

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка мобільного додатку мовою Java

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-20-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»


(підпис)

Сироти Владислава
Борисовича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Горбовий Артур
Юліанович, д.т.н.,
професор
(прізвище, ім'я, по батькові,
науковий ступінь, вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Омельчук Антон
Анатолійович, к.т.н.
(прізвище, ім'я, по батькові, науковий
ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних
технологій і систем протокол від «22» травня 2024 № 15

Завідувач кафедри


(підпис)

Філоненко Михайло
Миколайович, к.ф.-м.н.,
доцент
(прізвище, ім'я, по батькові,
науковий ступінь, вчене звання)

Реєстрація № 36

24.05.2024
(дата)


(підпис відповідальної особи)

Результати перевірки

до захисту
(до захисту, на доопрацювання)

22.05.2023
(дата)


(підпис)

Результат захисту

5 (відмінно)
(оцінка)

06.06.2024
(дата)


(підпис)

Голова комісії:

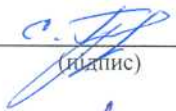

(підпис)

В.М. Франчук

Члени комісії:


(підпис)

М.М. Філоненко


(підпис)

С.М. Грищенко


(підпис)

В.О. Ніжегородцев

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	27		06.06.2024 Франчук В.М.  Філоненко М.М.  Грищенко С.М.  Ніжегородцев В.О.  Суховерхівка Л.П. 
2. Дотримання вимог щодо оформлення	від 0 до 10	8		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		
4. Рецензент	від 0 до 5	5		
РАЗОМ БАЛІВ	від 0 до 50	44		

Оцінювання кваліфікаційної роботи науковим керівником - 46 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
90	A	5 (відмінно)	06.06.2024	Франчук В.М.  Філоненко М.М.  Грищенко С.М.  Ніжегородцев В.О.  Суховерхівка Л.П. 

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій

Кафедра комп'ютерних та інформаційних технологій і систем

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: «Інформаційні управляючі системи і технології (за галузями)»

Затверджую

Завідувач кафедри  Філоненко М.М.

«23» травня 2023

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Сироті Владиславу Борисовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Розробка мобільного додатку мовою Java
Затверджена наказом ректора від «20» жовтня 2023 № 1411
2. Строк здачі закінченої роботи «20» травня 2024
3. Мета роботи Полягає у розробці та впровадженні програми про надання першої допомоги, яка працює в умовах відсутності інтернет-підключення та ефективно допомагає у критичних ситуаціях.
Об'єктом дослідження є можливість надання критичної інформації та термінової допомоги в умовах відсутності інтернет-підключення через додаток, написаний мовою Java.
Предметом дослідження є технічні, медичні та соціальні аспекти розробки та впровадження програми першої допомоги на мобільні девайси.
Орієнтовна база дослідження - Державна служба України з надзвичайних ситуацій.
4. Вихідні дані до роботи: інформація та поради щодо розробки мобільних додатків, основана на відгуках та статтях у інтернеті.
5. Перелік графічного матеріалу: 92 сторінки, 7 таблиць, 12 рисунків, список використаних джерел із 40 найменувань, 5 додатків.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
 - Розробка програмного забезпечення: аналіз технічних можливостей та вимог для розробки програми першої допомоги.
 - Валідація медичних аспектів, забезпечуючи відповідність інформації найвищим медичним стандартам та рекомендаціям.
 - Психологічна адаптація GUI: розробити інтуїтивно зрозумілий та емпатичний інтерфейс, який підтримує користувачів у стресових ситуаціях та сприяє швидкій реакції.
 - Тестування ефективності: перевірка функціоналу та ефективності програми, звертаючи увагу на її реальний вплив у різних ситуаціях.
 - Документація всіх етапів розробки, тестування та вдосконалення програми для створення звіту.

