, який робить його швидким і ефективним у розробці. Це може значно скоротити час, необхідний для розробки застосунку. Для застосунків для запису до медичних фахівців, які повинні бути запущені в найкоротші терміни, це може бути важливим фактором.
Розробка для майбутнього	Flutter є сучасним фреймворком, який підтримується компанією Google. Це означає, що Flutter буде продовжувати розвиватися і вдосконалюватися в майбутньому. Це важливо для застосунків для запису до медичних фахівців, які повинні бути здатними підтримувати найновіші функції та можливості.

Таким чином, для створення мобільного застосунку для запису до медичних фахівців обрано фреймворк Flutter.

2.3 Розробка моделі взаємодії користувача з мобільним застосунком для запису до медичних фахівців

З метою можливості розробки мобільного застосунку для запису до медичних фахівців створимо відповідну модель у нотації UML у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з мобільним застосунком для запису до медичних фахівців будемо враховувати те, що користувачі застосунку мають можливості реєстрації та авторизації, перегляду списку лікарів, деталі лікаря, запису на прийом, скасування або зміна запису, перегляд актуальних статей. При цьому зареєструвавшись як лікар, є можливість

заповнити анкету та бути в загальному списку лікарів, щоб надавати послуги пацієнтам.

Модель взаємодії користувача з мобільним застосунком для запису до медичних фахівців повинна бути простою та інтуїтивно зрозумілою, щоб пацієнти могли легко та швидко записатися на прийом до потрібного лікаря.

Основний успішний сценарій користувача при роботі з мобільним застосунком для запису до медичних фахівців можна описати таким чином:

- вхід до застосунку. Пацієнт повинен мати можливість увійти до застосунку, використовуючи свій номер телефону або електронну адресу та пароль. Якщо пацієнт не має облікового запису, він може зареєструватися в застосунку, ввівши свої особисті дані;
- вибір лікаря. Пацієнт повинен мати можливість вибрати лікаря, до якого він хоче записатися. Застосунок повинен надавати список лікарів, які працюють у даному медичному закладі, а також їхні спеціалізації;
- вибір часу прийому. Пацієнт повинен мати можливість вибрати час прийому, який йому підходить. Застосунок повинен відображати графік роботи лікарів, щоб пацієнти могли побачити, які дні та години прийому доступні; підтвердження запису. Після того, як пацієнт вибрав лікаря та час прийому, він повинен мати можливість підтвердити запис. Після підтвердження застосунок має показати пацієнту актуальні записи до лікарів, які він може відмінити або змінити час прийому.

Ця модель взаємодії користувача є простою та інтуїтивно зрозумілою. Вона дозволяє пацієнтам швидко та легко записатися на прийом до потрібного лікаря.

Опишемо основні дії користувача, як лікаря, в мобільному застосунку для запису до медичних фахівців:

- вхід до застосунку. Лікар використовує свій унікальний обліковий запис для входу до застосунку. Якщо лікар використовує застосунок вперше, він може пройти процедуру реєстрації, надавши необхідну інформацію та створивши свій обліковий запис;

- заповнення Анкети Лікаря. Після входу в акаунт, лікар може заповнити свою анкету, надавши деталі про свою професійну інформацію, спеціалізацію, графік роботи та інші важливі деталі, може також завантажити своє фото; прийняття Записів. Лікар може переглядати запити на запис в свій графік та приймати їх або відхиляти, враховуючи свою наявність та обсяг роботи. Після прийому пацієнта лікар може відмітити прийом у застосунку;

- редагування анкети. Лікар може змінювати додану інформацію про себе, свій графік роботи, вказуючи свій час доступності для запису пацієнтів;

- підтвердження чи скасування заявки на запис від користувача;

- перегляд актуальних статей. Лікар має можливість переглядати актуальні статті, наукові матеріали та інші ресурси, які допомагають йому оновлювати свої знання та залишатися в курсі новин в галузі медицини.

Враховуючи сценарії роботи користувачів з мобільним застосунком для запису до медичних фахівців, модель взаємодії користувача з мобільним застосунком для запису до медичних фахівців подамо у вигляді діаграми прецедентів у нотації UML, зображеної на рис. 2.1.

Таким чином, створено модель взаємодії користувача з мобільним застосунком для запису до медичних фахівців, яка подана у вигляді діаграми прецедентів у нотації UML, та дозволяє розроблювати застосунок для процесу запису користувачів до медичних фахівців.

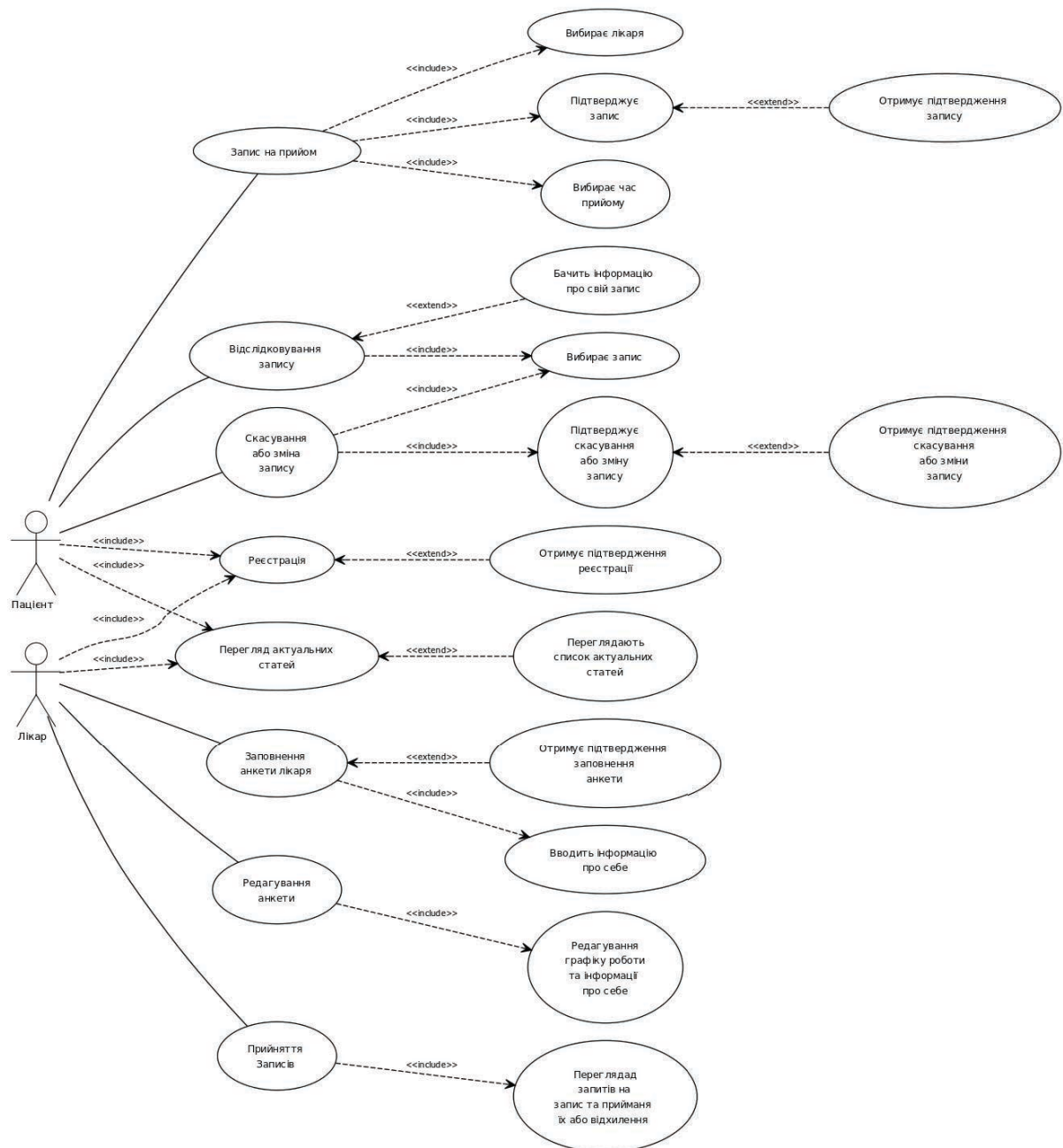


Рисунок 2.1 – Модель взаємодії користувача з мобільним застосунком для запису до медичних фахівців у вигляді діаграми прецедентів у нотації UML

2.4 Серверна частина застосунку

Серверна частина мобільного застосунку для запису до медичних фахівців реалізована на основі хмарної платформи Firebase. Firebase - це платформа для розробки додатків, яка допомагає створювати та розвивати

програми та ігри, які подобаються користувачам. Підтримується Google та користується довірою мільйонів компаній по всьому світу [6]. Він включає в себе базу даних Firestore, сервіси розгортання, а також інструменти для управління користувачами та безпеки.

Для зберігання даних про користувачів, медичних фахівців та записів на прийомі використовується база даних Cloud Firestore. Cloud Firestore - це гнучка, масштабована база даних для мобільних, веб- та серверних розробок на основі Firebase та Google Cloud [7]. Це документо-орієнтована база даних, яка дозволяє зберігати дані у вигляді документів, що складаються з ключів-значень. Це робить її ідеальною для зберігання даних про користувачів, які часто змінюються, наприклад, інформацію про контакти, адресу та медичний статус.

Для управління користувачами використовується сервіс Firebase Authentication. Firebase Authentication - це сервіс, який дозволяє реєструвати користувачів, перевіряти їхню автентичність та управляти їхніми обліковими записами. Він надає внутрішні сервіси, прості у використанні SDK та готові бібліотеки інтерфейсів для автентифікації користувачів у вашому додатку. Вона підтримує автентифікацію за допомогою паролів, телефонних номерів, популярних об'єднаних постачальників ідентифікаційних даних [8].

Для забезпечення безпеки застосунок використовується сервіс Firebase Security. Firebase Security - це набір інструментів, які дозволяють захистити застосунок від атак. Вони забезпечують надійний, повністю настроюваний захист даних у хмарному сховищі Firestore, базі даних та хмарному сховищі [9].

Для зберігання зображень користувачів використовується сервіс Firebase Cloud Storage. Cloud Storage для Firebase - це потужний, простий та економічно ефективний сервіс зберігання об'єктів, створений компанією Google [10]. Використовує ефективну та надійну хмарну інфраструктуру, щоб забезпечити розробникам додатків можливість зберігання та обробки вмісту, створеного користувачем, з високою швидкістю та безпекою.

2.5 Висновки за розділом 2

Для розробки мобільного застосунку для запису до медичних фахівців обрано мову програмування Dart, яка є ефективною для створення мобільних застосунків. Для реалізації проекту використано фреймворк Flutter, що дозволяє швидко створювати зручні та естетичні мобільні додатки.

Створено модель взаємодії користувача з мобільним застосунком для запису до медичних фахівців, яка подана у вигляді діаграми прецедентів в нотатції UML. Ця модель дозволяє здійснювати зручний та ефективний процес запису до лікарів через мобільний застосунок. Користувачі можуть переглядати доступних медичних фахівців, обирати зручний час та дату, а також здійснювати скасування або перенесення запису.

Вибір мови програмування Dart та фреймворку Flutter надає можливість швидкої розробки та одночасно забезпечує високу продуктивність та кросплатформенність для забезпечення доступу до застосунку на різних мобільних платформах.

Для серверної частини мобільного застосунку для запису до медичних фахівців використано хмарну платформу Firebase, Firestore для зберігання даних, Firebase Authentication для аутентифікації користувачів, Firebase Security для забезпечення безпеки та Firebase Storage для зберігання файлів.

3 ОПИС МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Структура мобільного застосунку для запису до медичних фахівців

Структуру мобільного застосунку для запису до медичних фахівців подамо у вигляді діаграм в нотації UML (рис. 3.1 – 3.3).

Загальна діаграма класів мобільного застосунку для запису до медичних фахівців зображена на рис. 3.1.

