

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«МОДЕЛЮВАННЯ ДОВІДНИКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕДИЧНИХ ЗАКЛАДІВ ІЗ ОСОБИСТИМ КАБІНЕТОМ ПАЦІЄНТА ТА ЛІКАРЯ НА ОС ANDROID»

Студента 2м курсу, 2 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Суржика Максима
Олександровича

підпис студента

Науковий керівник
кандидат технічних наук,
доцент

Рзаєва Світлана
Леонідівна

підпис керівника

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

Токар Володимир
Володимирович

підпис керівника

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення.

XML (англ. Extensible Markup Language) – розширювана мова розмітки.

AS (англ. Android Studio) – середовище розробки мобільних застосунків.

Firestore – модуль управління базами даних. Безкоштовний аналог віддаленої бази даних.

БД – база даних.

КНТЕУ 121 02з-13.МР							
Зм.	Аркуш	№ докум	Підпис	Дата			
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		27.02.2021			
<i>Керівник</i>		<i>Рзаєва С.Л.</i>		27.02.2021			
<i>Гарант</i>		<i>Токар В.В.</i>		27.02.2021			
<i>Розроб.</i>		<i>Суржик М.О.</i>		27.02.2021			
<i>СПИСОК СКОРОЧЕНЬ ТА ТЕРМІНІВ</i>							
<i>Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень</i>					Стадія	Аркуш	Аркушів
					ПС	2	61
<i>Факультет інформаційних технологій</i>							
<i>2м курс, 2 група</i>							

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ ТА ТЕОРЕТИЧНА ЧАСТИНА БАЗИ ДАНИХ ЛІККАРЕНЬ.....	6
1.1. Основні поняття та визначення бази даних клінічної лікарні.....	6
1.2. Огляд існуючих клінічних систем.....	7
1.3. Огляд середовища розробки мобільних додатків.....	10
1.4. Висновки до розділу 1.....	12
РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ МОБІЛЬНОГО ДОДАТКУ.....	13
2.1. Створення бази даних мобільного додатку та її таблиць.....	18
2.2. Розгляд полів та таблиць, створення запитів.....	22
2.3. Опис коду бази даних мобільного додатку.....	26
2.4. Висновки до розділу 2.....	31
РОЗДІЛ 3 РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО ДОДАТКУ.....	32
3.1. Розробка інтерфейсу та дизайну мобільного додатку.....	32
3.2. Розробка навігації мобільного додатку.....	35
3.3. Створення налаштувань мобільного додатку.....	39
3.4. Висновки до розділу 3.....	43
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТКИ.....	47
Технічне завдання.....	47

					КНТЕУ 121 02з-13.МР			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		27.02.2021		3	3	61
<i>Керівник</i>		<i>Рзаєва С.Л.</i>		27.02.2021		<i>Факультет інформаційних технологій 2м курс, 2 група</i>		
<i>Гарант</i>		<i>Токар В.В.</i>		27.02.2021				
<i>Розроб.</i>		<i>Суржик М.О.</i>		27.02.2021	<i>ЗМІСТ</i>			

ВСТУП

Актуальність. Створення сучасних електронних обчислювальних машин дозволило автоматизувати обробку даних у багатьох сферах людської діяльності. Без сучасних систем обробки даних важко уявити сьогодні передові виробничі технології, управління економікою на всіх її рівнях, наукові дослідження, освіту, видавничу справу, функціонування засобів масової інформації, проведення великих спортивних змагань. Значно розширило сферу застосування систем обробки даних поява персональних комп'ютерів. На сьогоднішній день в лікарнях України запис до лікаря та перевірка існуючих лікарів можлива тільки в лікарні по місці або на веб-сторінках. На даний момент в Україні в 2020р з'явився перший додаток на мобільних пристроях, який надав змогу переглянути наявних лікарів місцевої лікарні а також запису до них на прийом. Наразі веб-версія додатку електронної черги існує в вигляді мобільного додатку і містить наступну інформацію:

- дані пацієнта;
- дані лікаря та область його роботи;
- список вільних та зайнятих лікарів;
- можливість онлайн запису на прийом до лікаря;
- графік роботи лікарні;
- графік роботи лікарів;

Актуальність створюваного додатку полягає в розширенні можливостей уже створеного аналогу додатку. Використання додатку має розширити можливості для звичаних пацієнтів, та скоротити час, який витрачається на простій в чергах, як на запис до лікаря так і безпосередньо до самого лікаря.

Мета дослідження: розробка довідниково-інформаційної системи медичних закладів із особистим кабінетом пацієнта та лікаря на ОС Android.

Об'єкт дослідження: Проектування довідниково-інформаційної системи.

					КНТЕУ 121 2м-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

Предмет дослідження: Методи та моделі проектування довідниково-інформаційної системи системи взаємодії лікарів та пацієнтів для міських лікарень

У відповідності з метою дослідження поставлено завдання:

- проаналізувати функції інформаційно-довідникової системи, її збір, обробка даних для подальшого зберігання та надання відомостей користувачам за відповідними запитами;
- аналіз існуючих середовищ розробки мобільних додатків;
- розробка бази даних;
- проектування архітектури мобільного додатку;
- розробка програмного забезпечення довідниково-інформаційної системи;
- протестувати створений додаток.

Вид і зміст наданої інформації формується під впливом накопичених в довідкових сховищах даних. Процес видачі запитуваних відомостей заснований на асоціативному пошуку в базах (сховищах) з подальшою переробкою їх в зручний для користувача формат. Інформаційно-довідкова система архівної галузі дозволяє накопичувати і зберігати дані протягом тривалих проміжків часу.

Методи дослідження: метод аналізу, метод синтезу даних.

Наукова новизна дослідження використання великої кількості застосунків (запис на прийом, нагадування приймати ліки, перевірка лікарів, рейтинг лікарів та ін.) в одному додатку який об'єднує інформацію про всю лікарню.

Практичним значенням дослідження є використання великої кількості застосунків (запис на прийом, нагадування приймати ліки, перевірка лікарів, рейтинг лікарів та ін.) в одному додатку який об'єднує інформацію про всю лікарню. В результаті буде створено застосунок для поліпшення відвідування міських клінічних лікарень.

									Аркуш
									5
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР

РОЗДІЛ 1

МОБІЛЬНІ ДОДАТКИ ЯК ОСНОВА ВЗАЄМОДІЇ ЛІКАРІВ ТА ПАЦІЄНТІВ ДЛЯ МІСЬКИХ ЛІКАРЕНЬ

1.1. Основні поняття та визначення бази даних клінічної лікарні

Для коректного визначення вимог, необхідних для створення мобільного додатку клінічної лікарні, визначено такі поняття:

База даних – це структуроване сховище даних. Для додавання інформації, її обробки або для доступу до інформації, яка зберігається в базах даних на комп'ютерах, використовуються системи управління базами даних (СУБД) такі як SQL. Сервер баз даних Firebase SQL достатньо швидкий, надійний та легкий в управлінні.

Система управління базами даних (СУБД) – це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

До теоретичних положень теми можна віднести історію використання та розвитку мобільних додатків а також розвиток електронних систем пов'язаних з лікарнями: В результаті технічного розвитку в Україні декілька років тому впроваджується створення централізованої бази даних всіх лікарів та лікарень для полегшення перегляду доступних лікарів та запису до них на прийом, а також в зв'язку з ситуацією в світі запроваджено онлайн запис на вакцинування від COVID-19. У зв'язку з можливістю електронної передачі даних через інтернет створюються спеціалізовані сайти та бази даних клінік і лікарень, які в свою чергу беруть дані з ЦБД. Спочатку це торкнулось тільки лікарень в місті Київ, але в ході розвитку

Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 02з-13.МР			
Зав. кафедри	Криворучко О.В.			16.04.2021	Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень	Стадія	Аркуш	Аркушів
Керівник	Рзаєва С.Л.			16.04.2021		P1	6	61
Гарант	Токар В.В.			16.04.2021		Факультет інформаційних технологій 2м курс, 2 група		
Розроб.	Суржик М.О.			16.04.2021				
					МОБІЛЬНІ ДОДАТКИ ЯК ОСНОВА ВЗАЄМОДІЇ ЛІКАРІВ ТА ПАЦІЄНТІВ ДЛЯ МІСЬКИХ ЛІКАРЕНЬ			

інновації з кожним роком зростають і тепер кількість лікарень, що є підключеними до ЦБД збільшується з великою швидкістю.

До узагальнених етапів проектування БД відноситься: дослідження предметної області, аналіз даних (сутностей та їх властивостей), визначення відносин між сутностями і визначення первинних і вторинних (зовнішніх) ключів.

База даних - це інструмент для збору, зберігання і структурування інформації. SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Розшифровується як Structured Query Language - мова структурованих запитів, яка була розроблена для роботи з БД, а саме, щоб отримувати/добавляти/змінювати дані, мати можливість опрацьовувати великі масиви інформації та швидко отримувати структуровану та згруповану інформацію.

1.2. Огляд існуючих клінічних систем

На даний час в Україні уже існує декілька якісно створених баз даних для перегляду доступних лікарів та запису до них на прийом. Ці бази даних існують в різних формах. Так, наприклад для запису на прийом не відвідуючи лікарню існують спеціалізовані сайти (<https://medlife.pro/ua> (рис. 1.1), <https://helsi.me> (рис. 1.2) і т.д.). Також існують програми для запису на прийом до лікаря, які знаходяться безпосередньо в лікарні, та до яких зазвичай мають доступ лише лікарі та медперсонал.

Існуючі бази даних почали активно використовуватися людьми молодого та середнього віку, а також людьми що мають доступ в інтернет. Для людей похилого віку, які звикли ходити до лікарень, а також для людей в яких немає доступу до мережі інтернет створено програми які використовуються в лікарнях. За допомогою цих програм доступ до запису можливий по номеру телефона або картці

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			7

пацієнта, яка може бути створена за декілька хвилин. Саме тому ці системи останні декілька років набирають популярність.

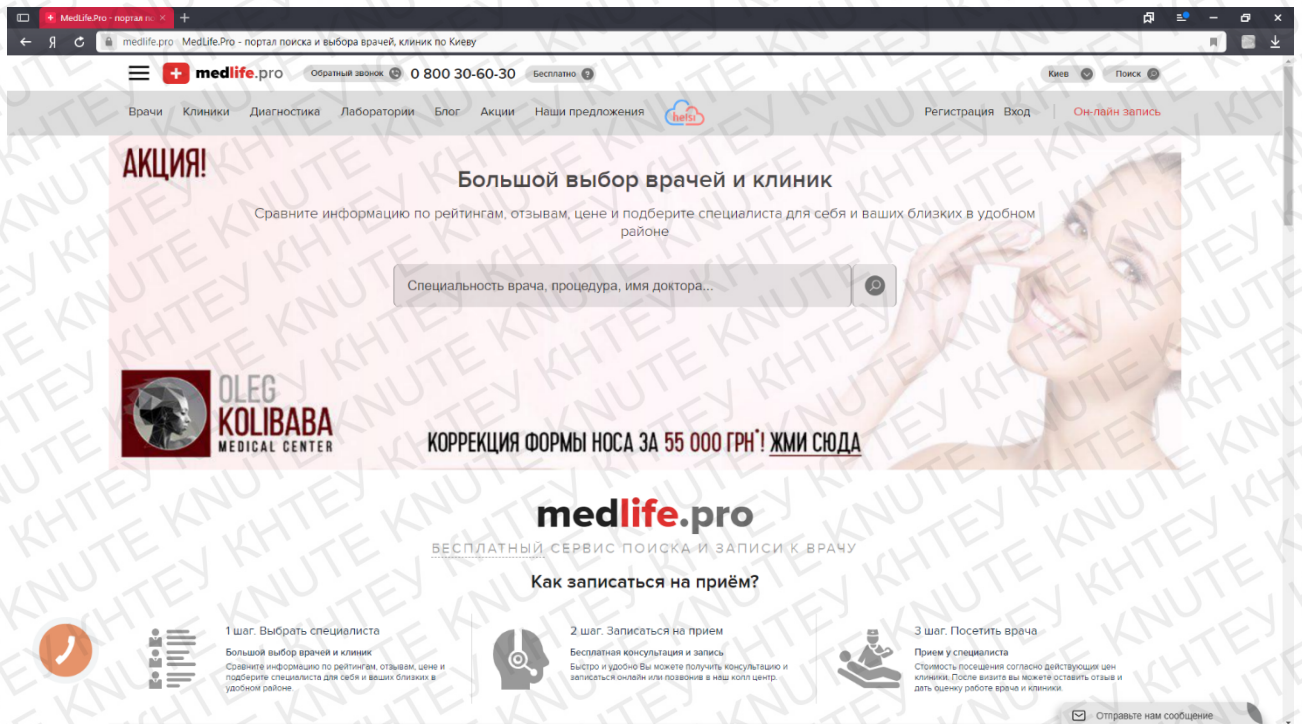


Рис 1.1. Сайт «Medlife.pro»