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.10.2023	До 15.10.2023
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	29.12.2023	29.12.2023
3	II розділ (аналіз експериментальної частини про методики досліджень)	15.02.2024	15.02.2024
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	15.04.2024	15.04.2024
5	Висновки	01.05.2024	01.05.2024
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	06.05.2024	06.05.2024
7	Отримання відгуку керівника	28.05.2024	28.05.2024
8	Отримання зовнішньої рецензії	30.05.2024	30.05.2024

8. Дата видачі завдання «23» жовтня 2023

9. Керівник кваліфікаційної роботи


(підпис)

Горбовий А.Ю.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Омельчук А.А.
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Сирота В.Б.
(прізвище, ініціали)

АНОТАЦІЯ

Розробка та впровадження Програми Першої Допомоги яка не вимагає інтернет-підключення.

Мета роботи полягає в розробці та впровадженні програми першої допомоги, яка працює в умовах відсутності інтернет-підключення.

Завданнями роботи є розробка програмного забезпечення, валідація медичних аспектів, психологічна адаптація GUI, тестування продуктивності та аналіз соціального впливу.

Предметом дослідження є програма першої допомоги, а об'єктом – технічні, медичні та соціальні аспекти цієї програми.

Базою дослідження виступає ретельний аналіз літературних джерел, технічний розвиток програмного забезпечення та експертні консультації з медичних питань.

За результатами дослідження сформульовані новаторські підходи до розробки програми першої допомоги, яка може функціонувати в умовах обмеженого чи відсутнього інтернет-підключення.

Отримані результати можуть бути використані для вдосконалення інших систем надання першої допомоги та розробки подібних програм, а також у навчальних закладах для підготовки фахівців з надання медичної допомоги.

Кваліфікаційна робота містить 79 сторінок, 6 таблиць, 16 рисунків, список використаних джерел 40 найменувань, 5 додатків.

Ключові слова: Програма першої допомоги, відсутність інтернет-підключення, технічний розвиток, екстрені ситуації, мобільні додатки, медична екстрена допомога.

ABSTRACT

Development and Implementation of a First Aid Program that Does Not Require Internet Connection.

The aim of the work is the development and implementation of a first aid program that operates in the absence of an Internet connection.

The tasks of the work include the development of software, validation of medical aspects, psychological adaptation of the interface, performance testing, and analysis of social impact.

The subject of the research is the first aid program, while the object is the technical, medical, and social aspects of this program.

The research is based on a thorough analysis of literary sources, technical development of software, and expert consultations on medical issues.

As a result of the study, innovative approaches to the development of a first aid program that can function in conditions of limited or no Internet connection have been formulated.

The obtained results can be used to improve other first aid systems and the development of similar programs, as well as in educational institutions for the training of medical professionals.

The qualification work consists of 79 pages, 6 tables, 16 figures, a list of used sources with 40 titles, and 5 appendices.

Keywords: First aid program, lack of Internet connection, technical development, emergency situations, mobile applications, medical emergency aid.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IDE - Integrated Development Environment (Інтегроване середовище розробки)

UX - User experience (Досвід користувача)

НС - Надзвичайна ситуація

URI - Uniform Resource Identifier (Єдиний ідентифікатор ресурсу)

GUI - Graphical user interface (Графічний інтерфейс користувача)

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ	6
1.1. Аналіз задачі, засобів та методів її вирішення.....	6
1.2. Ретроспектива мови програмування Java: аналіз особливостей та можливостей	11
1.3. Аналіз існуючого програмного забезпечення за тематикою роботи.....	13
1.4. Огляд середовища розробки Android Studio	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	21
2.1. Проектування загального алгоритму роботи програми.....	21
2.2. Розробка програмного забезпечення.....	22
РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ, ЙОГО ТЕСТУВАННЯ ТА ВАРІАНТИ ПОКРАЩЕННЯ	43
3.1. Опис роботи з програмним додатком та експлуатація	43
3.2. Тестування роботи програмного забезпечення.....	45
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ	56

ВСТУП

У світі, насиченому технологічними новинками, питання надання першої допомоги стає не тільки актуальним, але й невід'ємним елементом громадської безпеки. Однак у багатьох випадках, коли критичний момент вимагає швидкого та відповідального реагування, доступ до інтернету може бути обмеженим або відсутнім, ставлячи під загрозу ефективність системи надання допомоги.