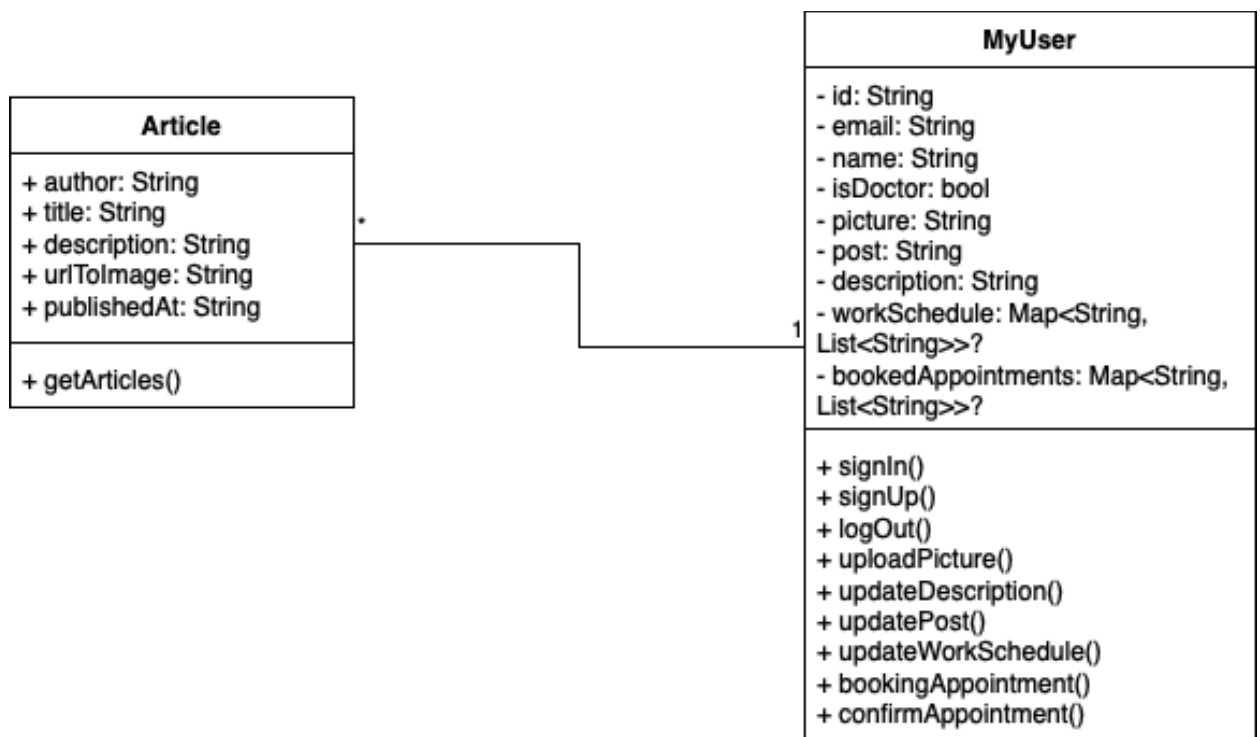


Рисунок 3.1 – Діаграма класів мобільного застосунку для запису до медичних фахівців у нотації UML

Як видно з діаграми, основними класами мобільного застосунку для запису до медичних фахівців є: Користувач (**MyUser**), Стаття (**Article**).

Діаграма пакетів мобільного застосунку для запису до медичних фахівців зображена на рис. 3.2.



Рисунок 3.2 – Діаграма пакетів мобільного застосунку для запису до медичних фахівців у нотації UML

Діаграма компонентів мобільного застосунку для запису до медичних фахівців зображена на рис. 3.3.

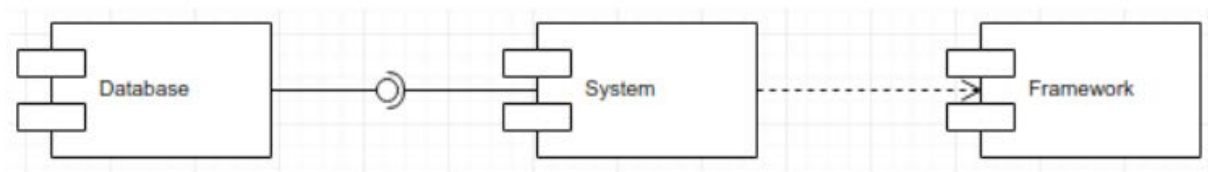


Рисунок 3.3 – Діаграма компонентів мобільного застосунку для запису до медичних фахівців у нотації UML

Як видно з рисунку, основними компонентами мобільного застосунку для запису до медичних фахівців є база даних, програма, фреймворк.

3.2 Функціонування мобільного застосунку для запису до медичних фахівців

Загальну схему функціонування мобільного застосунку для запису до медичних фахівців подамо у вигляді діаграмі станів, зображеної на рис. 3.4.

Діаграма діяльності мобільного застосунку для запису до медичних фахівців зображена на рис. 3.5.

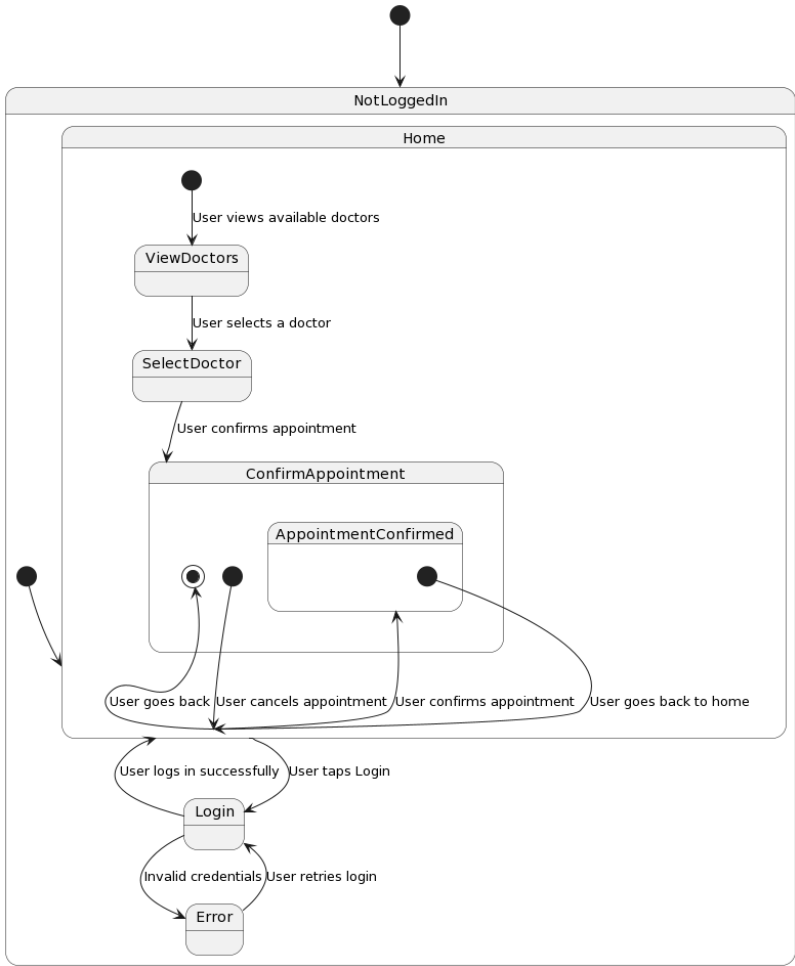


Рисунок 3.4 – Загальна схема функціонування мобільного застосунку для запису до медичних фахівців у вигляді діаграмі станів, поданої у нотації UML

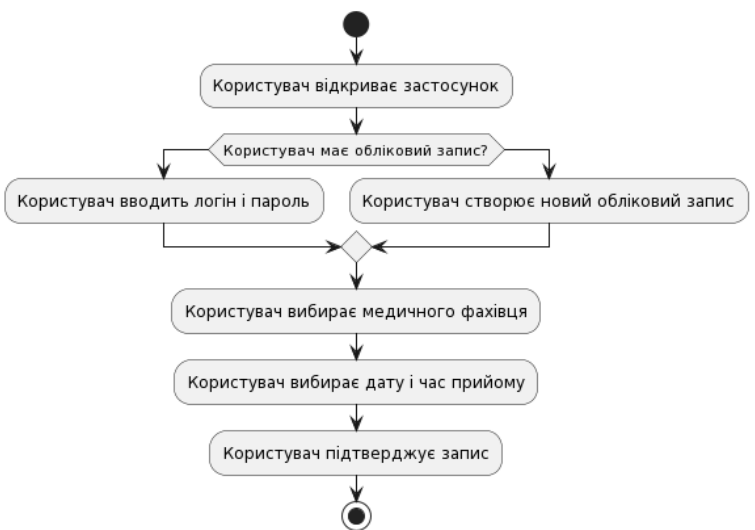


Рисунок 3.5 – Діаграма діяльності мобільного застосунку для запису до медичних фахівців у нотації UML

Діаграма послідовності мобільного застосунку для запису до медичних фахівців у нотації UML зображена на рис. 3.6.

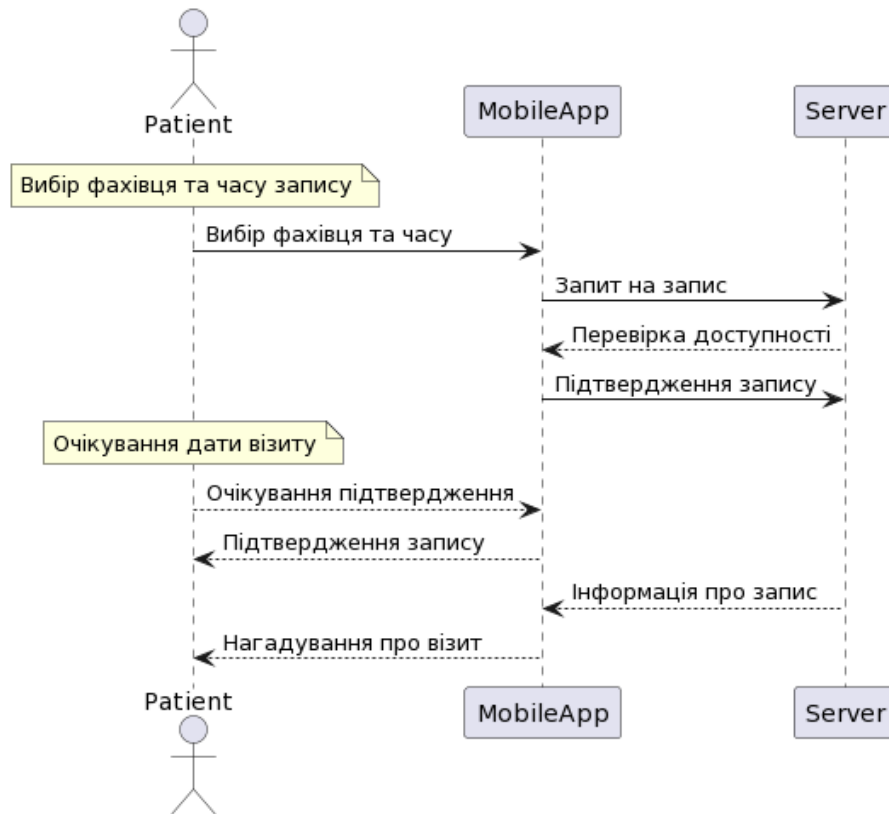


Рисунок 3.6 – Діаграма послідовностей мобільного застосунку для запису до медичних фахівців у нотації UML

3.3 Висновки за розділом 3

Розроблено мобільний застосунок для запису до медичних фахівців. Запропоновано структуру мобільного застосунку для запису до медичних фахівців, яку подано за допомогою структурних діаграм в нотації UML.

Визначено, що основними класами мобільного застосунку для запису до медичних фахівців є: Користувач (MyUser), Стаття (Article).

Описано функціонування мобільного застосунку для запису до медичних фахівців за допомогою моделей у нотації UML основі діаграм поведінки та діаграм послідовності.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

4.1 Призначення й умови застосування мобільного застосунку

Мобільний застосунок для запису до медичних фахівців розроблено з використанням мови Dart та фреймворку Flutter. Забезпечуючи можливість реєстрації як пацієнта, так і лікаря, цей застосунок надає функціонал для зручного та ефективного ведення записів на прийоми та спілкування між медичними фахівцями та пацієнтами в реальному часі.

Мобільний застосунок для запису до медичних фахівців забезпечує виконання таких функцій:

- реєстрація користувача як пацієнта чи лікаря з обов'язковим заповненням основних контактних даних (для незареєстрованих користувачів);
- авторизація користувача;
- запис на прийом (вибір лікаря зі списку доступних медичних фахівців, обрання дати та часу для запису на прийом);
- відслідковування запису (пацієнт може переглядати актуальні записи на прийом та отримувати сповіщення про них);
- скасування та зміна запису (можливість пацієнта скасувати або змінити свій запис, враховуючи вільні години прийому);
- анкета лікаря (лікар може додати свою анкету з основною інформацією та графіком роботи);
- перегляд статей (пацієнт та лікар можуть переглядати актуальні медичні статті прямо в застосунку).

Для експлуатації застосунку для запису до медичних фахівців необхідно мобільний телефон з підтримкою Інтернету.

Функціональні характеристики розробленого мобільного застосунку для запису до медичних фахівців наведено у технічному завданні (додаток А).

4.2 Характеристики мобільного застосунку

Мобільний застосунок для запису до медичних фахівців може працювати цілодобово. Він не вимагає специфічних умов експлуатації, за винятком наявності мобільного пристрою користувача та наявності інтернету. Інтерфейс додатку для запису до медичних фахівців спрощений і не вимагає особливої кваліфікації користувача.

Вхідні дані додатку включають інформацію про користувача, таку як логін та пароль, необхідні для авторизації та ідентифікації пацієнта чи лікаря. Додатково, вхідні дані включають інформацію, необхідну для запису на прийом та відслідковування стану запису.