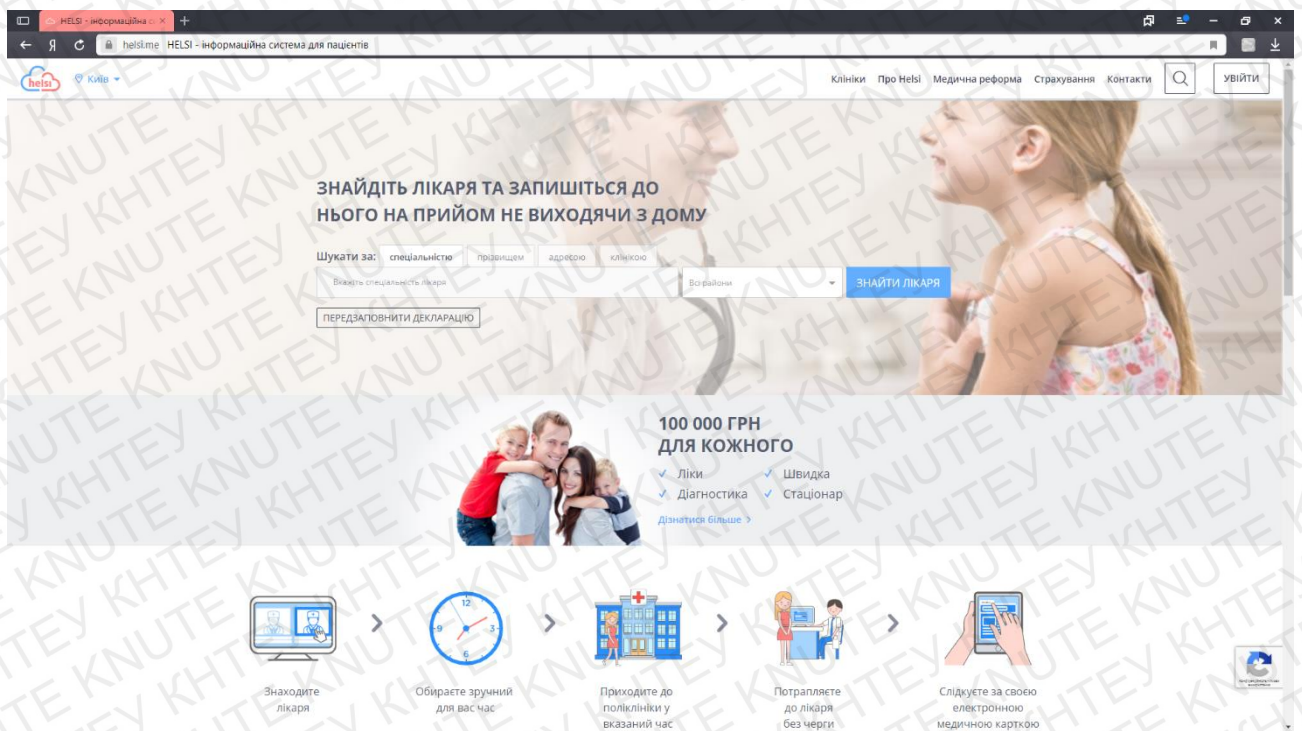


Рис 1.2. Сайт «Helsi»

									Аркуш
									8
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР

Також в Україні була створена ЦБД eHealth. eHealth — електронна система охорони здоров'я, що забезпечує обмін медичною інформацією та реалізацію програми медичних гарантій населення. Міністерство охорони здоров'я регулює впровадження eHealth на нормативно-правовому рівні. Національна служба здоров'я України укладає договори з постачальниками медичних послуг; аналізує та використовує дані з метою прогнозування потреб населення у медичних послугах, розробки програми медичних гарантій, встановлення тарифів, здійснення моніторингу договорів; здійснює оплати за медичні послуги на умовах договору.

Приблизно 8 місяців тому було випущено мобільний додаток Helsi.me, який є прямим конкурентом даного проекту. З доступних функцій там є:

- Пошук лікаря, інформація про лікаря;
- Можливість виставити оцінки лікарю, переглядати їх;
- Зробити запис на прийом до лікаря на обрані дату та час;
- Обирати між державними та приватними лікарями;
- Робити запис як для дітей так і для дорослих;
- Подивитись необхідну інформацію на прийом;
- Дізнатись «прайс» на всі послуги;
- Переглянути результати аналізів і діагностики;
- Мати доступ до назначень лікаря;
- Переглянути необхідні ліки за наявністю в сертифікованих лікарнях поруч;

З отриманих даних можливо зробити висновок, що система та проект «Helsi» має достатній розвиток за останні декілька років, вона підлаштовується під ситуацію в світі, саме тому останні оновлення виходять на користь вакцинавання від COVID-19. Цілями даного проекту є впровадження медичного додатку з повною історією хвороб та перенесення всіх даних медичної картки в БД мобільного додатку.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 02з-13.МР

Аркуш

9

1.3. Огляд середовища розробки мобільних додатків

Android Studio – безкоштовне середовище розробки на основі IntelliJ IDEA, надає інтегровані інструменти для розробки й налагодження додатків для платформи Android. *Android Studio* включає в себе:

- Android SDK;
- інструменти для розробки, тестування і налаштування;
- останню версію платформи Android для компіляції;
- останню версію образу Android для запуску програм.

Основні властивості Android Studio:

- редактор WYSIWYG. Рендеринг додатків в реальному часі;
- консоль розробника: підказки для оптимізації, помічник для перекладу;
- підтримка білдів на основі Gradle;
- інструменти для вимірювання продуктивності, зручності користування, сумісності і т. д.;
- proGuard і можливості підпису додатків;
- майстер шаблонів для створення стандартних дизайнів і компонентів Android;
- багатий редактор шарів з можливістю перетягування компонентів інтерфейсу, перегляду шарів з різноманітними налаштуваннями екрану.

Переваги:

- відображення всіх змін, внесених у режимі реального часу;
- потужний емулятор Android-пристроїв;
- інтуїтивно зрозумілий інтерфейс;
- легка локалізація програми;
- підтримка можливості маркування коду.

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			10

Android Studio – офіційне середовище розробки додатків для операційної системи, покликане замінити собою Eclipse. Остання, між іншим, на даний момент є найпопулярнішим інструментом у колі Android 69 розробників. Процес розробки в середовищі Android Studio став ще більш гнучким, ніж у головного конкурента. Це досягається за рахунок відображення всіх робочих файлів прямо в структурі проекту. Вкрай корисною є також можливість бачити всі візуальні зміни проекту в режимі реального часу. Ще дана SDK дає можливість протестувати майбутню роботу на різних пристроях. У вбудованому емуляторі Android девайсів можна встановити різноманітні технічні конфігурації і дозволи екрану для тестування. Крім того, при використанні емуляції є можливість отримати інформацію про приблизний рівень продуктивності для того або іншого пристрою [2]. Android Studio включає в себе інноваційні інструменти для упаковки і маркування коду, які допоможуть не загубитися у великому проекті. Також в програмі реалізована функція Drag-n-Drop, яка дозволяє перетягувати компоненти проекту прямо у вікні середовища розробки. Для реалізації можливості відправки майбутнім додатком Push-повідомлень, у пакеті Android Studio є спеціальний інструментарій під назвою Google Cloud Messaging. Ще дана SDK сильно спрощує процес локалізації додатків.

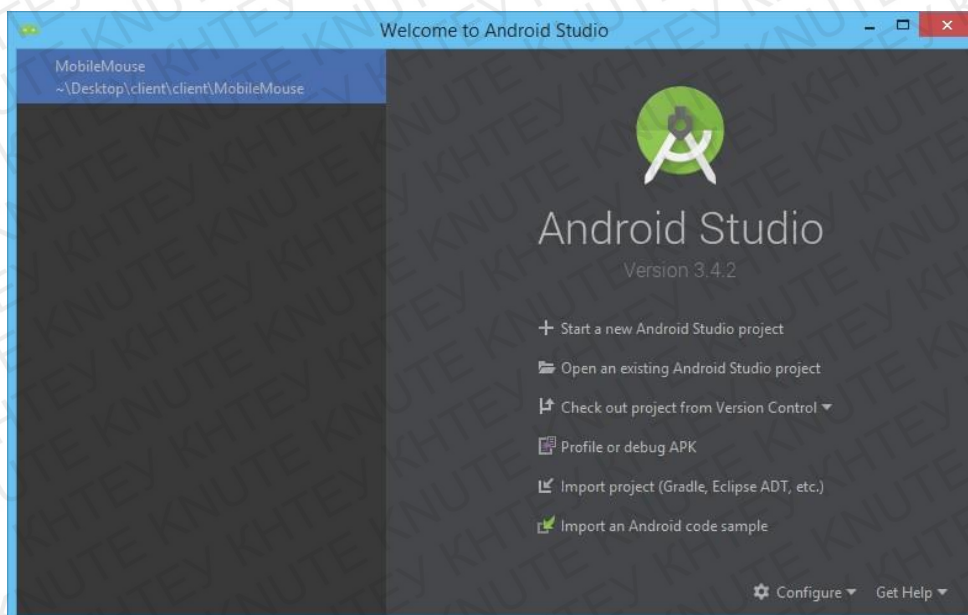


Рис 1.3. Стартове вікно Android Studio

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			11

1.4. Висновки до розділу 1

В Україні створено ЦБД eHealth, до якої прив'язують інші локальні бази даних. ЦБД впроваджується Національною службою здоров'я України.

В даний час набирають популярність електронні бази даних лікарень, саме тому вирішено було обрати напрямком створення мобільної версії додатку клінічної лікарні.

Електронні бази даних для віддаленого запису на прийом набрали великої популярності, окрім того на їх створення виділяється багато коштів, тому там іде велика конкуренція. Вирішено було створювати мобільний додаток, функціями якого є перегляд своєї медичної історії, вибір лікаря, запис до нього на прийом, отримання додаткової консультації у лікаря через листування.

Android Studio – безкоштовне середовище розробки на основі IntelliJ IDEA, надає інтегровані інструменти для розробки й налагодження додатків для платформи Android. Великі можливості, що надає дане середовище роблять його топ 1 застосунком для створення додатків для мобільних пристроїв на ОС Android.

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			12

РОЗДІЛ 2

МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ МОБІЛЬНОГО ДОДАТКУ

2.1. Опис архітектури мобільного додатку

Довідниково-інформаційна система призначена для організації взаємодії пацієнтів та працівників міських лікарень за допомогою мобільних пристроїв.

Основною метою розробки – комп'ютеризація процесів організації відвідування лікарів пацієнтами, а також планування, резервування місць та інформування відвідувачів лікарні на наявність вільних лікарів.

Дана система повинна володіти характеристиками і виконувати наступні задачі:

- простота у використанні, система повинна мати зрозумілий інтерфейс;
- система повинна працювати з максимальною кількістю версій ОС Android;
- система повинна надавати розгорнуту інформацію щодо лікарів, наявних в лікарні;
- система повинна використовувати мінімальну кількість технічних ресурсів, та не навантажувати мобільний пристрій;
- система повинна працювати швидко, користувач повинен отримувати актуальну інформацію миттєво;
- процес входу до системи повинен передбачати розділення користувачів на лікарів та пацієнтів;

					КНТЕУ 121 02з-13.МР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.		24.05.2021	Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ МОБІЛЬНОГО ДОДАТКУ	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		24.05.2021		P2	13	61
Гарант		Токар В.В.		24.05.2021		Факультет інформаційних технологій 2м курс, 2 група		
Розроб.		Суржик М.О.		24.05.2021				

- система має передбачати захист даних користувачів від несанкціонованого доступу зловмисників;

- збір та зберігання даних будуть проводитися на захищених серверах;

Процес проектування розробки

У процесі проектування розробки системи головним завданням є пред'явлення функціональних та системних вимог щодо програмного забезпечення.

Функціональні вимоги до розробки додатку.

1. Надання доступу авторизованим користувачам.

2. Розробка ролей користувачів: «Лікар», «Пацієнт».

3. Обов'язкова наявність, присутність і можливість пошуку лікарів по загальній базі даних лікарні а також перегляд інформації по необхідному лікарю.

4. Наявна можливість редагування, видалення та додавання події, що пов'язана з записом на прийом до лікаря для лікарів, і можливість видалення і додавання події для ролей пацієнт.

Системні вимоги до розробки додатку:

1. Інтуїтивно зрозумілий інтерфейс.

2. Операційна система Android 4.6 та вище.

3. Використання з'єднання для обміну даними з сервером - протокол TCP/IP.

4. Серверна частина, що використовує SQL Lite, повинна зберігати базу даних системи.

5. Серверна частина буде створена на базі фреймворку Google Firebase, та повинна підтримувати оновлення даних в реальному часі.

Основними особливостями середовища є:

– Інтуїтивно зрозумілий інтерфейс для користувача;

– SQL Lite – компактна версія мови SQL до складу якого входять: об'єкти послідовності, функції перетворення, логічні функції, дата та час;

					КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

- масштабованість та продуктивність – система може приймати великі навантаження та обслуговувати відразу велику кількість користувачів;
- безпека, що забезпечує ретельну перевірку бази даних та основні ролі серверів, що визначені користувачем;
- повнотекстовий пошук – компонент, що реалізований на повнотекстовій функції пошуку у SQL Server, дозволяє користувачам і додаткам виконувати повнотекстовий запити до символічних даних.

6. Клієнтська частина повинна бути розроблена у середовищі Android Studio 7.2 на мові програмування Java, що є сучасною об'єктно-орієнтованою мовою програмування. Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

7. Використання Google Firebase для Android Studio, за допомогою якого буде реалізовано зв'язок з базою даних. Інструмент є в відкритому доступі і надає розробнику швидший та продуктивніший спосіб реалізації «живої» бази даних, яка відповідає потребам користувача. Firebase надає в режимі реального часу базу даних та бекенд як службу. Ця служба надає розробникам застосунків API, який дозволяє синхронізувати дані застосунків між клієнтами та зберігати їх у хмарі Firebase. Компанія також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних

					КНТЕУ 121 02з-13.МР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

також доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, Ember.js та Backbone.js REST API використовує протокол подій із сервером, який є інтерфейсом для створення HTTP-з'єднань для отримання push-повідомлень від сервера. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері.

Для полегшення проектування використовуємо уніфіковану мову моделювання (Unified Modeling Language, UML), яка є графічною мовою для специфікації, візуалізації, проектування та документування систем.

На початку розробки необхідно реалізувати архітектуру системи управління особистим кабінетом. Отже, розглянемо розроблену діаграму компонентів, що описує організаційну сторону та підключення фізичних компонентів у проєктованій системі. Зазначена діаграма показує системні вимоги до розробки (рис. 2.1).