Завдання роботи полягає у тому, щоб розробити програму, яка забезпечить користувачів необхідною інформацією та інструкціями щодо надання першої допомоги навіть коли інтернет-підключення відсутнє або обмежене. Це вимагає не тільки розробки ефективного програмного забезпечення, але і врахування медичних стандартів та психологічних аспектів ефективної комунікації в екстрених ситуаціях.

Кваліфікаційна робота вивчає технічні, медичні та соціальні аспекти розробки та впровадження такої програми. Аналізуючи потреби користувачів та враховуючи можливі ризики, ми прагнемо створити інноваційний інструмент, який забезпечить не лише навички надання першої допомоги, але й підтримку в реальному часі, не залежно від доступу до мережі.

Метою кваліфікаційної роботи є створення інноваційної програми першої допомоги, що функціонує в умовах відсутності інтернет-підключення, та її ретельний аналіз, валідація та оптимізація цього інструменту для досягнення найвищого рівня ефективності.

Завдання кваліфікаційної роботи включають в себе ретельний розгляд і вирішення ряду ключових завдань, спрямованих на досягнення визначених цілей:

1. Розробка програмного забезпечення: провести аналіз технічних можливостей та вимог для розробки програми першої допомоги, яка функціонує в умовах обмеженого чи відсутнього інтернет-підключення.

2. Валідація медичних аспектів: систематично переглянути та валідувати медичний зміст програми, забезпечуючи його відповідність найвищим медичним стандартам та рекомендаціям.

3. Психологічна адаптація GUI: розробити інтуїтивно зрозумілий та емпатичний інтерфейс, який підтримує користувачів у стресових ситуаціях та сприяє швидкій реакції.

4. Тестування ефективності: провести серію тестів та сценаріїв в реальних умовах для перевірки функціоналу та ефективності програми, звертаючи увагу на її реальний вплив у різних ситуаціях.

5. Документація та звітність: детально задокументувати всі етапи розробки, тестування та вдосконалення програми для створення звіту, який буде служити джерелом інформації для подальших досліджень та розвитку.

Об'єктом дослідження є програма першої допомоги, розроблена для функціонування в умовах відсутності інтернет-підключення. Предметом аналізу є технічні, медичні та соціальні аспекти цієї програми, а також її вплив на процеси надання екстреної допомоги та загальний рівень безпеки в громаді.

Науково-практична значимість цього дослідження полягає в тому, що воно спрямоване на розв'язання конкретних проблем, а також вносить вагомий внесок у різні сфери, включаючи науку, медицину та соціальні аспекти:

1. Технічний прогрес: розробка програми першої допомоги, яка працює в умовах відсутності інтернет-підключення, відкриває шлях до технологічних інновацій у сфері медичної допомоги. Це може послужити основою для подальших досліджень та розробок у галузі мобільних додатків та технологій збереження життя.

2. Медична допомога в умовах обмеженого зв'язку: програма має потенціал покращити доступність та якість першої допомоги в ситуаціях, коли інтернет-з'єднання недоступне, таких як природні катастрофи, аварії чи зони з обмеженим доступом.

3. Соціальна відповідальність: розробка програми, яка реагує на виклики екстрених ситуацій без інтернет-підключення, підсилює безпеку та готовність громади до дій у критичних обставинах, сприяючи загальному зростанню рівня безпеки та взаємодопомоги.

Питання кваліфікаційної роботи було досліджено та описано на базі розробки програмного додатку на мобільні девайси, який оперативно надає необхідну інформацію для забезпечення контролю та допомоги у критичних ситуаціях в умовах без підключення до інтернету.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1. Аналіз задачі, засобів та методів її вирішення

Метою кваліфікаційної роботи є розробка повноцінного мобільного додатка для забезпечення першої допомоги в умовах, коли інтернет-підключення відсутнє. Даний програмний продукт буде спрямований на надання необхідних знань та інструкцій для надання ефективної першої допомоги в екстрених ситуаціях.