Вихідні дані додатку включають інформацію про успішну або невдалу реєстрацію, запис на прийом, а також повідомлення про будь-які зміни у записі, які користувач отримує від медичного фахівця. Застосунок сприяє ефективній організації та підтримці запису до медичних фахівців, забезпечуючи зручний інтерфейс для взаємодії між пацієнтами та лікарями.

4.3 Опис роботи з мобільним застосунком та результати тестування

Для використання мобільного застосунку для запису до медичних фахівців користувач повинен встановити його на своєму мобільному пристрої. Після встановлення з'являється головний екран, який дозволяє користувачеві отримати доступ до основних функцій застосунку, він відображений на рис. 4.1. Після його відображення, користувачу необхідно вибрати: авторизуватися (натиснути кнопку «Логін») або зареєструватися (натиснути кнопку «Зареєструватися»).

Для реєстрації (рис. 4.2) користувач повинен ввести своє ім'я, повинен ввести свою електронну пошту (e-mail) в відповідному полі, має створити пароль (password) для захисту свого облікового запису, повторити введення паролю для підтвердження його правильності та уникнення помилок, обрати

свою роль: «Я пацієнт» чи «Я лікар». Це важливо для застосунку, оскільки функціонал облікового запису може відрізнятися в залежності від обраної ролі та після натиснути кнопку «Зареєструватися».

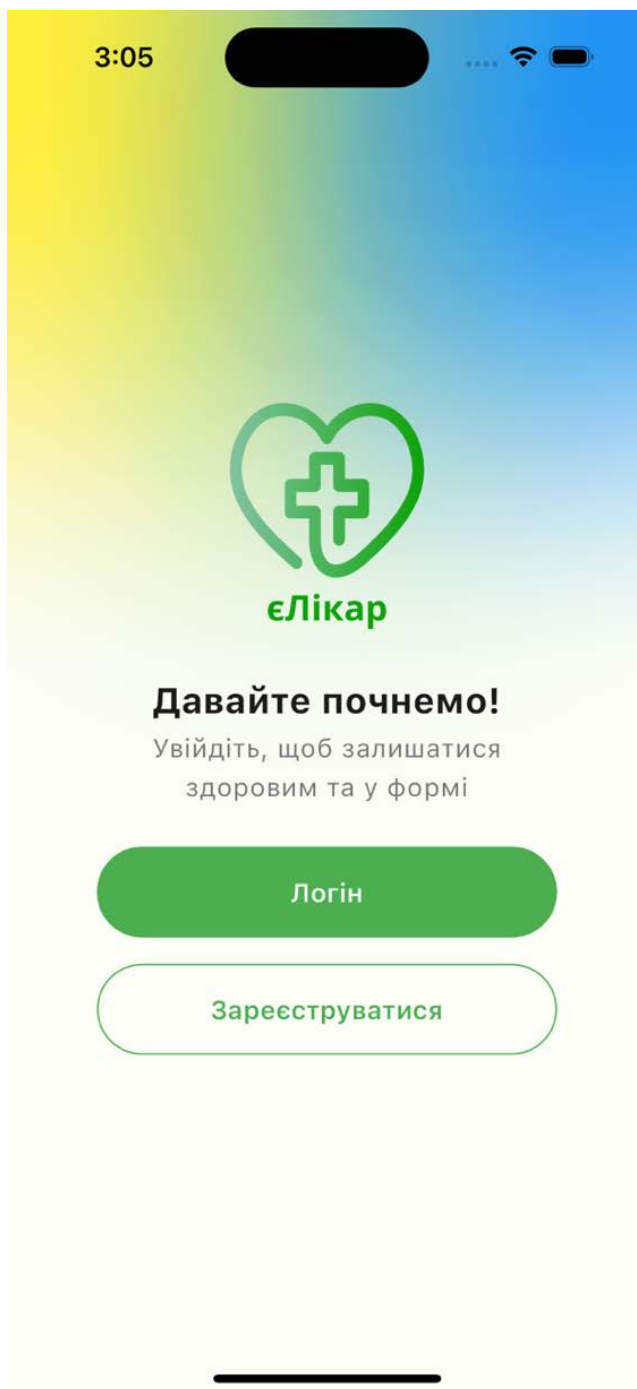


Рисунок 4.1 – Головний екран мобільного застосунку для запису до медичних фахівців

The screenshot shows a mobile application interface for registration. At the top, the status bar displays the time 1:29, a black pill-shaped notch, and icons for cellular signal, Wi-Fi, and battery. The background of the app is a green-to-blue gradient. The title 'Реєстрація' is centered in white. Below it are four white input fields with rounded corners, each containing a gray icon and placeholder text: a person icon for 'Введіть своє ім'я', an envelope icon for 'Введіть свою електронну пошту', a lock icon for 'Введіть свій пароль', and a circular arrow with a lock icon for 'Повторіть свій пароль'. Below these fields are two toggle buttons: 'Я пацієнт' (highlighted in green) and 'Я лікар' (gray). At the bottom is a large green button labeled 'Зареєструватись'. Below this button is the text 'Маєте акаунт? Увійти' with 'Увійти' in green.

Рисунок 4.2 – Екран «Реєстрація» мобільного застосунку для запису до медичних фахівців

Для входу (рис. 4.3) користувач повинен ввести свою електронну пошту (e-mail) та пароль (password) та натиснути кнопку «Увійти».

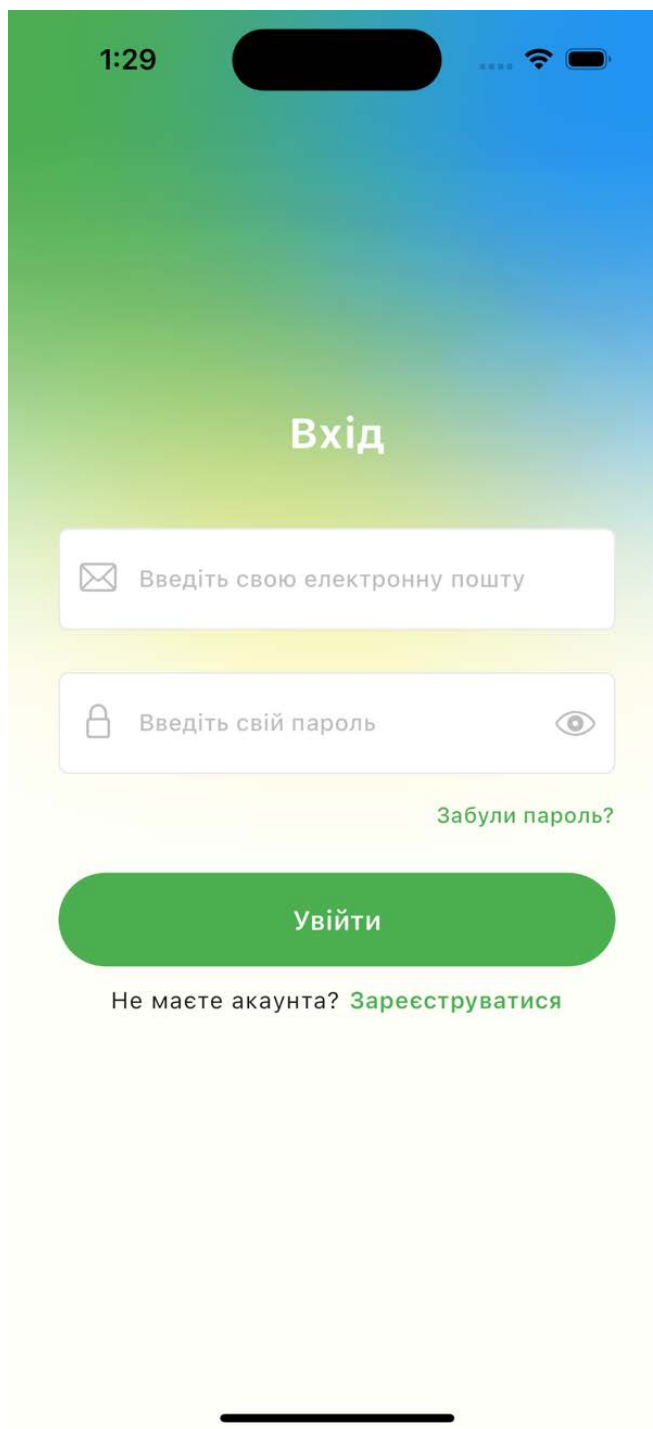


Рисунок 4.3 – Екран «Вхід» мобільного застосунку для запису до медичних фахівців

Після успішної авторизації в мобільному застосунку користувач автоматично перенаправляється на «Домашню сторінку» (рис. 4.4), яка відображає загальну інформацію, посилення на «Всі лікарі» та актуальні статті.

На цій сторінці в відділі «Статті про здоров'я» користувач може знайти корисні матеріали та новини з обраної галузі медицини та ознайомитись більш детально з статтею.

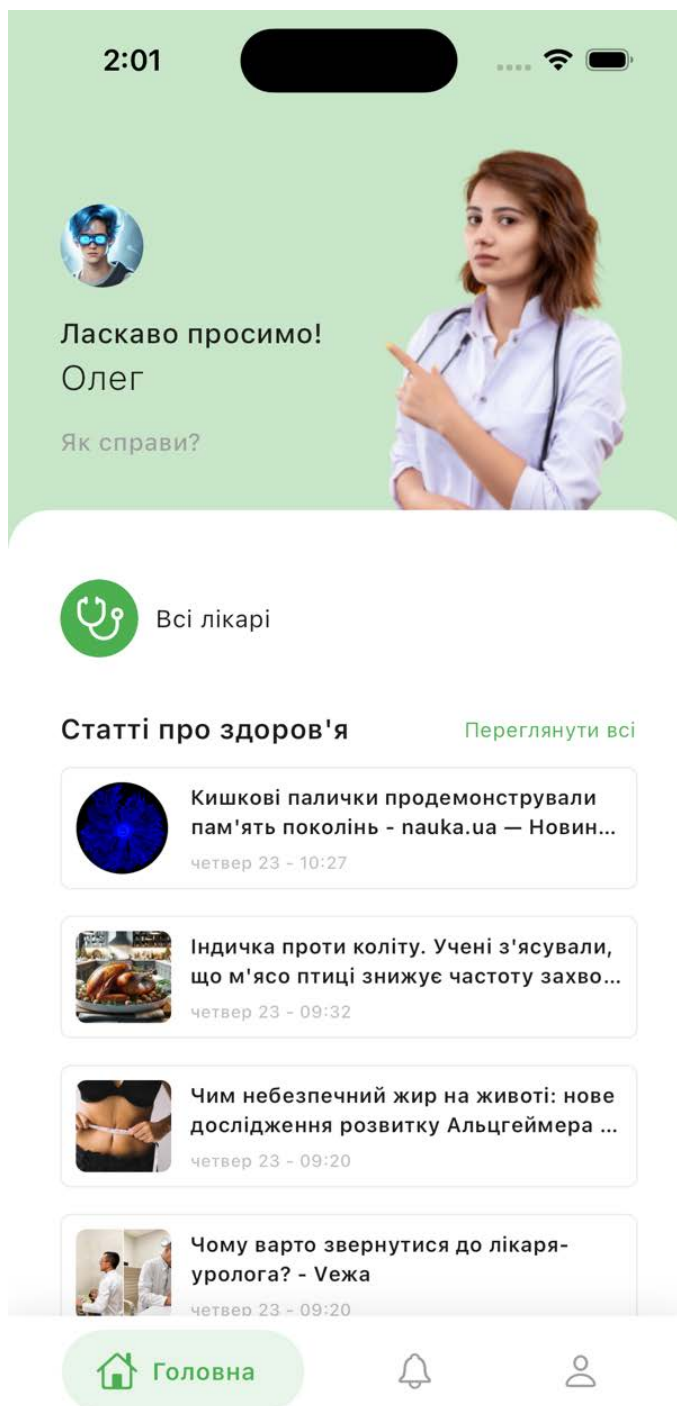


Рисунок 4.4 – Екран «Домашня сторінка» мобільного застосунку для запису до медичних фахівців

За бажанням перейти до екрану «Профіль» (рис. 4.5), користувач натискає на іконку «User» в навігаційній панелі.