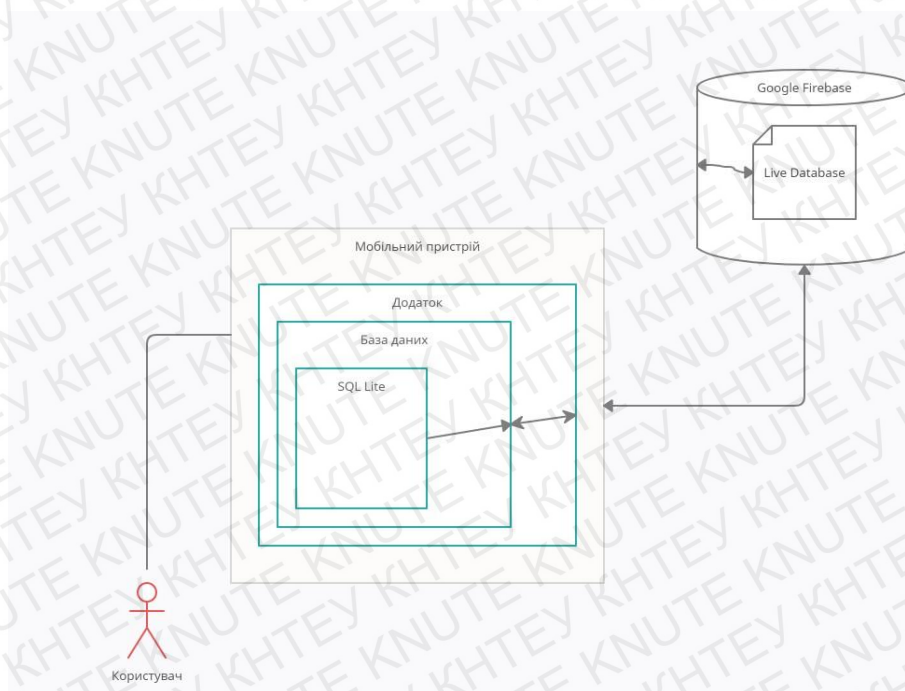


Рис. 2.1. Діаграма компонентів системи користувача

									Аркуш
									16
Зм.	Аркуш	№ докум	Підпис	Дата					

Наступним етапом наведено спроектовану діаграму логічної моделі, що є цінною для візуалізації функціональних вимог системи. Дана діаграма відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктноорієнтованого програмування. На цій діаграмі показано класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Одразу кілька користувачів можуть працювати з базою даних у додатку. Роль користувача визначається після процесу авторизації, у разі якщо це «Лікар», він має право переглядати інформацію щодо наявних сеансів, редагувати їх, редагувати свій профіль, змінювати свій графік. Можливості адміністратора з управління системою: додавання, редагування, видалення та роботи із користувачами наявні лише у Адміністратора додатку. Якщо авторизований користувач «Пацієнт», він має право перегляду доступних лікарень, подання заявки на сеанс до лікаря, редагування власного профілю.

Після операції запуску програми, додаток проводить процес підключення до бази даних у локальній мережі. У разі відсутності зв'язку з сервером видається повідомлення щодо неможливості підключення до бази даних і відбувається автоматичний вихід з програми. У випадку вдалого підключення, програма виводить першу форму – форма авторизації.

При виході користувача з системи програма завершить свою роботу, в протилежному випадку користувач може продовжити працювати з нею. В кожному такому випадку створюється і відображається нова форма. Після дійсного входу користувача у програму відбувається визначення його ролі, в залежності від цього

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			17

буде відображена відповідна форма. Користувач продовжує виконувати свою роботу з довідниково-інформаційною системою міської лікарні до того часу, поки не завершить програму.

2.2. Створення бази даних мобільного додатку та її таблиць

Призначення такої системи є зрозумілим, це можливість перегляду своєї медичної історії, вибір лікаря, запис до нього на прийом, отримання додаткової консультації у лікаря через листування.

Такий тип мобільного додатку використовується не тільки в міських клінічних лікарнях, їх можливо порівняти навіть з функціями мобільних додатків для запису до перукаря чи замовлення їжі.

До вхідної інформації можна віднести наявність основних засобів для розробки, глибоке поглиблення в систему програми і найголовнішим фактором: мати знання ООП і аналізувати всі результати.

В реляційних базах даних і плоских базах даних, таблиця це набір елементів даних (значень), які організовані з використанням моделі вертикальних стовпчиків (з різними іменами) і горизонтальних рядків. Таблиця має визначену кількість стовпчиків, в той час як кількість рядків може різнитися в різні моменти.

СУБД розроблена на Google Firebase та складається з 7 таблиць:

1. Пацієнт - Patient (ID пацієнта, прізвище, ім'я, по батькові, адреса, телефон, привязана медична картка, лікарі пацієнта, прийоми, активні чати, улюблені лікарі, направлення, бан);
2. Лікар - Doctor (ID лікаря, прізвище, ім'я, по батькові, ID спеціалізація, ID категорії, номер кабінету, прийоми, активні чати, пацієнти лікаря, оцінка лікаря, бан, ID цін на послуги);
3. Спеціалізація - Specialisation (ID спеціалізації, спеціалізація);

									Аркуш
									18
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР

4. Категорія - Category (ID категорії, категорія);
5. Прийом пацієнту - Reserption (ID лікаря, ID пацієнта, ID медичної картки, дата прийому, номер прийому, чат прийому, відміна прийому);
6. Діагноз - Diagnosis (ID діагнозу, діагноз);
7. Ліки - Drugs (ID ліків, ліки, номер);
8. Картка пацієнта – Medical Card (ID карти, ID пацієнта, ID лікаря пацієнта, ID діагнозу, ID ліків, ID всіх прийомів до лікаря, історія хвороб);
9. Чат – Chat (ID чату, ID пацієнта, ID лікаря, повідомлення);
10. Послуги – Price List (ID прайс листу, ID лікаря, послуга, ціна);

Концептуальна(рис. 2.2), логічна(рис. 2.3) та фізична(рис. 2.4) моделі бази даних клінічної лікарні:

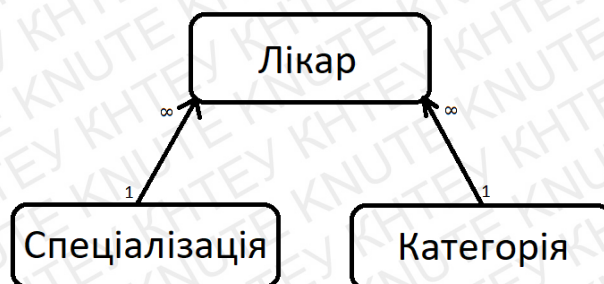


Рис. 2.2. Концептуальна модель бази даних мобільного додатку

Концептуальна модель включає в себе десять таблиць які показують основні принципи роботи даного мобільного додатку. Основними можливо виділити перші дві таблиці, адже решта таблиць це «допомога» в взаємодії перших двох таблиць. Запис на прийом створює нову заявку прийому, паралельно з цим створюється новий чат з лікарем, якщо його не було до цього. При створенні нового користувача також створюється і його медична картка, туди вноситься і діагноз і ліки за час відвідування медичного закладу, користувач має можливість вільно переглядати

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			19

дану інформацію. У відвідувачів є можливість переглядати послуги та ціну на кожного лікаря окремо та виставляти лікарям оцінку всередині додатку після прийому лікаря. Дана база даних має 4 віртуальні таблиці, які створюються лише на момент відкриття необхідного вікна в мобільному додатку.

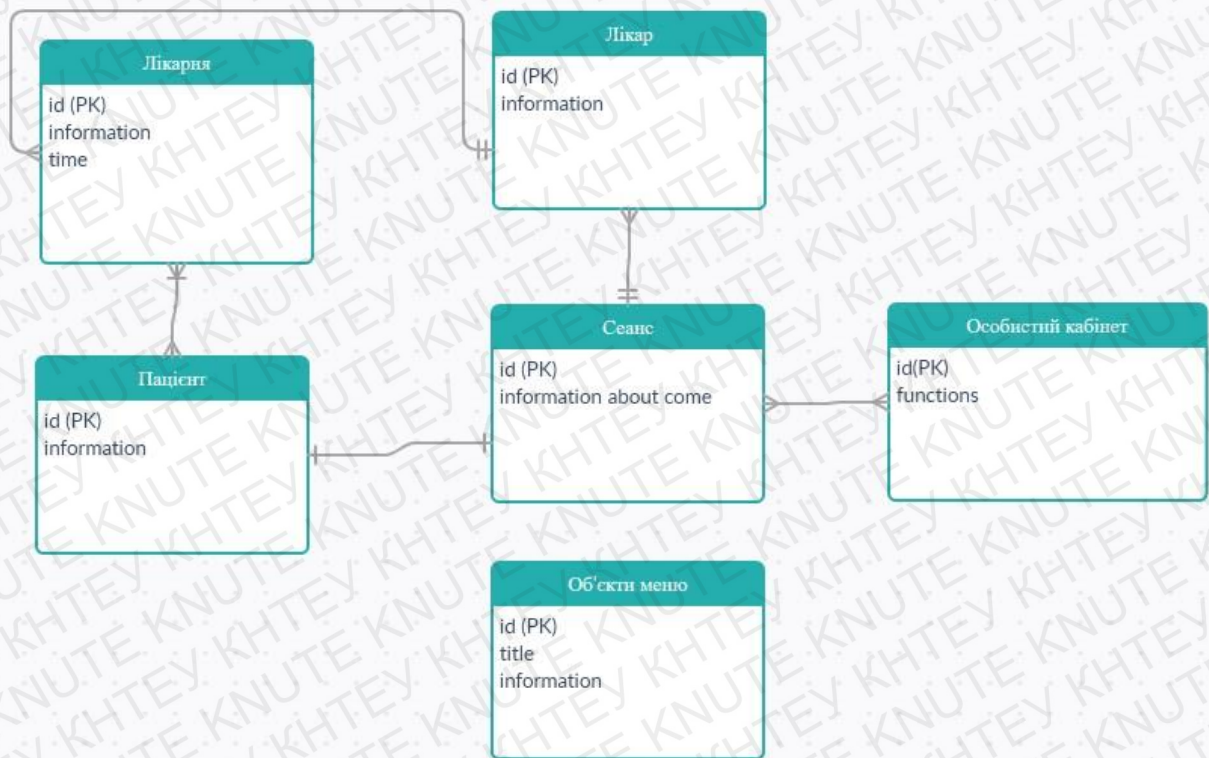


Рис. 2.3. Логічна модель бази даних мобільного додатку

Логічна модель бази даних має більш розширену структуру, тут можливо побачити які рядки є присутніми в кожній таблиці, а також їх Primary та Foreign keys. В базі даних присутні звязки один до багатьох та один до одного. Це все обумовлено тим, що під час створення програми вдалося спростити базу даних, а саме вхідні дані з таблиці «text». Додаток отримує відразу весь урок без зображень, потім ділить текст на частини по спеціальному знаку, та створює в пам'яті свій

									Архив
									20
Зм.	Архив	№ докум	Підпис	Дата					

власний віртуальний список, де замість спеціального знаку він вставляє зображення отримане з бази даних.

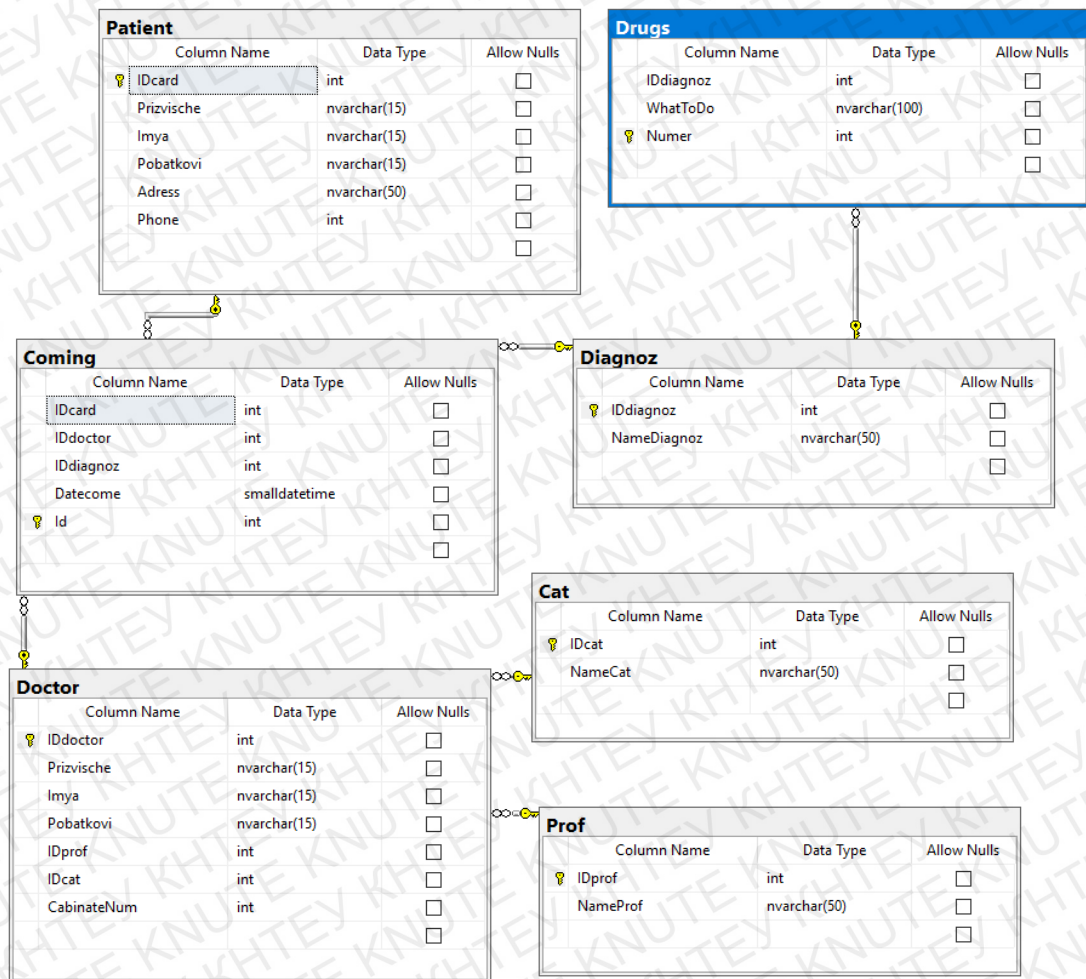


Рис. 2.4. Фізична модель бази даних мобільного додатку

Фізична модель бази даних вказує на тип даних що знаходяться в таблиці. Перевірку на проходження тесту чи перетворення активної кнопки в неактивну було вирішено замінити перевіркою на значення «1» та «0» в базі даних. При розблокуванні елемента дані в базі даних змінюються та записуються.

2.3. Розгляд полів та таблиць, створення запитів

Поле — це мінімальна (але найважливіша) порція інформації в записі, над якою визначені операції введення, виведення, перетворення тощо. Воно має ім'я, значення, характеризується типом і низкою додаткових властивостей.

Назви полям дає користувач, назви типів є стандартні, а значення полів впливають зі змісту конкретної задачі.

Очевидною унікальною властивістю кожного поля є його Ім'я. Крім імені в полі є ще властивість Підпис. Підпис – це та інформація, що відображається в заголовку стовпця. Її не треба плутати з іменем поля, хоча якщо підпис не заданий, то в заголовку відображається ім'я поля. Різним полям, наприклад, можна задати однакові підписи. Це не перешкодить роботі комп'ютера, оскільки поля при цьому як і раніше зберігають різні імена.

Перед створенням таблиці треба визначитись для чого призначаються різні типи полів. Основна властивість текстового поля – розмір.

Числове поле служить для введення числових даних. Воно теж має розмір, але числові поля бувають різними, наприклад для введення цілих чисел і для введення дійсних чисел. В останньому випадку крім розміру поля задається також розмір десяткової частини числа.

Для введення логічних даних, що мають тільки два значення (Так чи Ні; 0 чи 1; Істина чи Неправда і т.п.), служить спеціальний тип – Логічне поле. Неважко догадатися, що довжина такого поля завжди дорівнює 1 байту, оскільки цього більш ніж досить, щоб виразити логічне значення.

Дуже цікаве поле Лічильник. На перший погляд це звичайне числове поле, але воно має властивість автоматичного нарощування. Якщо в базі є таке поле, то при введенні нового запису в нього автоматично вводиться число, на одиницю

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			22

більше, ніж значення того ж поля в попередньому записі. Це поле зручне для нумерації записів.

Проаналізувавши поставлену задачу, були створені наступні таблиці:

«Doctor» – це таблиця, в якій містяться основна інформація про лікарів лікарні, їх ім'я, прізвище, по батькові, їх порядковий номер, спеціалізація, категорія, чат з активними пацієнтами, прийоми лікарів, їх прайс та оцінка. Ця таблиця пов'язана з таблицями «Category», «Specialisation», «Reception», «Price List», «Patient», «Chat».

«Patient» – це таблиця, в якій міститься інформація про пацієнтів, їх ім'я, прізвище, по батькові, їх порядковий номер, номер телефону, адреса, картка пацієнта, активні чати, прийоми, лікарі, улюблені лікарі. Ця таблиця пов'язана з усіма таблицями, окрім таблиць спеціалізації та категорії лікаря.

«Reception» – це таблиця зі списком пацієнтів які прийдуть на прийом до лікарів в указаний час для консультації. Лікарі можуть дати пацієнту направлення та описати діагноз, дані заносяться до медичної картки пацієнта. Також лікарі можуть виписати пацієнту ліки, дані також вносяться до медичної картки пацієнта. В даній таблиці зберігається інформація порядкового номеру пацієнту, лікаря, медична карта пацієнта, можливість відмінити прийом, час даного прийому. Якщо у пацієнта не було створено чату з лікарем до цього часу, він створюється автоматично. Ця таблиця пов'язана з таблицями «Patient», «Doctor», «Medical card», «Chat» та «Diagnosis».

«Diagnosis» – це таблиця зі списком діагнозів. Там є інформація про порядковий номер діагнозу та його назву. Ця таблиця пов'язана з таблицями «Medical Card».

«Drugs» – це таблиця зі списком певних ліків. В ній наявна інформація про номер ліків, їх порядковий номер та найменування. Ця таблиця пов'язана з таблицею «Medical Card».

									Аркуш
									23
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР

«Specialisation» – це таблиця із списком спеціалізацій. В ній наявна інформація про порядковий номер та найменування спеціалізації. Ця таблиця пов’язана з таблицею «Doctor».

«Caregory» – це таблиця із списком категорій. В ній наявна інформація про порядковий номер та найменування категорії. Ця таблиця пов’язана з таблицею «Doctor».

«Medical Card» – це таблиця, в якій містяться вся історія хвороб пацієнта, а також будь які дані щодо його аналізів, ліків та діагнозів. Ця таблиця пов’язана з таблицями «Drugs», «Patient», «Doctor», «Diagnosis».

«Chat» – це таблиця в якій зберігаються дінні листування між пацієнтом та лікарем з можливістю відправляти мультимедійні файли, такі як фото. В ній наявна інформація про час листування та саме повідомлення. Ця таблиця пов’язана з таблицями «Doctor», «Patient».

«Price list» – це таблиця в якій зберігаються ціни на послуги того чи іншого лікаря, а також вартість аналізів чи інших послуг лікарні. В ній є найменування послуги та ціна за неї. Ця таблиця пов’язана з таблицею «Doctor».

Запити дають широкі можливості для вибору, сортування і обчислення з використанням даних однієї таблиці. Можна налаштувати конкретний пошук будь чого в нашій базі даних, наприклад, теми та звязані з ними підтеми.

Запит на вибірку можна використовувати не тільки для відбору даних, але і для їх поновлення. Запит на вибірку має ряд властивостей, які можна використовувати для зміни роботи запиту.

В режимі таблиці доступні самі різні операції з даними - огляд, сортування, фільтрація, поновлення і друк. Але достатньо часто приходиться проводити обчислення і огляд даних з декількох таблиць. Відобразити потрібні дані можна за допомогою запитів.

									Аркуш
									24
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР

Після виконання запиту на вибірку (який відбирає інформацію з таблиць і інших запитів бази даних, в той час як при виконанні запиту на зміну дані вставляються, поновлюються або видаляються) Google Firebase створює набір записів, які містять відібрані дані. В більшості випадків з набором записів можна працювати так само, як з таблицею: можна проглянути і відібрати інформацію, роздрукувати і поновити дані. Але на відміну від реальної таблиці, цей набір записів фізично не існує в базі даних. Access створює набір записів з даних таблиць тільки під час виконання запиту. Якщо змінити дані в наборі записів, Google Firebase внесе відповідні зміни в таблицю, на базі яких побудований запит.

Приклад написання коду запитів для незалежної таблиці «dict» можливо переглянути на рисунку 2.5

```
package com.example.diplomepy.databse.dao;

import androidx.lifecycle.LiveData;
import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.Query;
import androidx.room.Update;

import com.example.diplomepy.databse.entities.DictModel;

import java.util.List;

@Dao
public interface DictDao {

    @Update
    void update(DictModel note);

    @Insert
    void insert(DictModel note);

    @Delete
    void delete(DictModel note);

    @Query("SELECT * FROM dict ORDER BY id_dict ")
    LiveData<List<DictModel>> getAllDict();
}
```

Рис. 2.5. Створення інтерфейсу запитів для таблиці «dict»

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

2.4 Опис коду бази даних мобільного додатку

Щоб керувати та синхронно користуватися програмою і базою даних Google Firebase ми повинні підключити цю базу даних до нашого мобільного додатку. Для цього необхідно створити окремий клас, який ми назвемо MainDatabase(рис. 2.7).

Перед тим як створити базу даних потрібно закинути вже створену базу даних в ассети додатку(рис. 2.6).

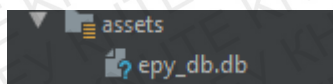


Рис. 2.6 Перенесення створеної бази даних до додатку

```
package com.example.diplomepy.database;

import androidx.room.*;

@Database(entities = {MainTest.class, MainTestPart2.class, TextTestModel.class, ImageTestModel.class, DictModel.class, QuestionTestModel.class, AnswerTestModel.class}, version = 1)
public abstract class MainDatabase extends RoomDatabase {

    private static MainDatabase instance;
    private static ExecutorService executorService = Executors.newFixedThreadPool(4);

    public static synchronized MainDatabase getInstance(Context context) {
        if (instance == null) {
            instance = Room.databaseBuilder(context.getApplicationContext(),
                MainDatabase.class, "main")
                .createFromAsset("epy_db.db")
                .build();
        }
        return instance;
    }

    public abstract MainPart2Dao mainPart2Dao();

    public abstract TextDao textDao();

    public abstract PictureDao pictureDao();

    public abstract DictDao dictDao();

    public abstract MainDao mainDao();

    public abstract QuestionDao questionDao();

    public abstract AnswerDao answerDao();
}
```

Рис. 2.7. Створення класу MainDatabase

Для безперебійного використання бази даних потрібно створити «ентітки» кожної таблиці(рис. 2.8).

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			26

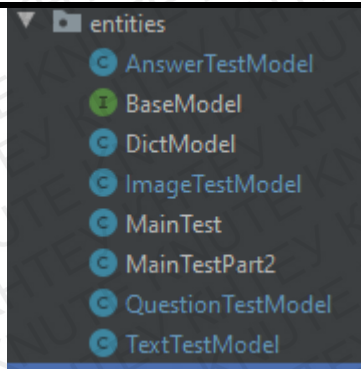


Рис. 2.8. Створення класів для кожної таблиці з бази даних

Приклад опису одного з класів бази даних (передача полів таблиці «main») вказано на рис. 2.9.

```
package com.example.diplomepy.databse.entities;

import androidx.room.Entity;
import androidx.room.PrimaryKey;

@Entity(tableName = "main")
public class MainTest {

    @PrimaryKey(autoGenerate = true)
    private int id_main;
    private String name;
    private int counter;
    private int blocked;
    public MainTest(String name, int counter, int blocked, int id_main) {
        this.name = name;
        this.counter = counter;
        this.blocked = blocked;
        this.id_main = id_main;
    }

    public int getId_main() { return id_main; }
    public String getName() { return name; }
    public int getCounter() { return counter; }
    public int getBlocked(){return blocked;}

    public void setCounter(int counter) { this.counter = counter; }
}
```

Рис. 2.9. Приклад передачі таблиці «main»

						Архив
						27
Зм.	Архив	№ докум	Підпис	Дата		

КНТЕУ 121 02з-13.МР

Наступним кроком є передача інтерфейсів(запитів) для кожної з цих таблиць. Було створено 10 таблиць та 14 інтерфейсів. Це все обумовлено тим, що інтерфейси відповідають за надання необхідних даних користувачу. Саме тому було створено 2 додаткових інтерфейси які об'єднують декілька таблиць в одну і виводять лише необхідні дані. Створені інтерфейси необхідно помістити в окремий package та назвати dao(рис. 2.10).

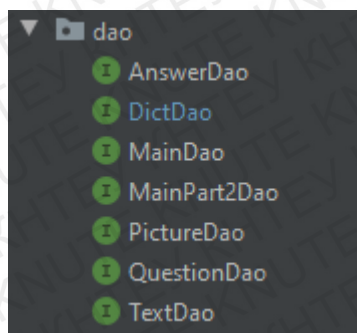


Рис. 2.10. Створені інтерфейси

Приклад створення одного з інтерфейсів вказано на рис. 2.10.

Наступним і останнім кроком для повноцінного функціонування бази даних є створення Адаптеру для кожної таблиці. Адаптер приймає значення з таблиці, в ньому вказується що необхідно робити з цими даними і які дані необхідно взяти з таблиці. Також він заносить дані до таблиці. Приклад створення такого адаптеру для таблиці «price list» можливо переглянути на рис. 2.11. Це найменший з адаптерів, адже він тільки приймає та передає дані. Решта адаптерів є складними і включають в себе від 100 стрічок коду.

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			28

```

package com.example.diplomepy.ui.question.adapter;

import ...

public class QuestionAdapter extends FragmentStateAdapter {

    private int countPage = 0;

    public QuestionAdapter(@NonNull FragmentActivity fragmentActivity) { super(fragmentActivity); }

    public void setCount(int countPage) { this.countPage = countPage; }

    @NonNull
    @Override
    public Fragment createFragment(int position) { return ArrayListFragment.newInstance(position); }

    @Override
    public int getItemCount() { return countPage; }
}

```

Рис. 2.11. Створення адаптеру для таблиці «price list»

Також через необхідність отримувати та виводити на один фрагмент дані з різних таблиць необхідно було створити особливий клас. В розробці мобільних додатків він має назву «сінглтон».(рис. 2.12). При створенні фрагменту, що виводить інформацію на екран неможливо в одному класі фрагменту виводити декілька списків таблиць. Саме тому створюється клас «сінглтон» який бере необхідні дані з таблиць і передає в різні фрагменти.

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			29

```

package com.example.diplomepy.u1.question;

import ...

public class Singleton {

    private static volatile Singleton instance;

    private int answeredCounter = 0;
    private int currentTestId = 0;
    private List<QuestionTestModel> lists;
    private List<MainTestPart2> testList;
    private List<MainTest> mainTestList;
    private int testMainId;

    public static Singleton getInstance() {
        Singleton localInstance = instance;
        if (localInstance == null) {
            synchronized (Singleton.class) {
                localInstance = instance;
                if (localInstance == null) {
                    instance = localInstance = new Singleton();
                }
            }
        }
        return localInstance;
    }

    void setListQuestions(List<QuestionTestModel> questions) { lists = questions; }

    public List<QuestionTestModel> getListQuestion() { return lists; }

    public void setListPart2(List<MainTestPart2> mainTestPart2s) { testList = mainTestPart2s; }

    public int getSizeListPart2() { return testList.size(); }

    public MainTestPart2 getTestByIndex(int index) { return testList.get(index); }

    public void setCurrentTestId(int id){
        if(id + 1 < testList.size()) {
            currentTestId = id + 1;
        }
    }
}

```

Рис. 2.12. Частина класу «Singleton»

Під час роботи з базою даних було створено 10 класів таблиць, 14 інтерфейсів для кожної з цих таблиць, та 2 віртуальних, життєвий цикл яких триває тільки під час запуску окремих функцій, 7 адаптерів для кожної з таблиць, особливий клас, що приймає та перенаправляє дані з таблиць, головний клас бази даних. Були створені запити для кожної з таблиць. У всіх таблицях використовуються 4 методи SELECT, INSERT, UPDATE та DELETE.

						Архив
						30
Зм.	Архив	№ докум	Підпис	Дата		

КНТЕУ 121 02з-13.МР

2.5. Висновки до розділу 2

У цьому розділі було детально описано структуру бази даних. Було описано кожну таблицю та призначення кожного стовпчику і взаємодію між ними. Розглянуто взаємодію Google Firebase та Android Studio на прикладі створення власного мобільного додатку. Було досліджено всі особливості створення бази даних для мобільного додатку, а також створення окремих класів. Для безперебійного функціонування бази даних на мобільних пристроях необхідно створити *Адаптер*, а для використання даних з різних таблиць в одному фрагменті клас «Singleton».