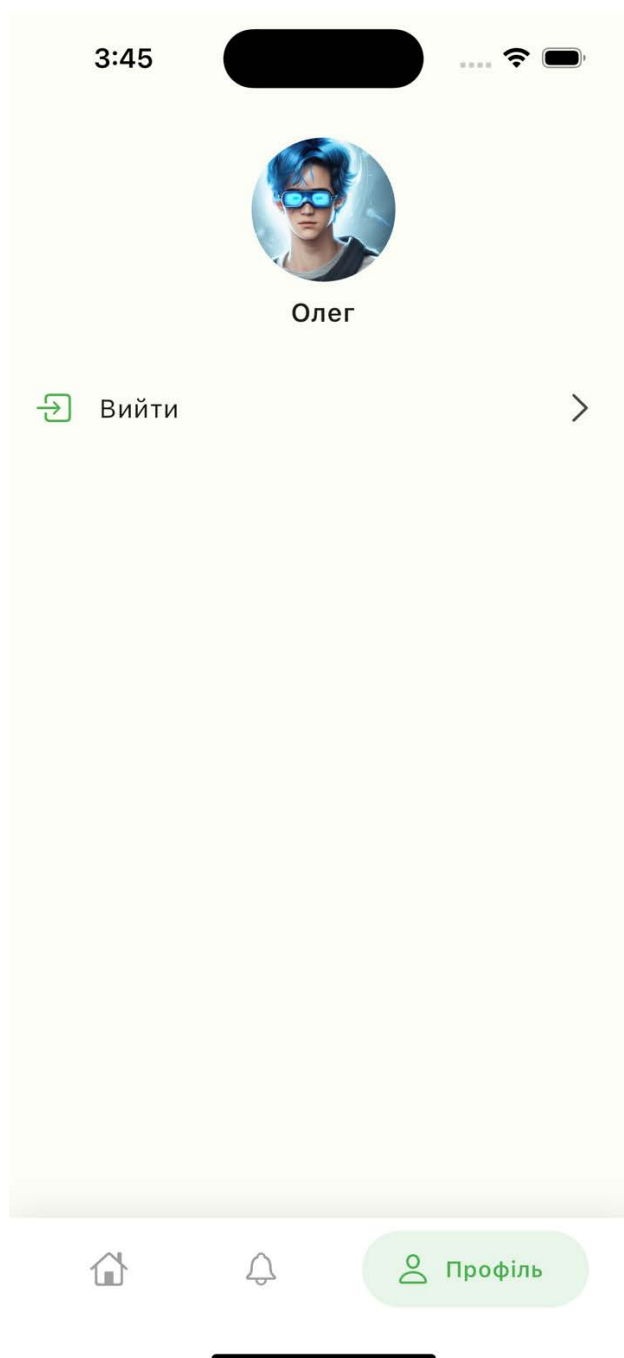


Рисунок 4.5 – Екран «Профіль пацієнта» мобільного застосунку для запису до медичних фахівців

Екран «Профіль лікаря» відображений на рис. 4.6.

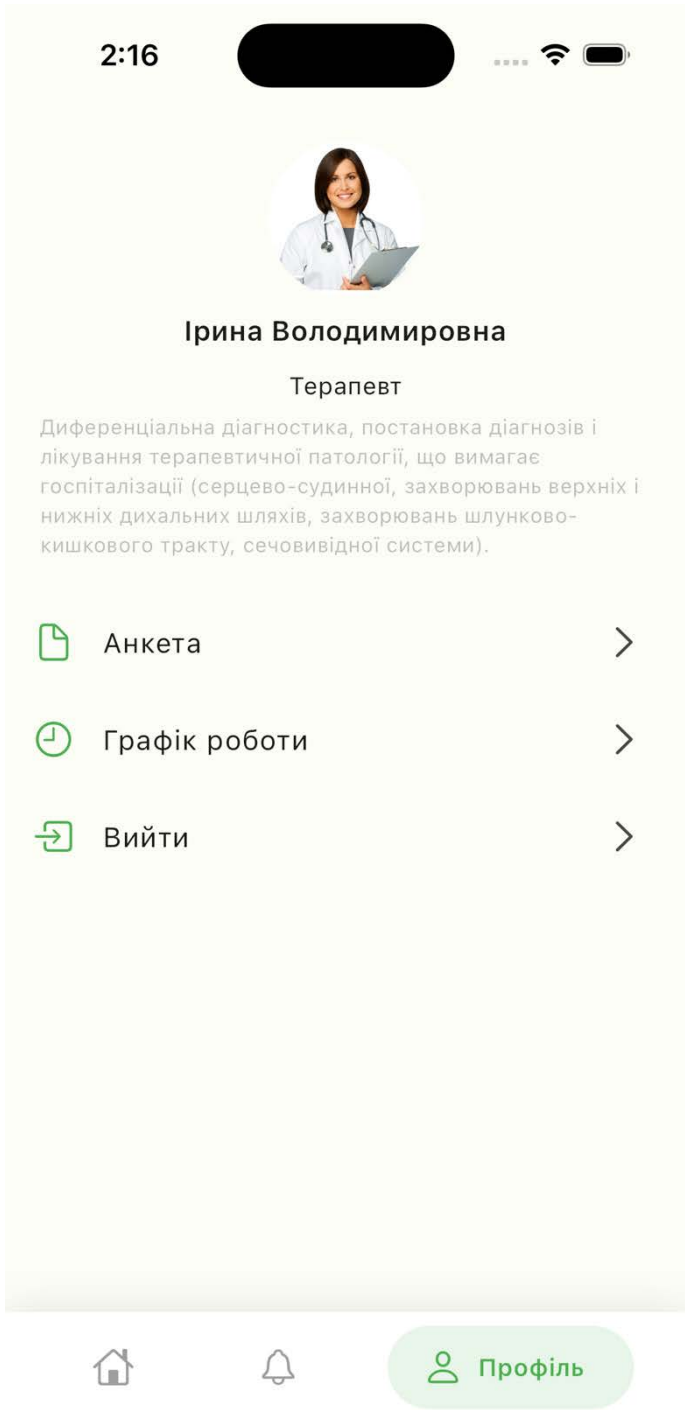


Рисунок 4.6 – Екран «Профіль лікаря» мобільного застосунку для запису до медичних фахівців

Екран для заповненням лікарем його анкети відображений на рис. 4.7.

2:16

Анкета

Оберіть спеціалізацію лікаря:

Терапевт

Інформація про Вас:

Диференціальна діагностика,
постановка діагнозів і лікування
терапевтичної патології, що вимагає
госпіталізації (серцево-судинної,
захворювань верхніх і нижніх
дихальних шляхів, захворювань
шлунково-кишкового тракту,
сечовивідної системи).

Зберегти

Рисунок 4.7 – Екран «Анкета лікаря» мобільного застосунку для запису до медичних фахівців

Екран для заповнення лікарем інформації про його графік роботи відображений на рис. 4.8.

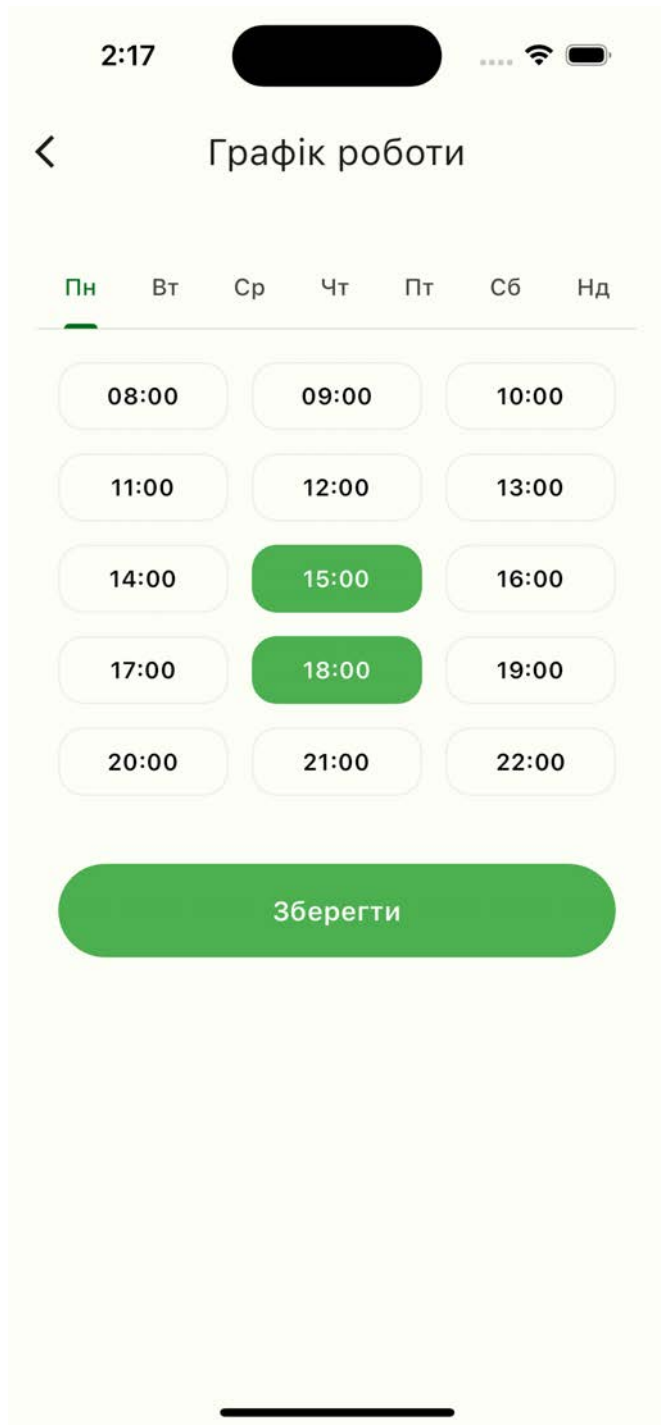


Рисунок 4.8 – Екран «Графік роботи» мобільного застосунку для запису до медичних фахівців

На домашній сторінці (рис. 4.4) користувач може перейти до відділу «Всі лікарі» (рис. 4.9), нажавши на відповідну кнопку. Тут йому доступний повний перелік усіх лікарів, які обслуговують через застосунок. Вибравши конкретного лікаря, користувач має можливість переглядати детальну інформацію про нього.

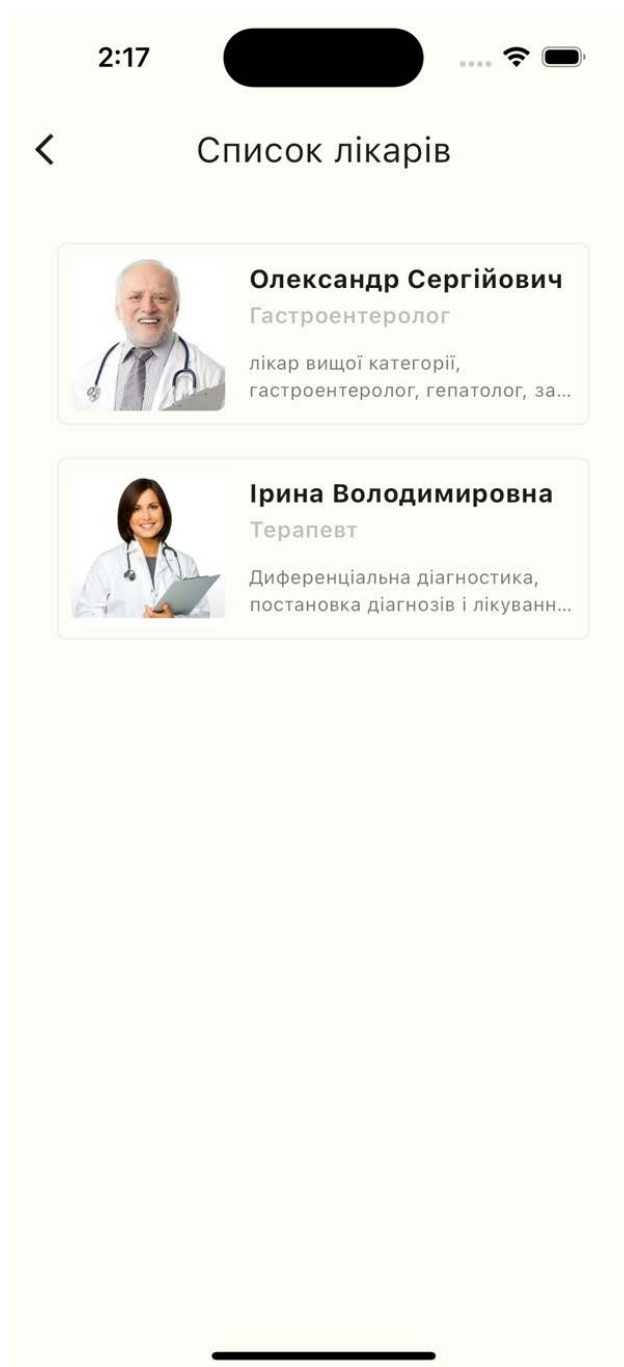


Рисунок 4.9 – Екран «Всі лікарі» мобільного застосунку для запису до медичних фахівців

По кліку на лікаря із списку користувач переходить на сторінку «Деталі лікаря» (рис. 4.10), яка містить повний обзор вибраного лікаря. Тут доступна інформація про освіту, досвід роботи, спеціалізацію та інші релевантні дані. Крім того, користувач може ознайомитись з графіком роботи лікаря, включаючи робочі дні та години прийому та є можливість записатися на прийом, обираючи з доступних годин роботи лікаря.