Було наведено приклади створення звязку кожної таблиці та класу, що відповідає цій таблиці. Наведено приклад створення інтерфейсу (запитів до бази даних за допомогою Android Studio).

Окремо було показано створення класу бази даних для підключення існуючої бази даних в мобільний додаток.

									Аркуш
									31
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 02з-13.МР				

РОЗДІЛ 3

РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО ДОДАТКУ

3.1. Розробка інтерфейсу та дизайну мобільного додатку

Для створення інтерфейсу програми вирішено було використовувати xml фрагменти в середовищі розробки Android Studio. Найкращим рішенням поставлених задач є перезаписання деяких фрагментів для повторного використання. Дані, що приходять з бази даних будуть відправлятися на один і той же фрагмент де вони будуть перезаписуватися відповідно до обраного індексу.

Перед початком роботи потрібно визначитися з поставленими задачами. Необхідно створити бокове меню, а так же необхідні вікна додатку (головна, курс, словник, налаштування, сайти та про нас). Для початку було створено бокове меню засобами Android studio(рис. 3.1).

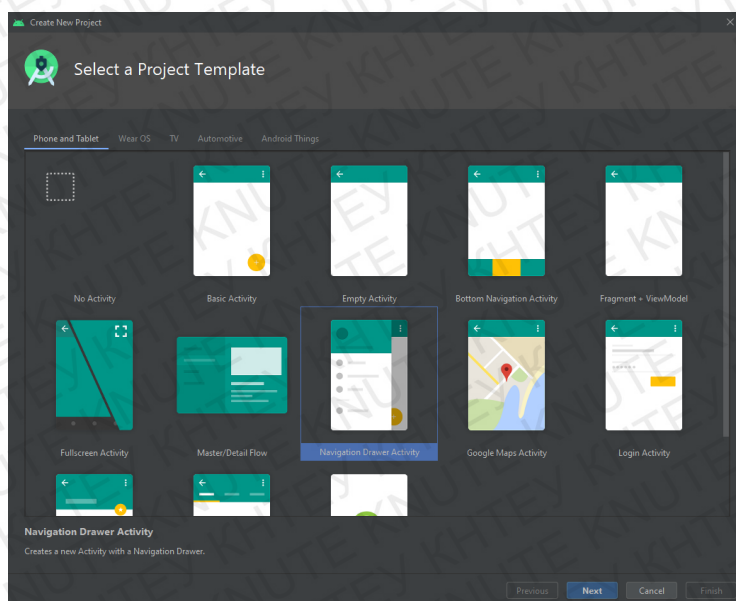


Рис. 3.1. Створення проекту за допомогою шаблону Navigation drawer activity

					КНТЕУ 121 02з-13.МР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.		21.06.2021	Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО ДОДАТКУ	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		21.06.2021		РЗ	32	61
Гарант		Токар В.В.		21.06.2021		Факультет інформаційних технологій 2м курс, 2 група		
Розроб.		Суржик М.О.		21.06.2021				

Наступним кроком створення проекту є робота з окремими вікнами (далі – фрагменти). Для початку потрібно створити прості вікна: Головна, сайти, про нас. Для створення кожного з цих фрагментів потрібно створити окремий xml-файл – розмітку нашого вікна(саме ця складова відповідає за зовнішню складову нашого фрагменту). Приклад створеного дизайну показано на рис. 3.2.

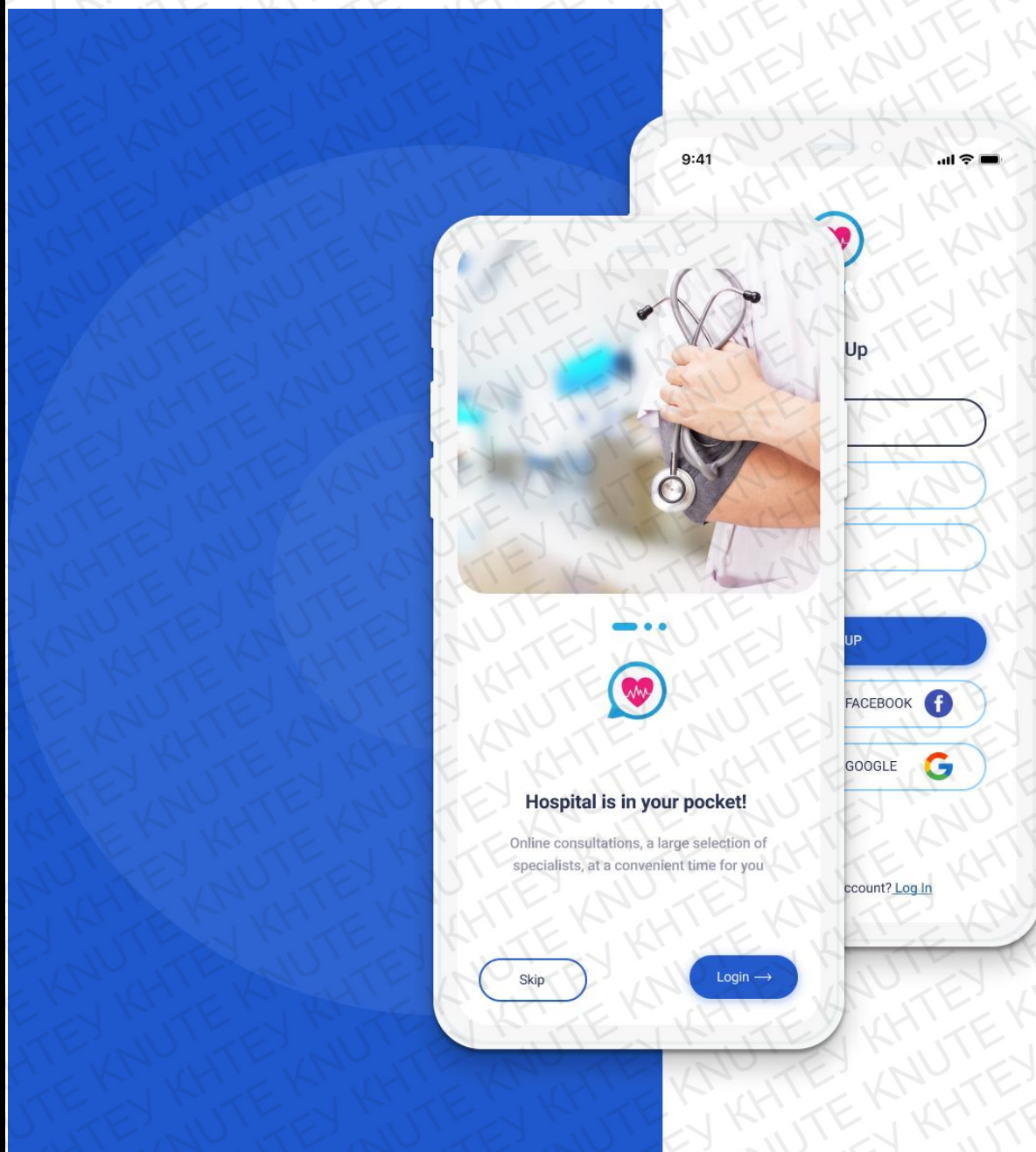


Рис. 3.2. Приклад створення дизайну для вікна логінінгу.

						Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 02з-13.МР	

Наступним кроком в створенні фрагменту є його програмний опис. Для цього потрібно створити окремий клас, який буде відповідати за цей фрагмент, а також за всі кнопки та інші складові з якими можлива взаємодія. Для створення класу потрібно перейти в директорію в якій зберігаються всі класи проекту. Нижче буде приведено приклад опису класу «Головна» для виведення інформації користувачу при переході на обраний фрагмент:

```
public class HomeFragment extends Fragment {

    public View onCreateView(@NonNull LayoutInflater inflater,
        ViewGroup container, Bundle savedInstanceState) {
        View root = inflater.inflate(R.layout.fragment_home, container, false);
        return root;
    }
}
```

А також приклад коду для переходу на сторонні ресурси(рис. 3.3). Для цього потрібно імпортувати бібліотеку UML та парсингу для посилань. Для доступу до створених кнопок необхідно використовувати Bundle.

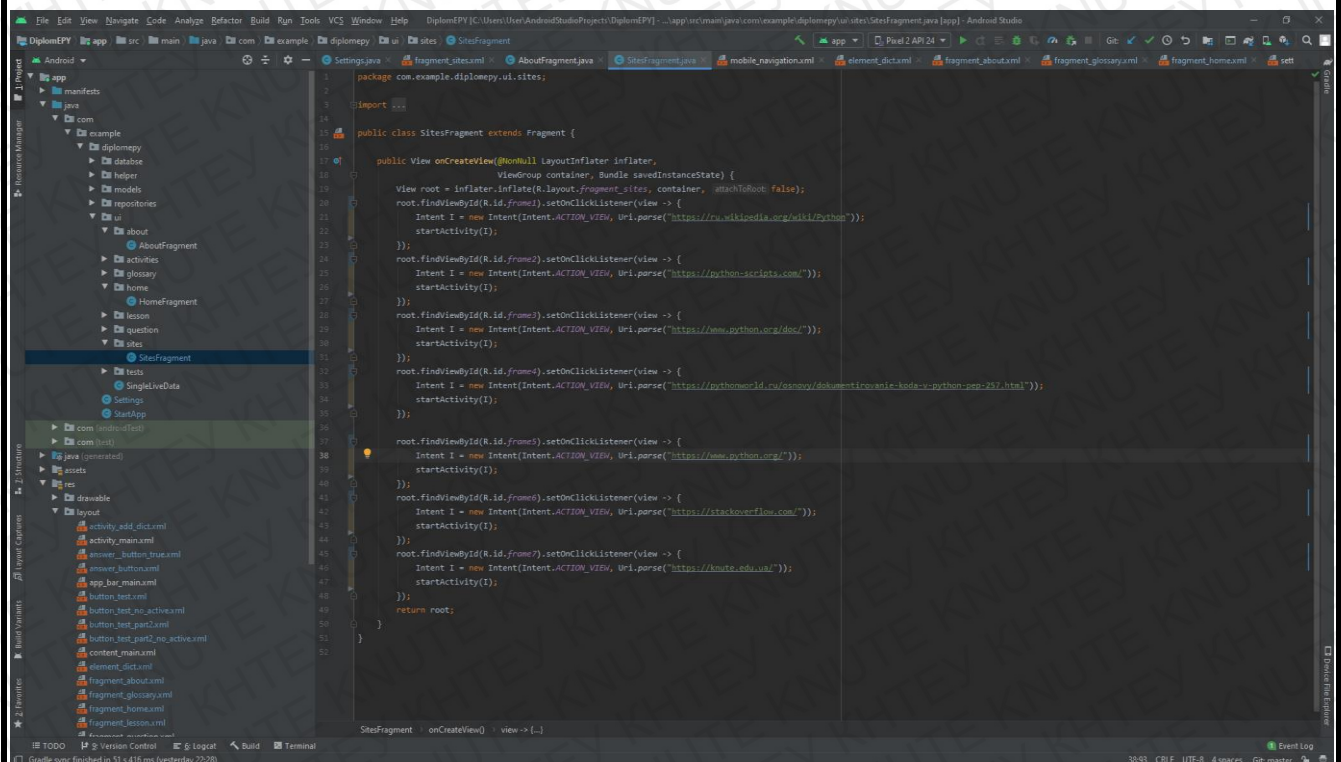


Рис. 3.3. Створення класу «FragmentSites» та парсинг всідних посилань

						Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

КНТЕУ 121 023-13.МР

Основні моменти зі створенням фрагменту закінчено. Для більш детального та функціонального створення фрагменту потрібно використовувати різні класи та стилі.

3.2. Розробка навігації мобільного додатку

Під час створення додатку з шаблону було створено стандартне меню з навігації. Воно потребує модифікацій та змін. Для початку було створено всі головні фрагменти мобільного додатку. Для того, щоб їх звязати необхідно в клас «MainActivity» внести зміни (рис. 3.4).

```
public class MainActivity extends AppCompatActivity {
    private AppBarConfiguration mAppBarConfiguration;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        final String text = PreferenceManager.getDefaultSharedPreferences( context: this)
            .getString("text", "@string/text_medium");

        TextHelper.setText(text, context: this, recreateApp: null);

        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);
        mAppBarConfiguration = new AppBarConfiguration.Builder(
            R.id.nav_home, R.id.nav_settings, R.id.nav_glossary, R.id.nav_about, R.id.nav_test, R.id.nav_sites, R.id.nav_settings)
            .setDrawerLayout(drawer)
            .build();
        NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController( activity: this, navController, mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment);
        return NavigationUI.navigateUp(navController, mAppBarConfiguration)
            || super.onSupportNavigateUp();
    }
}
```

Рис. 3.4. Клас «MainActivity» який є головним координуючим класом в мобільному додатку

						Архив
						35
Зм.	Архив	№ докум	Підпис	Дата		

КНТЕУ 121 02з-13.МР

В цьому класі необхідно імпортувати налаштування навігатора, а також вказати всі створені раніше фрагменти по їх id. Після цього потрібно оголосити сам навігатор та створити(або перейти) до створеного меню навігації (рис 3.5).

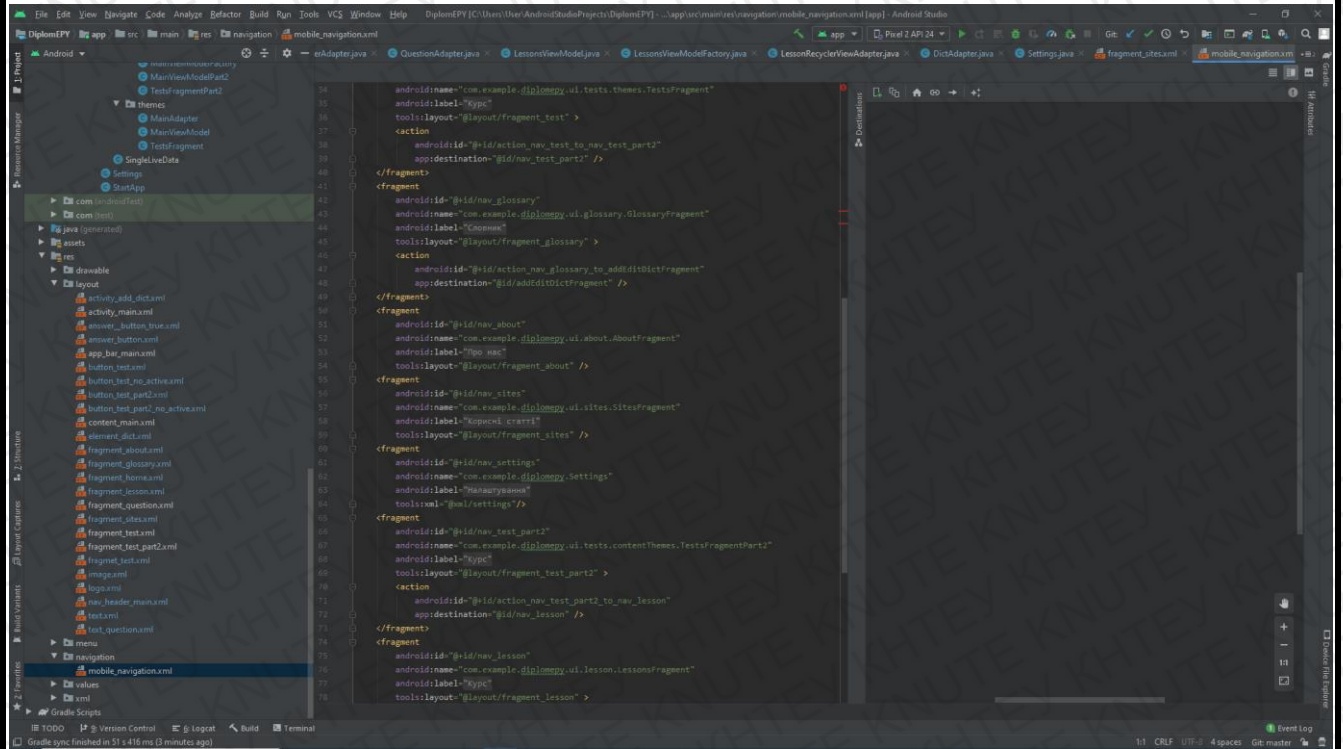


Рис. 3.5. Меню навігації

В меню навігації необхідно оголосити всі фрагменти, а також дати їм назву, яка буде відображатися вверху вікна. Також потрібно прописати всі зв'язки з додатковими вікнами, які були створені. Це необхідно зробити для того, щоб користувач мав змогу повертатися на попередні фрагменти.

Для безперебійного використання бокового меню потрібно створити окремий віртуальний інтерфейс, який буде виконуватися лише при натисканні на необхідну кнопку в створеному «тулбарі». Також цей інтерфейс потрібно зв'язати з головною сторінкою, для того, що би після запуску додатку користувач відразу міг використовувати бокове меню.

						Аркуш
						36
Зм.	Аркуш	№ докум	Підпис	Дата		

Також до навігації відноситься вікно запуску додатку, яке з'являється перед користувачем від час першого запуску додатку. Воно має назву – стартовий фрагмент. Для його створення потрібно створити окремий xml-файл та клас, що наслідується від материнського класу Activity. Приклад створення класу та xml-файлу показано на рис. 3.6. та рис 3.7.

```

package com.example.diplomepy.ui.activities;

import ...

public class LogoActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Logo);

        Window w = getWindow();
        w.setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        new Handler().postDelayed(() -> {
            Intent i = new Intent( packageContext: LogoActivity.this, MainActivity.class);
            startActivity(i);
            finish();
        }, delayMillis: 3000);
    }
}

```

Рис. 3.6. Створення класу «LogoActivity»

Також потрібно налаштувати сам запуск додатку, та порядок виконання Activity при його запуску. Стандартні налаштування спочатку запускають головну Activity => MainActivity. Для того, щоб це виправити потрібно внести зміни до маніфесту створюваного додатку, та вказати, що першим повинен запускатися стартовий фрагмент(рис. 3.8).

						Архиви
						37
Зм.	Архиви	№ докум	Підпис	Дата	КНТЕУ 121 02з-13.МР	

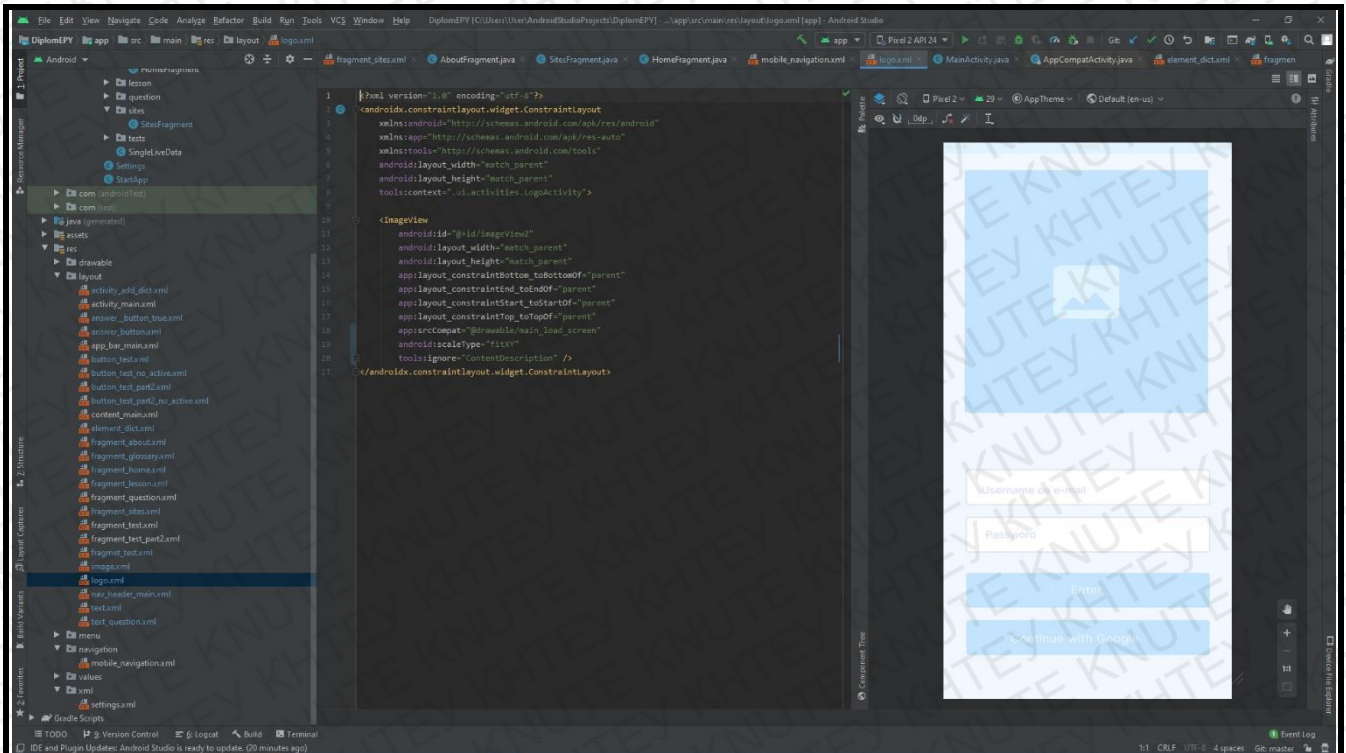


Рис. 3.7. Створення xml-файлу для стартового фрагменту

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.diplomepy">

    <application
        android:name=".StartApp"
        android:allowBackup="true"
        android:icon="@drawable/logo2"
        android:label="EPY"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ui.activities.LogoActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ui.activities.MainActivity"
            android:label="EPY"
            android:theme="@style/AppTheme.NoActionBar" />
    </application>
</manifest>

```

Рис. 3.8. Маніфест створюваного додатку

						Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

КНТЕУ 121 02з-13.МР

3.3. Створення налаштувань мобільного додатку

Створення налаштувань один з найбільш складних процесів. Адже після закриття додатку чи переходу з одного фрагменту до іншого всі налаштування повинні зберігатись до наступного входу, та використовуватись додатком при його активації. Для цього в Android Studio було використано спеціальні налаштування, які мають назву SharedPreferences. Це спеціальна бібліотека, що записує зміни додатку окремим файлом, та зчитує його при кожному завантаженні додатку.

Для початку потрібно в папці res створити окрему директорию, яка буде наслідувати SharedPreferences. Потім створюється xml-файл в якому вказується інтерфейс майбутніх налаштувань. Різниця звичайного файлу і файлу налаштувань полягає в внутрішньому конструкторі який надається Android Studio. Приклад створеного файлу налаштувань вказано на рис. 3.9.

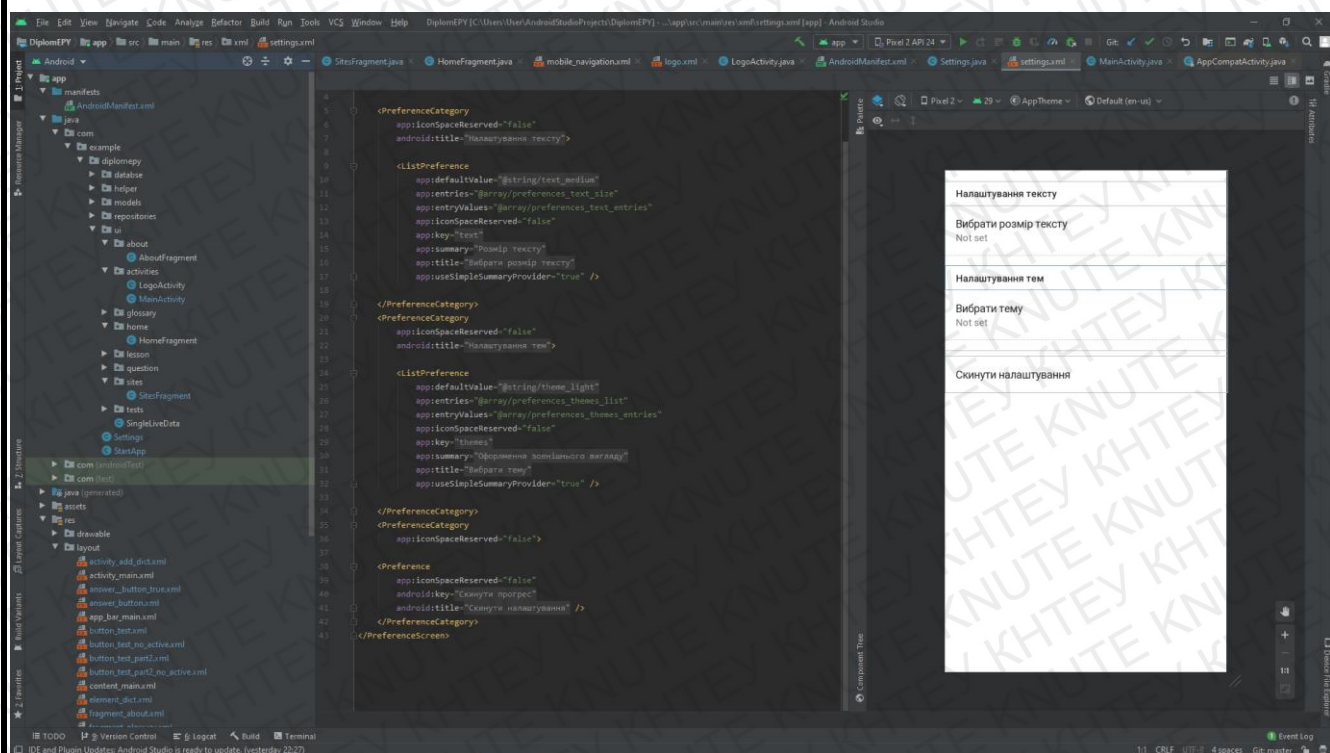


Рис. 3.9. xml-файл налаштувань мобільного додатку

					Аркуш
					39
Зм.	Аркуш	№ докум	Підпис	Дата	

Наступним кроком є створення окремого класу, який буде наслідувати `SharedPreferences` а також буде материнським для двох інших класів(`SharedPreferences.OnSharedPreferencesChangeListener`, `TextHelper.RecreateApp`). В цьому класі буде описано основні моменти налаштувань в додатку. Так наприклад головний клас налаштувань отримує методи з двох дочірніх класів для вибору необхідної теми та для необхідного розміру тексту, який вказує користувач при користуванні додатком. Приклад головного класу налаштувань вказано на рис 3.10.

```

package com.example.diplomepy;

import androidx.preference.PreferenceFragmentCompat;
import androidx.preference.SharedPreferences;
import androidx.preference.SharedPreferences.OnSharedPreferencesChangeListener;
import androidx.preference.TextHelper.RecreateApp;

public class Settings extends PreferenceFragmentCompat implements
    SharedPreferences.OnSharedPreferencesChangeListener, TextHelper.RecreateApp {

    @Override
    public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
        addPreferencesFromResource(R.xml.settings);

        final AlertDialog alertDialog = new AlertDialog.Builder(this)
            .setTitle("Вибір теми")
            .setPositiveButton("ОК", null)
            .setNegativeButton("Відміна", null)
            .create();

        Preference button = findPreference("Стильняк_портрет");
        button.setOnPreferenceClickListener(preference -> {
            alertDialog.show(getChildFragmentManager(), tag: "some_text");
        });

        return true;
    }

    @Override
    public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
        if (key.equals("theme")) {
            final Context context = getContext();

            if (context != null) {
                final String theme = sharedPreferences.getString(key, "@string/theme_light");
                TextHelper.setTheme(theme, context);
            }
        }

        if (key.equals("text_size")) {
            final Context context = getContext();

            if (context != null && getActivity() != null) {
                final String text = sharedPreferences.getString(key, "@string/text_medium");
                TextHelper.setText(context, getActivity(), text);
            }
        }
    }
}

```

Рис. 3.10. Створення основного класу налаштувань для мобільного додатку

Для того, щоб у додатку було декілька тем потрібно в ресурсах створити нові файли з кольорами. На рис 3.11 – приклад створення такого файлу для темної теми.

									Аркуш
									40
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#AA1919</color>
  <color name="colorPrimaryDark">#000000</color>
  <color name="colorAccent">#AA1919</color>
</resources>
```

Рис. 3.11. Створення нового ресурсу для нічного варіанту теми

Для зміни тексту потрібно створити новий файл стилів, в якому необхідно зазначити необхідні назви розміру шрифту а також їх локальний розмір. На рис 3.12 – приклад створеного файлу для зберігання розмірів та позначень шрифту.