Рисунок 4.10 – Екран «Профіль лікаря» мобільного застосунку для запису до медичних фахівців

Якщо у лікаря немає вільного часу на обрану дату, відображається відповідна інформація на сторінці (рис. 4.11).



Рисунок 4.11 – Екран «Профіль лікаря при відсутності вільного часу на обрану дату» мобільного застосунку для запису до медичних фахівців

Після успішної реєстрації на обрану годину, користувач отримує підтвердження та нагадування про прийом (рис. 4.12).

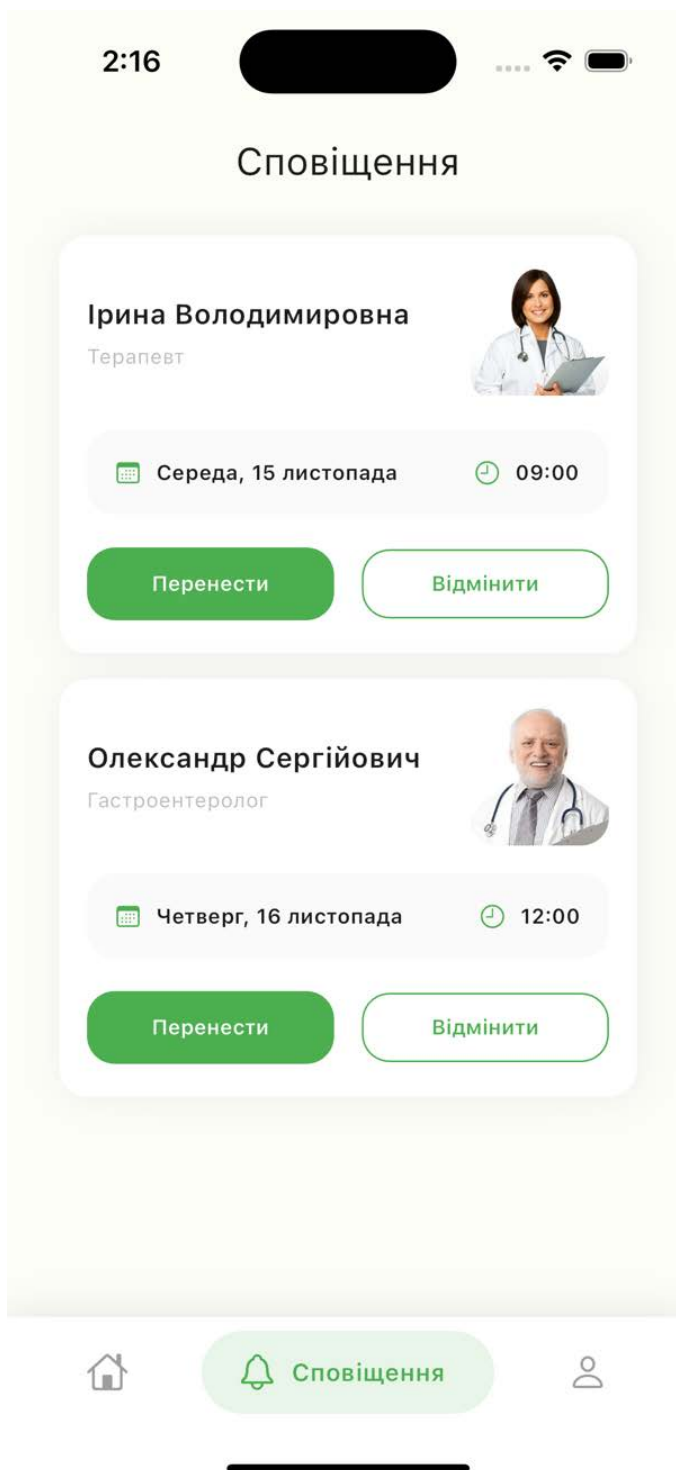


Рисунок 4.12 – Екран «Сповіщення про прийом» мобільного застосунку для запису до медичних фахівців

Сторінку для підтвердження лікарем записів на прийом представлено на рис. 4.13, на якій лікар може прийняти або відмінити запис на прийом від користувача.

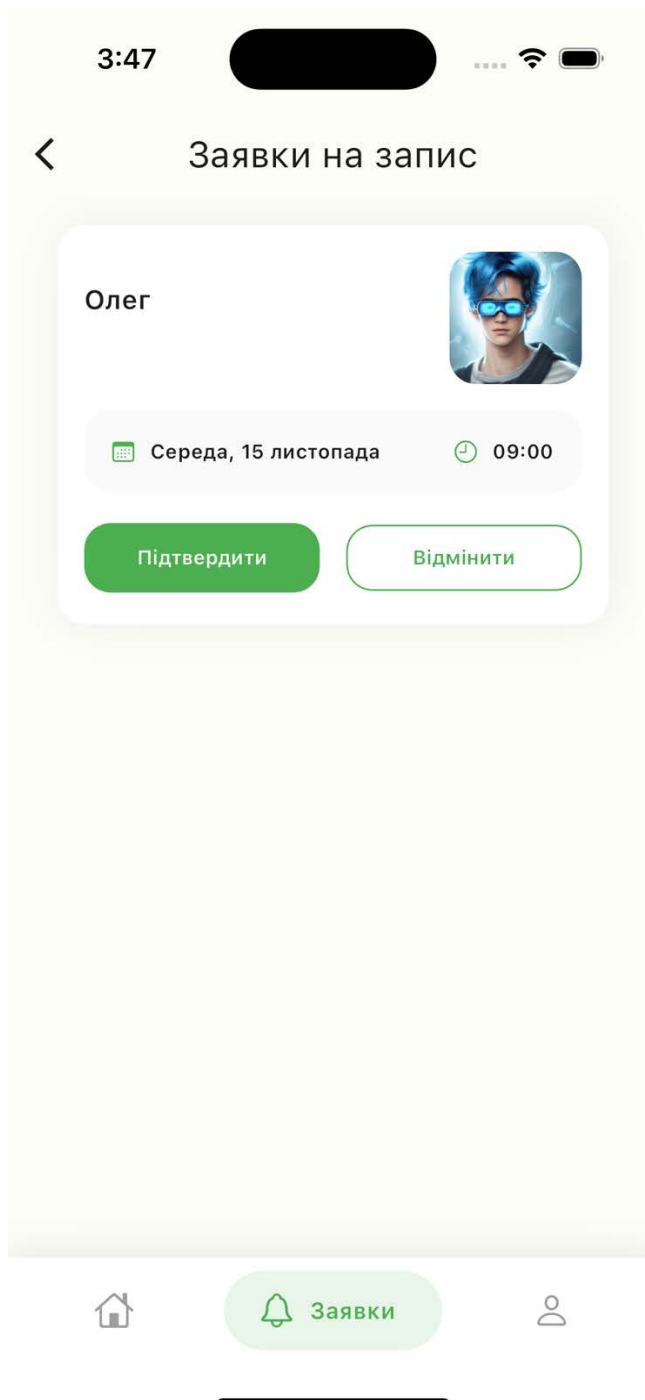


Рисунок 4.13 – Екран «Заявки на запис» мобільного застосунку для запису до медичних фахівців

Для забезпечення високої ефективності та надійності мобільного застосунку для запису до медичних фахівців проведено комплексне тестування його функціональності. Тестовий план включав в себе випробування основних функцій, взаємодії із системними ресурсами, а також сценарії користувачів.

Під час тестування основних функцій мобільного застосунку було перевірено його здатність ефективно реєструвати та управляти записами пацієнтів. Результати тестів підтвердили правильну роботу основних функціональних можливостей застосунку.

Тестування взаємодії мобільного застосунку із системними ресурсами включало аналіз його реакції на зміни в обсязі даних, швидкість обробки запитів та стійкість до навантаження. Застосунок продемонстрував високу стабільність і ефективність у відповідь на різноманітні умови експлуатації.

З метою забезпечення конфіденційності та цілісності медичних даних, проведено тестування безпеки мобільного застосунку. Перевірка на вразливості та шифрування даних були проведені для забезпечення високого рівня захисту від несанкціонованого доступу.

В результаті проведених тестів було виявлено, що мобільний застосунок для запису до медичних фахівців відповідає вимогам високої якості та надійності. Всі функціональні та безпекові аспекти були успішно випробувані, а застосунок продемонстрував стабільну та ефективну роботу під час різноманітних умов експлуатації.

Тестування мобільного застосунку для запису до медичних фахівців підтвердило його готовність до впровадження та використання в реальних умовах. За результатами тестів при роботі під керуванням різних мобільних операційних систем, показали, що застосунок для запису до медичних фахівців функціонує коректно та відповідає вимогам технічного завдання.

4.4 Висновки за розділом 4

У результаті проведеного дослідження та програмної реалізації мобільного застосунку для запису до медичних фахівців можна зробити висновок, що він був створений з метою оптимізації процесу запису до медичних фахівців. Проведене тестування відбувалося у різних умовах, включаючи різні мобільні операційні системи. Результати тестування

підтверджують стабільну та ефективну роботу мобільного застосунку, відповідно до вимог технічного завдання. Отримані дані свідчать про високий рівень функціональності та надійності розробленого застосунку.

Значущим аспектом є те, що застосунок успішно пройшов тестування на різних платформах, що підкреслює його універсальність та готовність до використання широким колом користувачів. На підставі здійснених досліджень можна також відзначити високу якість та ефективність розробленого додатку.

Отже, узагальнюючи, можна стверджувати, що дослідження та програмна реалізація мобільного застосунку для запису до медичних фахівців на базі Dart і Flutter були успішно завершені, забезпечивши високу якість, ефективність та готовність до широкого впровадження додатку.

ВИСНОВКИ

В ході виконання дипломної роботи з дослідження та програмної реалізації застосунку для запису до медичних фахівців було проведено комплексний аналіз та дослідження процесу розробки застосунку.

Досліджено сучасні тенденції у галузі медичного обслуговування та виявлено необхідність у високоефективному та зручному для користувачів засобі для запису до медичних фахівців. Визначено, що розробка мобільного додатку є важливим етапом в покращенні доступності та ефективності медичних послуг.

Вивчено основні характеристики мови програмування Dart та встановлено її придатність для створення переносних та швидкодіючих додатків. Обрано фреймворк Flutter як оптимальний інструмент для розробки кросплатформених мобільних додатків, що забезпечують однакову ефективність на мобільних операційних системах Android та iOS.

Детально розглянуто функціональні вимоги до розробки додатку для запису до медичних фахівців, визначено його основні складові та функції.

Враховуючи результати аналізу, обрано інструментарій для програмної реалізації застосунку, а саме мову програмування Dart та фреймворк Flutter. Цей вибір базується на їхній ефективності, можливості кросплатформенної розробки та забезпеченні високого рівня користувацької зручності. Вони є найбільш оптимальним для забезпечення швидкості та ефективності взаємодії з додатком.

Запропоновано структуру мобільного застосунку для запису до медичних фахівців, яку подано за допомогою структурних діаграм в нотації UML.

Описано функціонування мобільного застосунку для запису до медичних фахівців за допомогою моделей у нотації UML основі діаграм поведінки та діаграм послідовності.

Зроблено опис роботи користувача з мобільним застосунком для запису до медичних фахівців.

В ході тестування розробленого мобільного застосунку для запису до медичних фахівців виявлено високу його ефективність та відповідність вимогам технічного завдання. Програма успішно функціонує на різних мобільних операційних системах.

Практичне значення роботи визначається створенням мобільного застосунку для записів до медичних фахівців. Розроблений застосунок виявляється ефективним та зручним для користувачів, сприяючи оптимізації процесів обслуговування пацієнтів та удосконаленню системи медичного обслуговування. В цілому, результати дослідження та програмної реалізації свідчать про успішну розробку та впровадження системи для ефективного управління записами до медичних фахівців.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. HELSI - інформаційна система для пацієнтів [Електронний ресурс]. – Режим доступу: <https://helsi.me/>.
2. MeetDoc - Appointment Booking [Електронний ресурс]. – Режим доступу: <https://apps.apple.com/ua/app/meetdoc-appointment-booking/id1540476691>.
3. Сервіс запису до лікаря на прийом | DOC UA [Електронний ресурс]. – Режим доступу: <https://doc.ua/>.
4. Flutter [Electronic resource]. – Access mode: <https://uk.wikipedia.org/wiki/Flutter>.
5. Flutter (software) [Electronic resource]. – Access mode: [https://en.wikipedia.org/wiki/Flutter_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software)).
6. Firebase | Google's Mobile and Web App Development Platform [Electronic resource]. – Access mode: <https://firebase.google.com/>.
7. Firestore | Firebase [Electronic resource]. – Access mode: <https://firebase.google.com/docs/firestore>.
8. Firebase Authentication [Electronic resource]. – Access mode: <https://firebase.google.com/docs/auth>.
9. Firebase Security Rules [Electronic resource]. – Access mode: <https://firebase.google.com/docs/rules>.
10. Cloud Storage for Firebase [Electronic resource]. – Access mode: <https://firebase.google.com/docs/storage>.