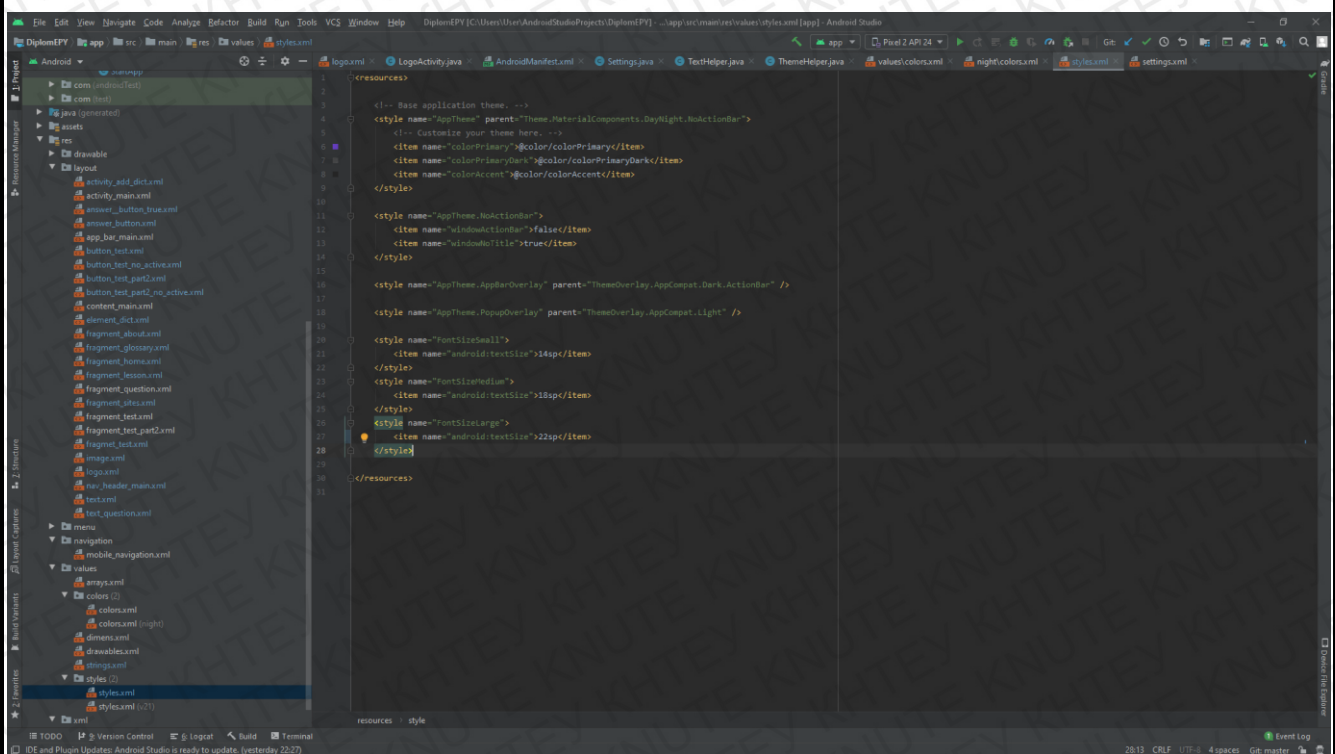


Рис. 3.12. Створення нових стилів для регулювання розміру шрифту за допомогою налаштувань

Також для того, щоб програма розуміла звідки потрібно брати ці ресурси, необхідно створити два допоміжні класи – для тем та для тексту. Назви ThemeHelper та TextHelper відповідно. В них потрібно сказати які ресурси беруться, та куди передаються. Останньою функцією налаштувань є функція

						Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

КНТЕУ 121 02з-13.МР

скидування налаштувань до початкових. Для коректної роботи та уникнення випадків випадкового натискання цієї кнопки було розроблено ще один клас, що відповідає за виклик AlertDialog – діалогових вікон з підтвердженням операції. Приклад створення цього класу показано на рис. 3.13.

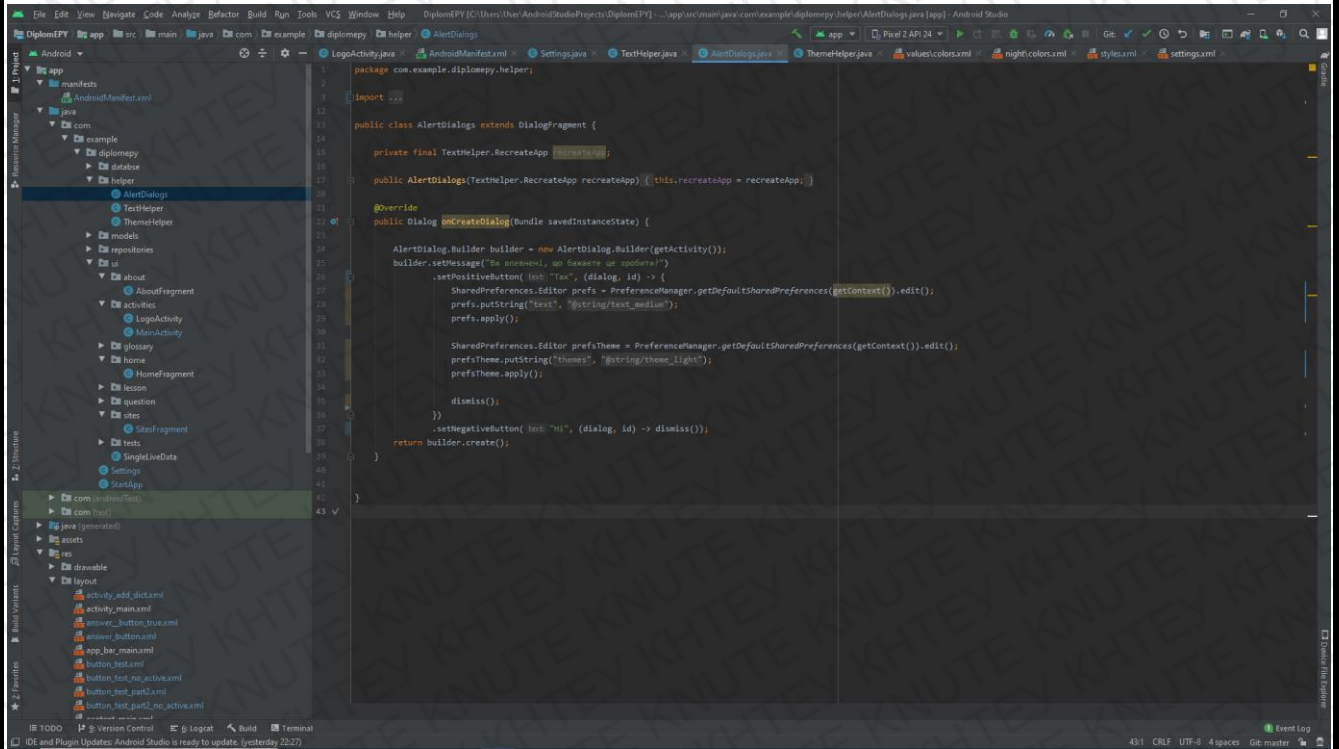


Рис. 3.13. Створення класу AlertDialog

					Аркуш
					42
Зм.	Аркуш	№ докум	Підпис	Дата	

3.4 Висновки до розділу 3

В цьому розділі детально описано створення мобільного додатку, а також всіх його складових.

Інтерфейс не є складним, і користувач легко зможе орієнтуватися в ньому. Приємні кольори підходять для всіх категорій людей. Дизайн створено згідно сучасних вимог.

Навігація повинна бути розроблена згідно офіційних вимог, працювати безперебійно та не мати прихованих багів. Користувач повинен вільно пересуватися всередині додатку в рамках доступу користувача.

Налаштування були мінімізовані, що дозволило вивести користувачу загальні налаштування додатку, для спрощення його використання.

					КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		43

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Результати проведеного дослідження дають підставу зробити такі висновки: Досліджено потреби даної програми для лікарень та пацієнтів, а також проведено аналіз її аналогів, які зараз використовуються в Україні.

Спроектовано створення власної бази даних та перевірка взаємодії Google Firebase та Android Studio. Під час створення власної бази даних було проведено дослідження та досліджено таблиці, типи таблиць, створення та використання запитів, для полегшення та об'єднання таблиць в одну цілісну базу даних. Також було проведено детальний розбір коду продукту, та описано кожну частину його функціональної складової (кнопки, пошук, переходи).

В третьому розділі детально описано етапи створення мобільного додатку, а також всіх його складових.

На основі виконаних завдань було створено власну програму «GN», а також проведено дослідження актуальності створеної програми на сьогоднішній день. З цього можливо зробити висновки, що всі поставлені завдання виконано і дослідження проведено.

Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 02з-13.МР			
Зав. кафедри		Криворучко О.В.		21.06.2021	Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		21.06.2021		ВП	44	61
Гарант		Токар В.В.		21.06.2021		Факультет інформаційних технологій		
Розроб.		Суржик М.О.				2м курс, 2 група		
					ВИСНОВКИ ТА ПРОПОЗИЦІЇ			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. 8-е изд. / М.: Вильямс, 2005. — 843 с.
2. Кренке Д. Теория и практика построения баз данных. / Д. Кренке – Спб.: Питер, 2005. – 844 с.
3. Герберт Шилдт. «Java 8. Руководство для начинающих». / Шилдт Г., 2014 – 139 с
4. Брюс Еккель. «Философия Java». / Еккель Б., 2016 – 40 с
5. Герберт Шилдт. С# 4.0: полное руководство. 1-е изд. / Шилдт Г., 2011 – 189 с.
6. Кристиан Нейгел. С # 4.0 і платформа .NET 4 для професіоналів.” Windows Forms”. 1-е изд. /К.Нейгел, Б.Ивьен, Глинн, К.Уотсон, М.Скиннер, 2011. – глава 39, – 1111 с.
7. Джозеф Албахари, Бен Албахари. С# 6.0. Справочник. Полное описание языка / Албахари Д. - Албахари Б, 2016. – 119с.
8. Пол Дейтел, Харви Дейтел. Как программировать на Visual С# 2012 / Дейтел П. – Дейтел Х, 2014 – 646с.
9. Рихтер Джордж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. / Ритхер Д., 2018 – 401 с.
10. Хортон А. Visual C++ 2008: базовый курс. Visual Studio®/ А. Хортон. – К.: Диалектика, 2009. – 150 с.
11. Когаловский М. Р. Энциклопедия технологий баз данных. 1-е изд. / М.: Финансы и статистика, 2002. — 800 с.
12. Дейт К. Дж. Введение в системы баз данных = Introduction to Database

					КНТЕУ 121 02з-13.МР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.		27.02.2021	Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		27.02.2021		СВД	45	61
Гарант		Токар В.В.		27.02.2021		Факультет інформаційних технологій 2м курс, 2 група		
Розроб.		Суржик М.О.		27.02.2021				
					ТЕХНІЧНЕ ЗАВДАННЯ			

Systems. 8-е изд. / М.: Вильямс, 2005. — 843 с.

Интернет ресурсы:

1. GitHub [Электронный ресурс] – Режим доступа до ресурсу:
<https://github.com/Maks4u>
2. AndroidDocumentation [Электронный ресурс] – Режим доступа до ресурсу:
<https://developer.android.com/reference/android/content/res/Configuration.html#fontScale>
3. CodingInFlow [Электронный ресурс] – Режим доступа до ресурсу:
<https://codinginflow.com/>
4. Освой Android играючи [Электронный ресурс] – Режим доступа до ресурсу:
<https://developer.alexanderklimov.ru/android/>
5. Stackoverflow [Электронный ресурс] – Режим доступа до ресурсу:
<https://stackoverflow.com/>

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			46

Технічне завдання

1. Загальні відомості

1.1. Найменування додатку – «Good Health».

1.1.1. Повне найменування додатку – «Good Health»

1.1.2. Скорочене найменування додатку – «GH»

1.2. Планові терміни початку та закінчення робіт

1.2.1. Початок робіт – 01.06.2021

1.2.2. Закінчення робіт – 01.11.2021

1.3. Порядок та пред'явлення результатів робіт – пред'явити готовий проект керівнику

1.4. Потенційні користувачі системи – люди віком від 16 років, які бажають слідкувати за своїм здоров'ям та відвідати лікарню.

2. Мета та призначення створення додатку

2.1. Призначення додатку – поліпшення умов відвідування лікарні, інформування користувача про здані аналізи та процедури

2.2. Мета створення додатку – створити додаток для поліпшення відвідування мільких лікарень

3. Вимоги до додатку

3.1. Вимоги до додатку в цілому – у додатку повинна бути реалізована можливість запису на прийом, перегляд власної мед карти, діагнозів, ліків, перегляду лікарів, виставка їм оцінок, перегляд прайсу, зміни нашлаштувань додатку.

3.1.1. Вимоги до структури та функціонування додатку, перелік підсистем – додаток базується на мові програмування Java виконаного в Android Studio з підключенням внутрішніх бібліотек та особливостей Gradle.

Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 02з-13.МР			
Зав. кафедри		Криворучко О.В.		27.02.2021	Проектування довідниково-інформаційної системи взаємодії лікарів та пацієнтів для міських лікарень	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		27.02.2021		ТЗ	47	61
Гарант		Токар В.В.		27.02.2021		Факультет інформаційних технологій 2м курс, 2 група		
Розроб.		Суржик М.О.		27.02.2021				
					ТЕХНІЧНЕ ЗАВДАННЯ			

- 3.1.1.1. Вимоги до засобів інформаційного обміну між компонентами додатку – зв'язок між компонентами додатку буде виконувати 2 окремі структури: Navigation та Google Firebase.
- 3.1.1.2. Вимоги до режимів функціонування додатку – додаток повинен функціонувати у режимі онлайн.
- 3.1.1.3. Вимоги до діагностування додатку – додаток повинен працювати безперебійно та успішно виконувати усі процеси.
- 3.1.1.4. Вимоги до режимів управління додатком – повинно бути реалізовано режим користувача(лікаря та пацієнта), режим системного адміністратора.
- 3.1.2. Показники призначення
- 3.1.2.1. Параметри, що характеризують ступінь відповідності системи призначенням – безперебійність роботи системи, відсутність помилок.
- 3.1.2.2. Вимоги до пристосованості додатку до змін – в додатку буде реалізовано живий пошук по лікарям а також записуватись на прийом да скасовувати його. Буде створено меню налаштувань, яке буде змінювати роботу всієї системи.
- 3.1.2.3. Вимоги до збереження працездатності додатку в різних ймовірних умовах – додаток буде зберігати дані після виходу, всі завершені теми зберігаться, налаштування після виходу з додатку зберігаються, при зверненні додатку він стає на «паузу» а не вимикається.
- 3.1.3. Вимоги до надійності
- 3.1.3.1. Вимоги до методів оцінки і контролю показників надійності на різних стадіях створення додатку – на стадії тестування дозволена поява помилок, на стадії здачі додатку не дозволена поява помилок.

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			48

3.1.4. Вимоги до ергономіки та технічної естетики – зі знярядь праці необхідний комп'ютер на робоче місце а також наявність телефону з операційною системою Android не нижче версії 6.0.

3.1.5. Вимоги до захисту інформації від несанкціонованого доступу – доступ до інформації надається тільки користувачу додатку, в окремих випадках доступ до інформації може отримувати системний адміністратор додатку.

3.1.6. Вимоги до захисту від впливу зовнішніх факторів – додаток приймає зміни тільки безпосередньо від операційної системи (встановлення системної теми).

3.1.7. Вимоги безпеки – додаток повинен бути безпечним для його користувачів.

3.2. Перелік підсистем додатку – Google Firebase.

3.3. Вимоги до видів забезпечення:

3.3.1. Вимоги до інформаційного забезпечення – інформаційне забезпечення повинне бути цілісним і захищеним від несанкціонованого доступу.

3.3.1.1. Вимоги до складу, структури і способів організації даних в додатку – усі дані повинні зберігатись з-за допомогою баз даних і управлятись СУБД.

3.3.1.2. Вимоги до інформаційного обміну між компонентами системи - передача інформації між компонентами системи має виконуватись стандартними протоколами на рівні програмного забезпечення або на рівні платформи (системи керування БД).

3.3.1.3. Вимоги щодо застосування систем управління базами даних – потрібно створити адаптери для кожної таблиці бази даних, з яких будуть вноситись дані.

									Аркуш
									49
Зм.	Аркуш	№ докум	Підпис	Дата					

КНТЕУ 121 02з-13.МР

3.3.1.4. Вимоги до структури процесу збору, обробки, передачі даних в системі представлення даних – процес збору, обробки та передачі даних в системі повинен безпечним та швидким.

3.3.1.5. Вимоги до захисту даних від руйнувань при аваріях і збоях в електроживленні системи – аварійний вихід з додатку зі збереженням даних в пам'яті телефону.

4. За допомогою додатку користувачі повинні мати змогу обирати необхідного лікаря, отримувати з нього інформацію, розпочинати чатування, записуватись до нього на прийом, скасовувати прийом до лікаря, виставляти оцінку після зустрічі. Також користувачі повинні мати змогу використовувати бокове меню для перегляду цін, доступних лікарів та власної інформації що знаходиться в медичній картці, в стрічці пошуку буде реалізовано живий пошук по лікарям, а також додавання та видалення лікарів зі списку улюблених. Користувач зможе в меню налаштувань налаштувати систему під себе, змінювати тему а також розмір шрифту у всьому додатку, та скинути налаштування до початкових. Програмне забезпечення повинно базуватись на використанні:

4.1. Android Studio 3.7.1

4.2. Java

4.3. Google Firebase

4.4. Бібліотеки Google Gradle

5. Для технічного забезпечення рекомендовано використовувати мобільний пристрій з такими характеристиками:

5.1. Android 6.0 та вище

5.2. Qualcomm Snapdragon S1 та вище

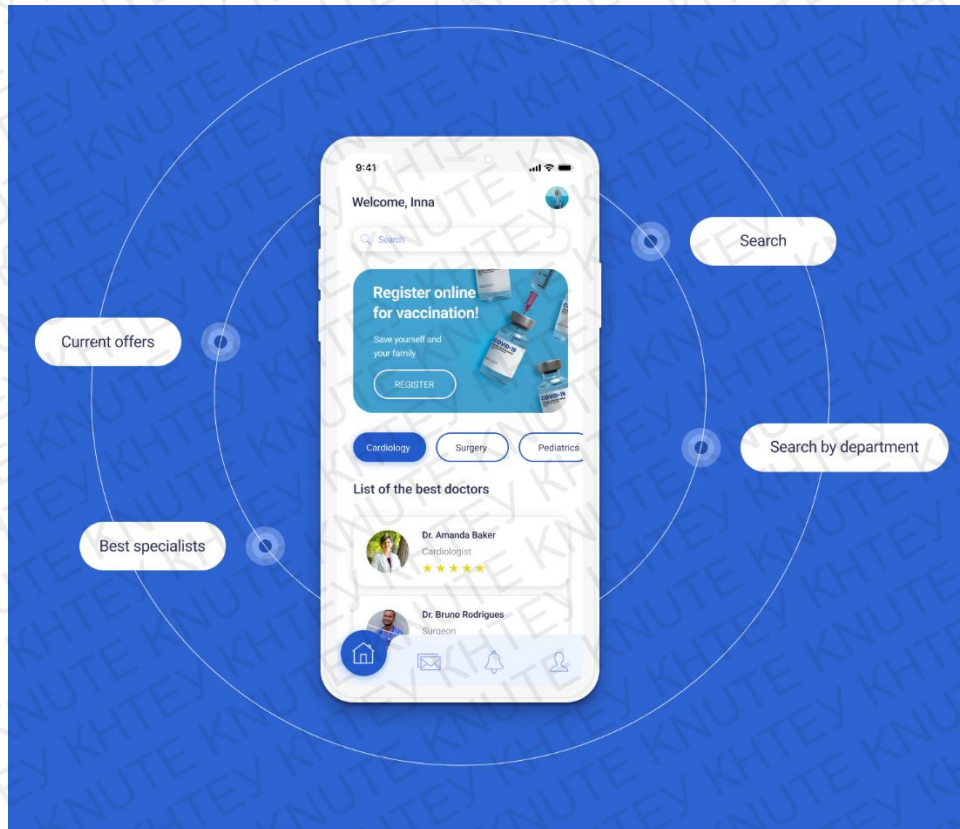
5.3. 1 гігабайт оперативної пам'яті

6. Вимоги до методичного забезпечення – не передбачаються, адже використання системи буде інтелектуально простим та зрозумілим.

						КНТЕУ 121 02з-13.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			50

ДОДАТКИ

Додаток А



USER FLOW



Додаток Б

Now it's time to drink FLEBODIA
10 min ago

You have an online consultation with Dr. Sheldon in 15 minutes
20 min ago

Now it's time to drink FLEBODIA
1 day ago

Now it's time to drink FLEBODIA
2 day ago

You had a consultation with Dr. Norson. Please, leave your review about the doctor.
2 day ago
[Leave review](#)

You had a consultation with Dr. Norson.



Username or e-mail

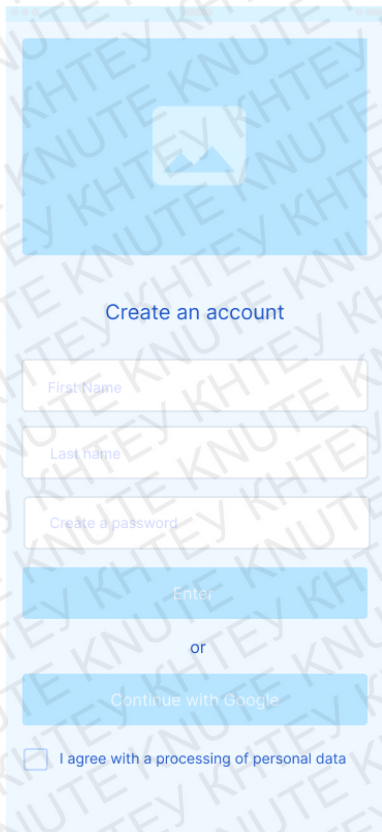
Wrong address

Password

Enter

Continue With Google

[Forgot your password? Restore](#)



Create an account

First Name

Last name

Create a password

Enter

or

Continue with Google

I agree with a processing of personal data

Back Measurements

Type of measurement
Temperature

Periodicity
Everyday

Time and duration
Start
9:00 AM
Finish
9:00 AM

Remind
10 min before

Save Cancel

Welcome, Inna

Q Search

Cardiology Surgery Pediatrics

List of the best doctors

[Doctor Card 1]

[Doctor Card 2]

[Doctor Card 3]

Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення та кібербезпеки
Освітній ступінь магістр
Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«10» листопада 2020 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Суржика Максима Олександровича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «моделювання довідниково-інформаційної системи медичних закладів із особистим кабінетом пацієнта та лікаря на ОС Android»

Затверджена наказом ректора від «28» грудня 2020 р. № 3923 (заочка)

2. Строк здачі студентом закінченого проєкту 25 листопада 2021

3. Цільова установка та вихідні дані до проєкту

Мета проєкту розробка довідниково-інформаційної системи медичних закладів із особистим кабінетом пацієнта та лікаря на ОС Android

Об'єкт дослідження проєктування довідниково-інформаційної системи

Предмет дослідження методи та моделі проєктування довідниково-інформаційної системи системи взаємодії лікарів та пацієнтів для міських лікарень

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ ТА ТЕОРЕТИЧНА ЧАСТИНА БАЗИ ДАНИХ ЛІККАРЕНЬ

1.1. Основні поняття та визначення бази даних клінічної лікарні.....

1.2. Огляд існуючих клінічних систем.....

1.3. Огляд середовища розробки мобільних додатків..... 10

1.4. Висновки до розділу 1

РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ МОБІЛЬНОГО ДОДАТКУ

2.1. Створення бази даних мобільного додатку та її таблиць

2.2. Розгляд полів та таблиць, створення запитів

2.3. Опис коду бази даних мобільного додатку

2.4. Висновки до розділу 2

РОЗДІЛ 3 РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО ДОДАТКУ

3.1. Розробка інтерфейсу та дизайну мобільного додатку.....

3.2. Розробка навігації мобільного додатку

3.3. Створення налаштувань мобільного додатку

3.4. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....

ДОДАТКИ.....

Технічне завдання

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми ВКР</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра</i>	22.12.2020	22.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ ТА ТЕОРЕТИЧНА ЧАСТИНА БАЗИ ДАНИХ ЛІККАРЕНЬ</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. МОДЕЛЮВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ МОБІЛЬНОГО ДОДАТКУ</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ МОБІЛЬНОГО ДОДАТКУ</i>	21.06.2021	21.06.2021
8.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
9.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
10.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
11.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
12.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
13.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
14.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021 25.11.2021	22.11.2021
15.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
16.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
17.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «10» листопада 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту Рзаєва С.Л.
(прізвище, ініціали, підпис)

9. Гарант освітньої програми Токар В.В.
(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Суржик М.О.
(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту _____
(підпис, дата)

Відмітка про попередній захист _____
(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента _____ Суржик М.О.
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____ Токар В.В.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« _____ » _____ 20 _____ р.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці довідниково-інформаційної системи медичних закладів із особистим кабінетом пацієнта та лікаря на ОС Android.

В результаті порівняльного аналізу аналогічних рішень визначено Методи та моделі проектування довідниково-інформаційної системи системи взаємодії лікарів та пацієнтів для міських лікарень.

Розробка серверної частини повинна бути виконана у середовищі Android Studio 7.2, Google Firebase та SQLite.

У відповідності з метою дослідження визначено завдання:

- проаналізувати функції інформаційно-довідникової системи, її збір, обробка даних для подальшого зберігання та надання відомостей користувачам за відповідними запитамі;
- аналіз існуючих середовищ розробки мобільних додатків;
- розробка бази даних;
- проектування архітектури мобільного додатку;
- розробка програмного забезпечення довідниково-інформаційної системи.

Результати проведеного дослідження дають підставу зробити такі висновки: Досліджено потреби даної програми для лікарень та пацієнтів, а також проведено аналіз її аналогів, які зараз використовуються в Україні.

Спроектовано створення власної бази даних та перевірка взаємодії Google Firebase та Android Studio. Під час створення власної бази даних було проведено дослідження та досліджено таблиці, типи таблиць, створення та використання запитів, для полегшення та об'єднання таблиць в одну цілісну базу даних. Також було проведено детальний розбір коду продукту, та описано кожну частину його функціональної складової (кнопки, пошук, переходи).

На основі виконаних завдань було спроектовано власну програму «ГН», а також проведено дослідження актуальності створеної програми на сьогоднішній день. З цього можливо зробити висновки, що всі поставлені завдання виконано і дослідження проведено.

Ключові слова: «ГН», «Firebase», «АС», «База даних», «Мобільний додаток», «довідниково-інформаційна система».

ABSTRACT

In accordance with the purpose of the study, the work is devoted to the development of a reference and information system of medical institutions with a personal account of the patient and doctor on Android.

As a result of the comparative analysis of similar decisions Methods and models of designing of reference and information system of system of interaction of doctors and patients for city hospitals are defined.

The development of the server part must be performed in the environment of Android Studio 7.2, Google Firebase and SQLite.

In accordance with the purpose of the study the task is defined:

- analyze the functions of the reference system, its collection, data processing for further storage and provision of information to users upon request;
- analysis of existing mobile application development environments;
- database development;
- designing the architecture of the mobile application;
- development of reference and information system software.

The results of the study give grounds to draw the following conclusions: The needs of this program for hospitals and patients are studied, as well as the analysis of its analogues, which are currently used in Ukraine.

Designed to create your own database and test the interaction of Google Firebase and Android Studio. While creating our own database, we researched and researched tables, table types, creating and using queries, to facilitate and merge tables into one single database. A detailed analysis of the product code was also performed, and each part of its functional component (buttons, search, transitions) was described.

Based on the completed tasks, the own program "GH" was designed, as well as a study of the relevance of the created program for today. From this it is possible to conclude that all tasks have been completed and the study was conducted.

Keywords: "GH", "Firebase", "AC", "Database", "Mobile application", "reference and information system".