ДОДАТОК А
Технічне завдання

Вступ

Мобільний застосунок для запису до медичних фахівців призначений для створення зручного та ефективного інструменту, який дозволяє користувачам легко та швидко записуватися на консультації та прийоми до медичних фахівців.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація застосунку для запису до медичних фахівців», затверджене наказом Національного університету «Запорізька політехніка» № 437 від 14 листопада 2023 р.

А.2 Призначення розробки

Мобільний застосунок для запису до медичних фахівців призначено для полегшення процесу запису на консультації та прийоми, забезпечуючи зручний інтерфейс для користувачів. Застосунок дозволяє пацієнтам швидко знаходити доступних лікарів, обирати зручний час та місце, а також отримувати нагадування про майбутні візити.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Мобільний застосунок для запису до медичних фахівців повинен забезпечити виконання таких функцій:

- реєстрація пацієнта чи лікаря;
- запис на прийом до лікаря;
- відслідковування пацієнтом запису на прийом;

- скасування або зміна пацієнтом запису;
- заповнення лікарем анкети (основна інформація, посада, графік роботи);
- підтвердження заявки на запис лікарем чи скасування;
- перегляд актуальних статей пацієнтом чи лікарем;
- інформація у пацієнта про актуальні записи, можливість скасування або зміни запису.

A.3.2 Вимоги до інтерфейсу програми користувачів

Інтерфейс мобільного застосунку для запису до медичних фахівців повинен бути інтуїтивно зрозумілим, легким у використанні та забезпечувати зручність для користувачів. Навігація має бути простою, а дизайн привабливим, сприяючи комфортному процесу запису на прийом до лікаря.

A.3.3 Вимоги до надійності

Мобільний застосунок для запису до медичних фахівців повинен гарантувати надійність у роботі, особливо у випадках критичних ситуацій. Забезпечення стійкості та ефективної обробки інформації є обов'язковим. Крім того, система повинна здати на велику кількість користувачів, забезпечуючи швидку та безперебійну реєстрацію на прийом до медичних фахівців.

A.3.4 Умови експлуатації

Мобільний застосунок для запису до медичних фахівців повинен бути ефективним і не вимагати спеціальних умов експлуатації. Користувачі повинні мати можливість здійснювати запис в будь-який час та з будь-якого місця, необхідна тільки доступність до Інтернету.

A.3.5 Вимоги до складу та параметрів технічних засобів

Для експлуатації застосунку для запису до медичних фахівців необхідний мобільний телефон з операційною системою Android чи iOS та можливістю входу в Інтернет.

A.3.6 Вимоги до маркування і пакування

ПЗ для запису до медичних фахівців може бути поширене на флеш-накопичувачах або інших засобах зберігання інформації. На пакуванні необхідно зазначити назву програми: «Мобільний застосунок для запису до медичних фахівців». Також слід вказати основні функції та корисність застосунку для користувачів.

A.4 Вхідні дані до роботи

Вхідними даними для мобільного застосунку, призначеного для запису до медичних фахівців, є інформація користувача (електронна пошта, пароль). Крім того, вхідні дані включають бажаний час та дату для запису.

Вихідними даними програми є підтвердження запису користувача на обраний час та дату, інформація про доступність обраного медичного фахівця та інструкції щодо подальших кроків. У разі, якщо обраного часу неможливо забезпечити, програма інформує про це повідомленням користувачу.

ДОДАТОК Б
Текст програми

B.1 sign_in_screen.dart

```
import 'dart:ui';

import
'package:edocor_app/blocs/authentication_bloc/authentication_bloc.dart';
import 'package:edocor_app/blocs/sign_in_bloc/sign_in_bloc.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';

import '../components/components.dart';
import 'components/components.dart';

class SignInScreen extends StatefulWidget {
  const SignInScreen({Key? key}) : super(key: key);

  @override
  State<SignInScreen> createState() => _SignInScreenState();
}

class _SignInScreenState extends State<SignInScreen> {
  final emailController = TextEditingController();
  final passwordController = TextEditingController();

  IconData iconPassword = CupertinoIcons.eye;
  bool obscurePassword = true;
  String? _errorMsg;
  final _formKey = GlobalKey<FormState>();
  bool signInRequired = false;

  @override
  Widget build(BuildContext context) {
    return BlocListener<SignInBloc, SignInState>(
      listener: (context, state) {
        if (state is SignInSuccess) {
          setState(() {
            signInRequired = false;
          });
          Navigator.pushNamed(context, '/');
        } else if (state is SignInProcess) {
          setState(() {
            signInRequired = true;
          });
        } else if (state is SignInFailure) {
          setState(() {
            signInRequired = false;
            _errorMsg = 'Невірний email або пароль.';
          });
        }
      },
      child: Scaffold(
```

```

body: SingleChildScrollView(
  child: SizedBox(
    height: MediaQuery.of(context).size.height,
    child: Stack(
      children: [
        Align(
          alignment: const AlignmentDirectional(20, -1.2),
          child: Container(
            height: MediaQuery.of(context).size.width,
            width: MediaQuery.of(context).size.width,
            decoration: const BoxDecoration(
              shape: BoxShape.circle,
              color: Colors.yellow,
            ),
          ),
        ),
        Align(
          alignment: const AlignmentDirectional(-2.7, -1.2),
          child: Container(
            height: MediaQuery.of(context).size.width / 1.3,
            width: MediaQuery.of(context).size.width / 1.3,
            decoration: const BoxDecoration(
              shape: BoxShape.circle, color: Colors.green),
          ),
        ),
        Align(
          alignment: const AlignmentDirectional(2.7, -1.2),
          child: Container(
            height: MediaQuery.of(context).size.width / 1.3,
            width: MediaQuery.of(context).size.width / 1.3,
            decoration: const BoxDecoration(
              shape: BoxShape.circle, color: Colors.blue),
          ),
        ),
        BackdropFilter(
          filter: ImageFilter.blur(sigmaX: 100.0, sigmaY: 100.0),
          child: Container(),
        ),
        Align(
          alignment: Alignment.center,
          child: SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                SingleChildScrollView(
                  child: SafeArea(
                    child: Column(
                      children: [
                        Padding(
                          padding: const EdgeInsets.all(30.0),
                          child: Column(
                            children: [

```

```

Form(
  key: _formKey,
  child: Column(
    children: [
      const Text(
        'Вхід',
        style: TextStyle(
          color: Colors.white,
          fontSize: 25,
          height: 1.35,
          fontWeight: FontWeight.bold,
        ),
      ),
      const SizedBox(
        height: 40,
      ),
      MyTextField(
        controller: emailController,
        hintText:
          'Введіть свою електронну
пошту',
        obscureText: false,
        keyboardType:
          TextInputType.emailAddress,
        prefixIcon: Icon(
          CupertinoIcons.mail,
          color: Colors.grey[400],
          size: 24,
        ),
        errorMsg: _errorMsg,
        validator: (val) {
          if (val!.isEmpty) {
            return 'Будь ласка,
заповніть це поле';
          } else if (!RegExp(
            r'^[\w-\.]{{2,4}}$'
          ).hasMatch(val)) {
            return 'Будь ласка,
введіть правильний e-mail';
          }
          return null;
        }
      ),
      const SizedBox(
        height: 25,
      ),
      MyTextField(
        controller: passwordController,
        hintText: 'Введіть свій
пароль',
        obscureText: obscurePassword,
        keyboardType:

```

TextInputType.visiblePassword,

CupertinoIcons.eye;

заповніть це поле';

```

prefixIcon: Icon(
  CupertinoIcons.lock,
  color: Colors.grey[400],
  size: 24,
),
suffixIcon: IconButton(
  onPressed: () {
    setState(() {
      obscurePassword =
        !obscurePassword;
      if (obscurePassword) {
        iconPassword =

      } else {
        iconPassword =
          CupertinoIcons
            .eye_slash;
      }
    });
  },
  icon: Icon(
    iconPassword,
    color: Colors.grey[400],
    size: 24,
  ),
),
validator: (val) {
  if (val!.isEmpty) {
    return 'Будь ласка,

  }
  return null;
},
),
const SizedBox(
  height: 15,
),
Row(
  mainAxisAlignment:
    MainAxisAlignment.end,
  children: [
    GestureDetector(
      onTap: () {},
      child: const Text(
        'Забули пароль?',
        style: TextStyle(
          fontSize: 13,
          fontWeight:
            FontWeight.w500,
          color: Colors.green),

```

```

        ),
      ),
    ],
  ),
  const SizedBox(
    height: 25,
  ),
  SizedBox(
    child: !signInRequired
      ? MyButton(
        onPressed: () {
          if (_formKey
            .currentState!
            .validate()) {
            context
              .read<
                SignInBloc>()
                .add(
                  SignInRequired(
                    emailController
                      .text,
                    passwordController
                      .text,
                  ),
                );
          }
        },
        text: 'Увійти',
      )
      : const Center(
        child:
          CircularProgressIndicator(),
        ),
      ),
  const SizedBox(
    height: 10,
  ),
  Row(
    mainAxisAlignment:
      MainAxisAlignment.center,
    children: [
      const Text(
        'Не маєте акаунта?',
        style: TextStyle(
          letterSpacing: 0.5),
      ),
      const SizedBox(
        width: 5,

```


B.2 sign_up_screen.dart

```
import 'dart:ui';

import
'package:edocter_app/blocs/authentication_bloc/authentication_bloc.dart';
import 'package:edocter_app/blocs/sign_up_bloc/sign_up_bloc.dart';
import 'package:edocter_app/repositories/user/models/my_user.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:toggle_switch/toggle_switch.dart';

import '../components/components.dart';
import 'components/components.dart';

class SignUpScreen extends StatefulWidget {
  const SignUpScreen({Key? key}) : super(key: key);

  @override
  State<SignUpScreen> createState() => _SignUpScreenState();
}

class _SignUpScreenState extends State<SignUpScreen> {
  final nameController = TextEditingController();
  final emailController = TextEditingController();
  final passwordController = TextEditingController();
  final repeatPasswordController = TextEditingController();

  bool isChecked = false;
  bool isDoctor = false;

  final _formKey = GlobalKey<FormState>();
  IconData iconPassword = CupertinoIcons.eye;
  bool obscurePassword = true;
  String? _errorMsg;
  bool signUpRequired = false;

  @override
  Widget build(BuildContext context) {
    return BlocListener<SignUpBloc, SignUpState>(
      listener: (context, state) {
        if (state is SignUpSuccess) {
          setState(() {
            signUpRequired = false;
          });
          Navigator.pushNamed(context, '/');
        } else if (state is SignUpProcess) {
          setState(() {
            signUpRequired = true;
          });
        }
      },
    );
  }
}
```

```

    } else if (state is SignUpFailure) {
      return;
    }
  },
  child: Scaffold(
    body: SingleChildScrollView(
      child: SizedBox(
        height: MediaQuery.of(context).size.height,
        child: Stack(
          children: [
            Align(
              alignment: const AlignmentDirectional(20, -1.2),
              child: Container(
                height: MediaQuery.of(context).size.width,
                width: MediaQuery.of(context).size.width,
                decoration: const BoxDecoration(
                  shape: BoxShape.circle,
                  color: Colors.yellow,
                ),
              ),
            ),
            Align(
              alignment: const AlignmentDirectional(-2.7, -1.2),
              child: Container(
                height: MediaQuery.of(context).size.width / 1.3,
                width: MediaQuery.of(context).size.width / 1.3,
                decoration: const BoxDecoration(
                  shape: BoxShape.circle, color: Colors.green),
              ),
            ),
            Align(
              alignment: const AlignmentDirectional(2.7, -1.2),
              child: Container(
                height: MediaQuery.of(context).size.width / 1.3,
                width: MediaQuery.of(context).size.width / 1.3,
                decoration: const BoxDecoration(
                  shape: BoxShape.circle, color: Colors.blue),
              ),
            ),
          ],
        ),
      ),
      BackdropFilter(
        filter: ImageFilter.blur(sigmaX: 100.0, sigmaY: 100.0),
        child: Container(),
      ),
      Align(
        alignment: Alignment.center,
        child: SizedBox(
          height: MediaQuery.of(context).size.height,
          child: SafeArea(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                SingleChildScrollView(
                  child: Padding(

```

```
padding: const EdgeInsets.all(30.0),
child: Form(
  key: _formKey,
  child: Column(
    children: [
      Column(
        children: [
          const Text(
            'Реєстрація',
            style: TextStyle(
              color: Colors.white,
              fontSize: 25,
              height: 1.35,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(
            height: 40,
          ),
          MyTextField(
            controller: nameController,
            hintText: 'Введіть своє ім\'я',
            obscureText: false,
            keyboardType: TextInputType.name,
            prefixIcon: Icon(
              CupertinoIcons.person,
              color: Colors.grey[400],
              size: 24,
            ),
            errorMsg: _errorMsg,
            validator: (val) {
              if (val!.isEmpty) {
                return 'Будь ласка, заповніть
це поле';
              }
              return null;
            },
          ),
          const SizedBox(
            height: 25,
          ),
          MyTextField(
            controller: emailController,
            hintText:
            'Введіть свою електронну
пошту',
            obscureText: false,
            keyboardType:
              TextInputType.emailAddress,
            prefixIcon: Icon(
              CupertinoIcons.mail,
              color: Colors.grey[400],
              size: 24,
```

це поле';

]+.)([\w-]{2,4}\$')

правильний e-mail';

це поле';

repeatPasswordController,

```

),
errorMsg: _errorMsg,
validator: (val) {
  if (val!.isEmpty) {
    return 'Будь ласка, заповніть

  } else if (!RegExp(
    r'^[\w-\.]@([\w-

    .hasMatch(val)) {
    return 'Будь ласка, введіть

  }
  return null;
},
),
const SizedBox(
  height: 25,
),
MyTextField(
  controller: passwordController,
  hintText: 'Введіть свій пароль',
  obscureText: obscurePassword,
  keyboardType:
    TextInputType.visiblePassword,
  prefixIcon: Icon(
    CupertinoIcons.lock,
    color: Colors.grey[400],
    size: 24,
  ),
  validator: (val) {
    if (val!.isEmpty) {
      return 'Будь ласка, заповніть

    }
    return null;
  },
),
const SizedBox(
  height: 25,
),
MyTextField(
  controller:

  hintText: 'Повторіть свій пароль',
  obscureText: obscurePassword,
  keyboardType:
    TextInputType.visiblePassword,
  prefixIcon: Icon(
    CupertinoIcons.lock_rotation,
    color: Colors.grey[400],
    size: 24,
  ),
),

```

це поле';

```

validator: (val) {
  if (val!.isEmpty) {
    return 'Будь ласка, заповніть

  } else if (val !=
    passwordController.text) {
    return 'Паролі не співпадають';
  }
  return null;
},
),
const SizedBox(
  height: 25,
),
ToggleSwitch(
  minWidth: MediaQuery.of(context)
    .size
    .width /
    3,
  initialLabelIndex: 0,
  animate: true,
  curve: Curves.linear,
  animationDuration: 300,
  cornerRadius: 20.0,
  activeFgColor: Colors.white,
  inactiveBgColor: Colors.grey[400],
  inactiveFgColor: Colors.white,
  totalSwitches: 2,
  labels: const [
    'Я пацієнт',
    'Я лікар'
  ],
  icons: const [
    CupertinoIcons.person,
    CupertinoIcons.person_add,
  ],
  activeBgColors: const [
    [Colors.green],
    [Colors.green]
  ],
  onToggle: (index) {
    if (index == 1) {
      isDoctor = true;
    } else {
      isDoctor = false;
    }
  },
),
const SizedBox(
  height: 50,
),
SizedBox(
  child: !signupRequired

```

```

? MyButton(
  onPressed: () {
    if
      .validate()) {
        MyUser myUser =
          MyUser.empty;
        myUser =
          email:
            .text,
            name:
              isDoctor: isDoctor,
            );
        setState(() {
          context
        .read<SignUpBloc>()
        .add(SignUpRequired(
          myUser,
          .text));
        });
      },
    text: 'Зареєструватись',
  )
: const Center(
  child:
    ),
),
const SizedBox(
  height: 10,
),
Row(
  mainAxisAlignment:
    MainAxisAlignment.center,
  children: [
    const Text(
      'Маєте акаунт?',
      style:
        TextStyle(letterSpacing:
0.5),
    ),
    const SizedBox(

```


B3. home_screen.dart

```

import 'package:edocor_app/blocs/my_user_bloc/my_user_bloc.dart';
import
'package:edocor_app/blocs/update_user_info_bloc/update_user_info_bloc.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:flutter_svg/svg.dart';

import '../profile/components/components.dart';
import 'components/articles_block.dart';

class HomeScreen extends StatefulWidget {
  HomeScreen({Key? key}) : super(key: key);

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return BlocListener<UpdateUserInfoBloc, UpdateUserInfoState>(
      listener: (context, state) {
        if (state is UploadPictureSuccess) {
          setState(() {
            context.read<MyUserBloc>().state.user!.picture = state.userImage;
          });
        }
      },
      child: Scaffold(
        body: SingleChildScrollView(
          child: BlocBuilder<MyUserBloc, MyUserState>(
            builder: (context, state) {
              if (state.status == MyUserStatus.success) {
                final picture = state.user!.picture;
                final name = state.user!.name;
                return Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Container(
                      padding: EdgeInsets.fromLTRB(
                        30, MediaQuery.of(context).padding.top + 15, 30, 0),
                      decoration: BoxDecoration(
                        color: Colors.green[100],
                      ),
                    ),
                    child: Row(
                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
                      children: [
                        Flexible(
                          child: Column(
                            crossAxisAlignment: CrossAxisAlignment.start,

```

```

children: [
  picture == null || picture.isEmpty
    ? ClipRRect(
      borderRadius:
BorderRadius.circular(48),

      child: Container(
        width: 48,
        height: 48,
        decoration: BoxDecoration(
          color: Colors.grey[200],
        ),
        child: Icon(
          CupertinoIcons.person,
          color: Colors.grey[700],
        ),
      ),
    )
  : Container(
    width: 48,
    height: 48,
    decoration: BoxDecoration(
      color: Colors.grey,
      shape: BoxShape.circle,
      image: DecorationImage(
        image: NetworkImage(
          picture,
        ),
        fit: BoxFit.cover),
    ),
  ),
  const SizedBox(
    height: 15,
  ),
  const Text(
    'Ласкаво просимо!',
    style: TextStyle(
      fontSize: 16,
      fontWeight: FontWeight.w500,
    ),
  ),
  Text(
    name,
    style: const TextStyle(
      fontSize: 20,
      fontWeight: FontWeight.w300,
    ),
  ),
  const SizedBox(
    height: 12,
  ),
  Text(
    "Як справи?",
    style: TextStyle(

```

```

fontWeight: FontWeight.w500,
color: Colors.grey[500],
),
),
const SizedBox(
height: 15,
),
],
),
),
Image.asset(
'assets/doctor.png',
width: 160,
height: 239,
)
],
),
),
Container(
transform: Matrix4.translationValues(0, -24, 0),
width: MediaQuery.of(context).size.width,
padding: const EdgeInsets.fromLTRB(30, 20, 30, 20),
decoration: const BoxDecoration(
color: Colors.white,
borderRadius: BorderRadius.only(
topLeft: Radius.circular(24),
topRight: Radius.circular(24),
),
),
child: Column(
children: [
SizedBox(
child: state.user!.post == null
? GestureDetector(
onTap: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) =>
const FormTile()));
},
child: const Row(
children: [
Icon(CupertinoIcons.pencil),
SizedBox(
width: 15,
),
Flexible(
child: Text(
'Заповніть анкету в профілі для
відображення Вас в списку лікарів.'),
),
],

```

```

        ),
        : Container(),
    ),
    const SizedBox(
      height: 20,
    ),
    GestureDetector(
      onTap: () {
        Navigator.pushNamed(context, '/doctors');
      },
      child: Row(
        children: [
          Container(
            width: 45,
            height: 45,
            decoration: BoxDecoration(
              color: Colors.green,
              borderRadius: BorderRadius.circular(50),
            ),
            child: Padding(
              padding: const EdgeInsets.all(7.0),
              child: SvgPicture.asset(
                'assets/icons/doctor.svg'),
            ),
          ),
          const SizedBox(
            width: 10,
          ),
          const Text('Bci nikapi'),
        ],
      ),
    ),
    const SizedBox(
      height: 30,
    ),
    const ArticlesBlock(),
  ],
),
)
],
);
} else {
  return Container();
}
},
),
),
);
}
}

```

ДОДАТОК В
Слайди презентації

Національний університет "Запорізька політехніка"
Кафедра програмних засобів

Дослідження та програмна реалізація застосунку для запису до медичних фахівців.

Виконав
ст. гр. КНТ-142М

О. В. Молчанов

Керівник
к.т.н., доцент

Т. В. Федорончак

Рисунок В.1 – Слайд 1

Об'єкт дослідження - запис на прийом до медичних фахівців.

Предмет дослідження – мобільний застосунок для запису до медичних фахівців.

Мета роботи – дослідження та розробка мобільного застосунку для зручного та ефективного запису до медичних фахівців.

Рисунок В.2 – Слайд 2

Завдання роботи

1. **Аналіз:** Вивчити існуючі системи онлайн-запису до медичних фахівців.
2. **Модель взаємодії:** Розробити модель взаємодії користувача та визначити функціональні вимоги.
3. **Проектування:** Створити архітектуру та врахувати технічні аспекти системи.
4. **Реалізація:** Розробити програмне забезпечення для онлайн-системи запису до медичних фахівців.
5. **Тестування:** Перевірити коректність роботи системи та виправити можливі помилки.

Рисунок В.3 – Слайд 3

Порівняльна характеристика застосунку для запису до медичних фахівців

Критерій, за яким порівнюють	Helsi	MeetDoc	DOC.ua
Наявність в Україні	+	-	+
Мова інтерфейсу	українська, російська, англійська	англійська, французька	українська, російська, англійська
Наявність медичних фахівців	+	+	+
Процес запису	+	+	+
Ціна	базовий тариф безкоштовний, платні тарифи від 129 грн/місяць	базовий тариф безкоштовний, платні тарифи від 15 USD/місяць	базовий тариф безкоштовний, платні тарифи від 99 грн/місяць
Функції	+	+	+
Критика	+-	+-	+-

Рисунок В.4 – Слайд 4

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Dart	Java	Rust
Зручність використання для створення застосунку для запису до медичних фахівців	+	+-	+-
Продуктивність	+	-	-
Легкість вивчення	+	+	+
Розробка мобільних додатків	+	+	-
Екосистема та фреймворки	+	+	+
Спільнота та підтримка	+	+	+-
Переносимість коду	+	-	+

Рисунок В.5 – Слайд 5

Вибір фреймворку Flutter

Критерій	Перевага
Нативність	Flutter забезпечує нативну продуктивність на всіх платформах, що означає, що застосунки, створені за допомогою Flutter, будуть працювати так само швидко і плавно, як і застосунки, створені за допомогою native-фреймворків. Це важливо для застосунків, які вимагають швидкої і плавної роботи, таких як застосунки для запису до медичних фахівців.
Розробка для кількох платформ	Flutter дозволяє розробляти застосунки для кількох платформ за допомогою одного коду. Це може значно заощадити час і витрати на розробку. Для застосунків для запису до медичних фахівців, які повинні бути доступні на різних платформах, це може бути важливим фактором.
Компонентний дизайн	Flutter використовує компонентний дизайн, який робить застосунки простішими в розробці та підтримці. Компонентний дизайн також дозволяє створювати застосунки, які легко адаптуються до різних розмірів екранів і пристроїв.
Швидка розробка	Flutter використовує байт-код, який робить його швидким і ефективним у розробці. Це може значно скоротити час, необхідний для розробки застосунку. Для застосунків для запису до медичних фахівців, які повинні бути запуснені в найкоротші терміни, це може бути важливим фактором.
Розробка для майбутнього	Flutter є сучасним фреймворком, який підтримується компанією Google. Це означає, що Flutter буде продовжувати розвиватися і вдосконалюватися в майбутньому. Це важливо для застосунків для запису до медичних фахівців, які повинні бути здатними підтримувати найновіші функції та можливості.

Рисунок В.6 – Слайд 6



Рисунок В.7 – Слайд 7

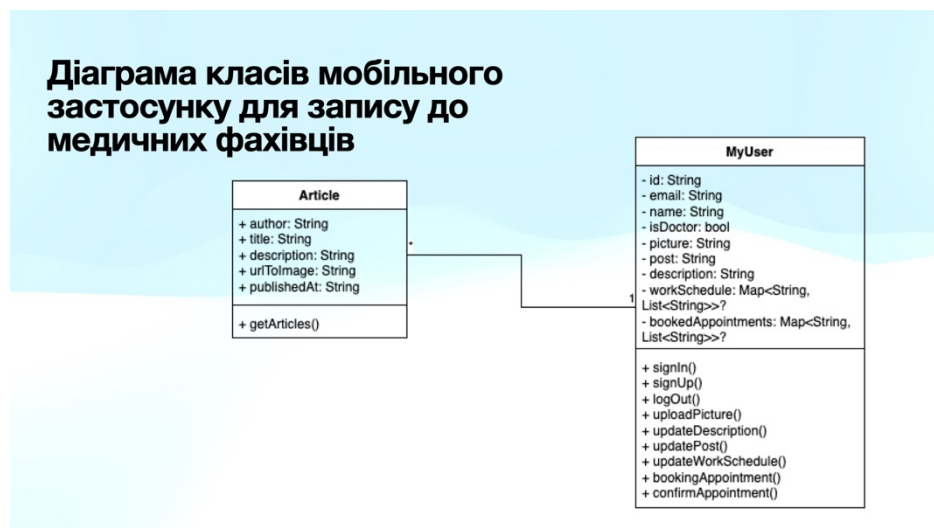


Рисунок В.8 – Слайд 8



Рисунок В.9 – Слайд 9

Загальна схема функціонування мобільного застосунку для запису до медичних фахівців у вигляді діаграми станів

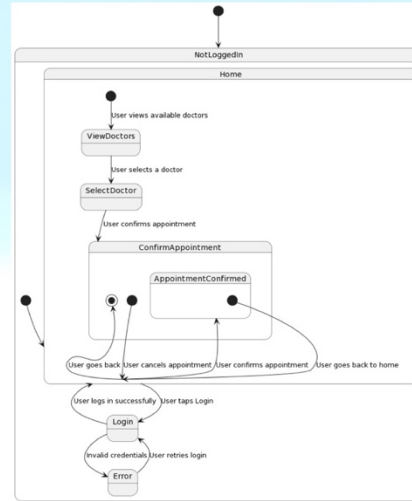


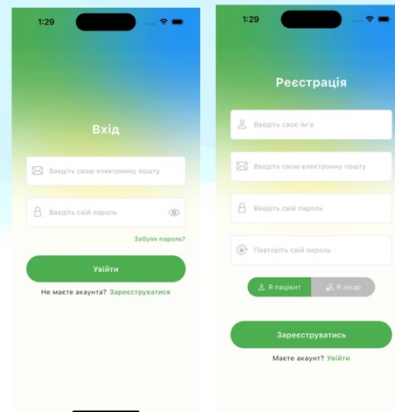
Рисунок В.10 – Слайд 10

Діаграма діяльності мобільного застосунку для запису до медичних фахівців



Рисунок В.11 – Слайд 11

Робота з мобільним застосунком



Форма реєстрації та авторизації користувача застосунку

Рисунок В.12 – Слайд 12

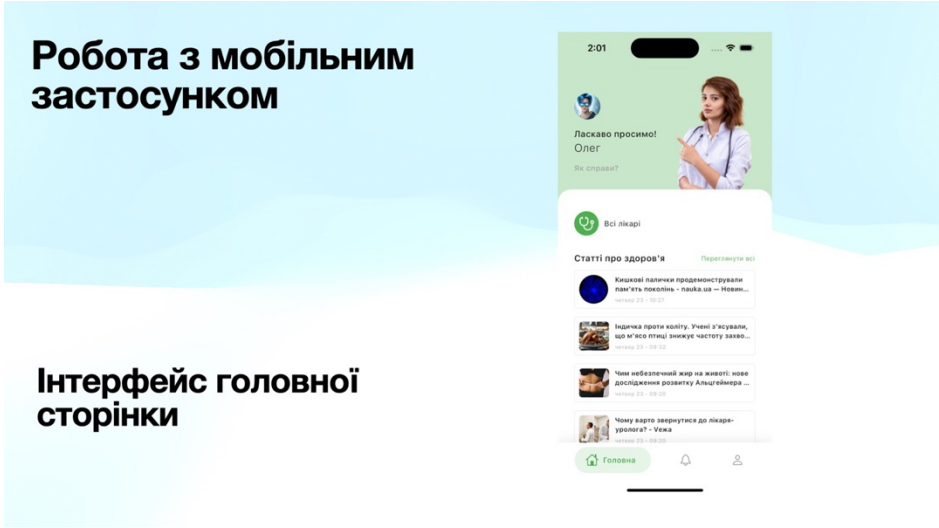


Рисунок В.13 – Слайд 13

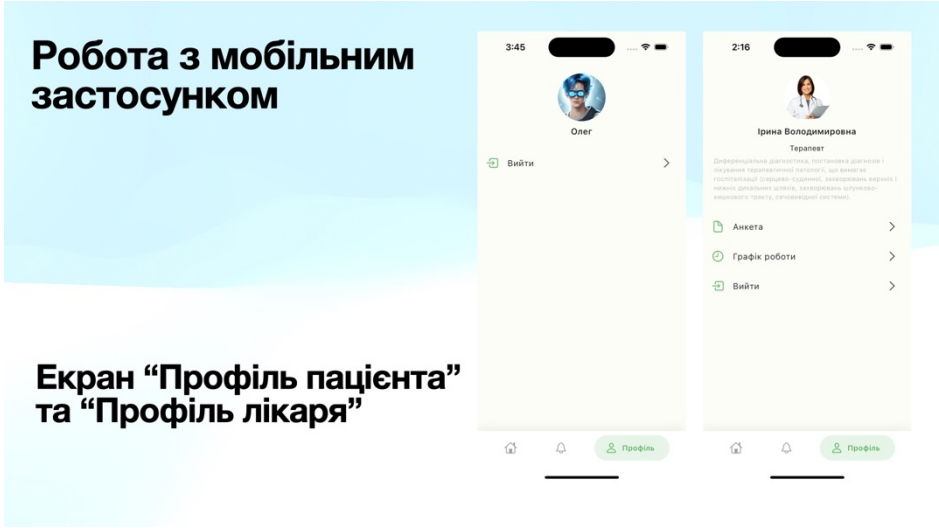


Рисунок В.14 – Слайд 14

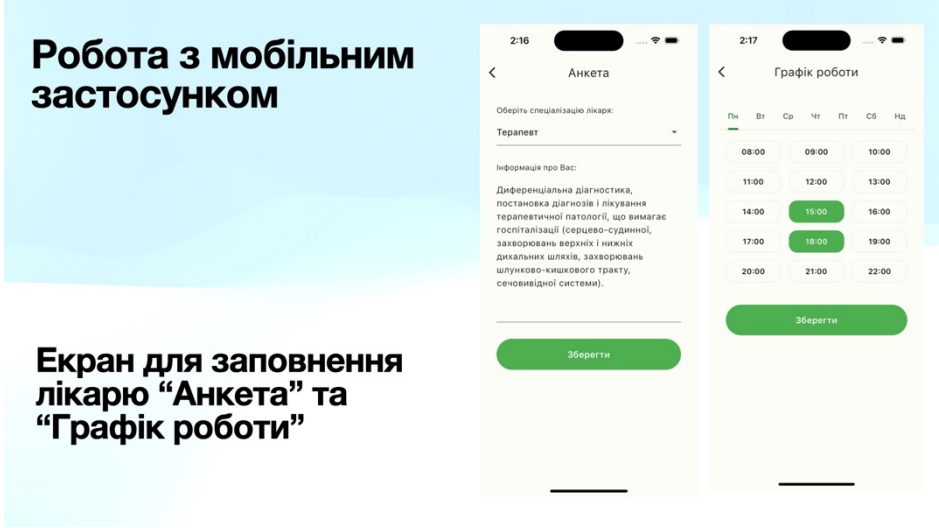


Рисунок В.15 – Слайд 15

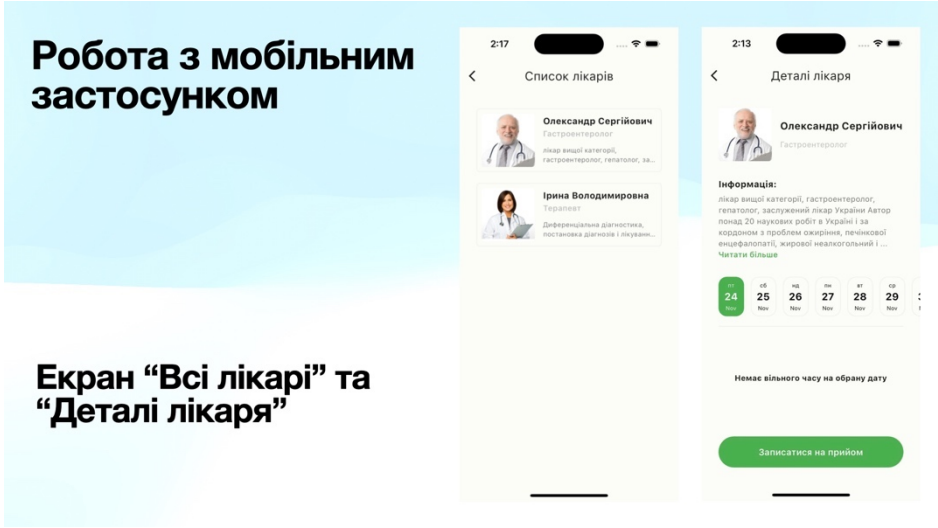


Рисунок В.16 – Слайд 16

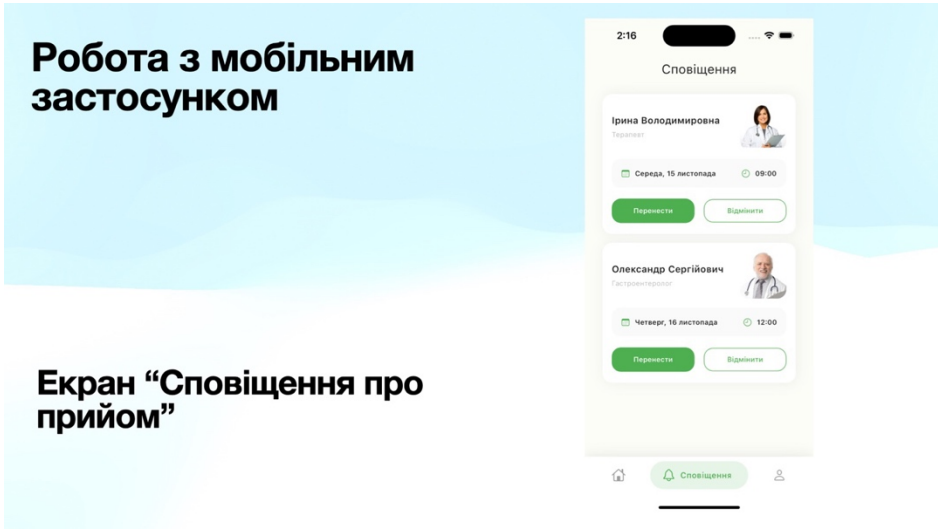


Рисунок В.17 – Слайд 17

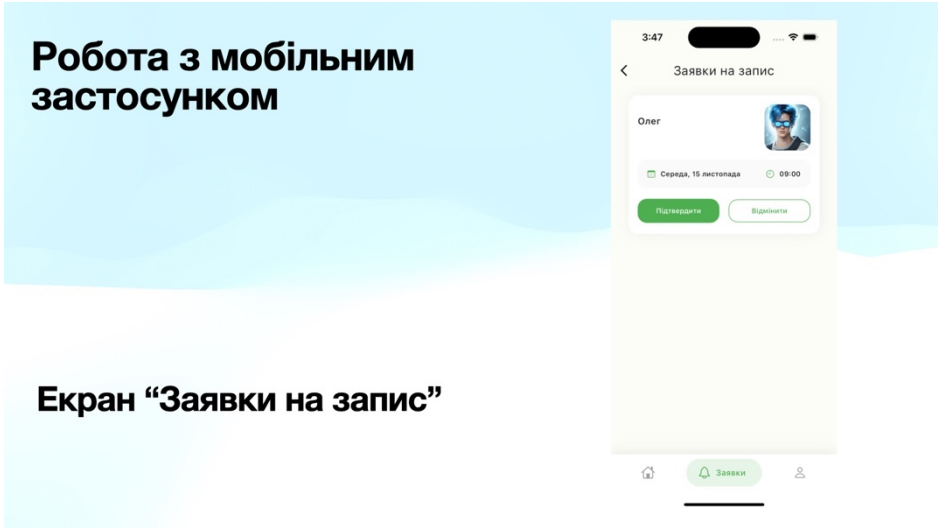


Рисунок В.18 – Слайд 18

Висновки

В ході виконання дипломної роботи з дослідження та програмної реалізації застосунку для запису до медичних фахівців було проведено комплексний аналіз та дослідження процесу розробки застосунку, визначено потребу у зручному мобільному додатку для запису до медичних фахівців.

Обрано мову програмування Dart та фреймворк Flutter для розробки кросплатформених додатків. Визначено функціональні вимоги та реалізовано структуру застосунку в нотації UML. Підкреслено важливість вибору Dart та Flutter для забезпечення швидкості та ефективності додатку. Проаналізовано основні класи та описано функціонал застосунку. Результати тестування свідчать про високу ефективність та відповідність вимогам.

Розроблений мобільний застосунок має практичне значення, сприяючи оптимізації обслуговування пацієнтів та удосконаленню системи медичного обслуговування. Робота свідчить про успішну розробку та впровадження системи для ефективного управління записами до медичних фахівців.

Рисунок В.19 – Слайд 19

Дякую за увагу!



Рисунок В.20 – Слайд 20