У контексті розробки та впровадження Програми Першої Допомоги, незалежної від інтернет-підключення, ключовими вимогами та цільовими завданнями є:

Автономність, Достовірність та Відсутність Залежності від інтернету:

Вимога: забезпечити можливість користування програмою без наявності активного інтернет-з'єднання.

Цільове завдання: розробити та інтегрувати локальне сховище даних, що дозволить доступати до необхідної інформації навіть при відсутності мережі.

Ілюстративні приклади додатків, що можна використовувати у режимі без інтернет-з'єднання, включають Monobank, "Дія", Glovo, Liki24, Bolt, Uklon та інші [1]. Ці додатки можуть працювати, навіть коли відсутнє з'єднання з інтернетом, надаючи користувачам можливість ефективно використовувати їх у ситуаціях відсутності зв'язку із цивілізацією.

Інтуїтивний та Доступний Інтерфейс:

Вимога: забезпечити зручний та зрозумілий GUI для швидкого доступу до необхідних інструкцій та функцій [2].

Цільове завдання: розробити інтуїтивний GUI, спрощений для користувачів різного рівня освіти та навичок, з метою надання швидкого доступу до необхідних функцій і інструкцій.

Широке покриття сценаріїв критичних ситуацій:

Вимога: включити різноманітні сценарії та ситуації, щоб відповідати різним випадкам надання першої допомоги.

Цільове завдання: розробити базу даних інструкцій, яка охоплює різні теми, включаючи травми, захворювання та інші невідкладні становища, забезпечуючи повноту та актуальність інформації для вирішення різних ситуацій.

Можливість Оновлення та Розширення:

Вимога: забезпечити можливість оновлення та розширення програми для врахування нових стандартів та потреб користувачів.

Цільове завдання: розробити гнучку архітектуру, що дозволяє легко додавати нові функції та оновлювати інформаційний контент, забезпечуючи простоту та швидкість впровадження змін.

Важливість функцій у розробці мобільного додатка для забезпечення першої допомоги в умовах відсутності інтернет-підключення визначається його спроможністю надавати ефективну та безпечну допомогу у НС. Основні функції, які є важливими для успішної реалізації завдань, можуть включати:

Офлайн-доступ [3] до інструкцій з першої допомоги: Забезпечення можливості отримання докладних інструкцій щодо надання першої допомоги без необхідності підключення до інтернету. Це включає в себе інформацію про надання допомоги при різних видах НС.

Візуалізація інструкцій: додаток може включати графічні або відеоматеріали, які допомагатимуть користувачам краще зрозуміти та відтворити прийоми першої допомоги. Візуальні матеріали можуть бути корисними в стресових ситуаціях.

Працездатність в автономному режимі: забезпечення функціонування додатка без доступу до інтернету для того, щоб користувач міг скористатися ним у будь-який час і в будь-якому місці [4].

Керівництво крок за кроком: надання послідовних і простих кроків для кожного прийому першої допомоги з метою забезпечення правильного та швидкого реагування.

Для розробки мобільного додатка для забезпечення першої допомоги в умовах відсутності інтернет-підключення, можна використовувати різноманітні технічні засоби та інструменти. Ось кілька ключових компонентів:

Мова програмування: вибір мови програмування залежить від платформи. Для мобільних додатків часто використовують мови, такі як Swift для iOS (SwiftUI) або Kotlin/Java для Android [5].

Інтегроване середовище розробки (далі IDE): використання спеціалізованого IDE, такого як Xcode для iOS або Android Studio для Android, допомагає полегшити розробку та тестування додатків.

Сучасне програмне забезпечення можна умовно поділити на три основні види: веб-додатки та сайти, гібридна розробка програмного забезпечення та нативні додатки.

Веб-додатки та сайти є найпопулярнішими, а їхній розвиток дозволяє користувачам ефективно взаємодіяти в інтернеті, забезпечуючи широкий функціонал для мобільних пристроїв. Гібридна розробка використовує HTML, JAVA та CSS для створення додатків, які поєднують переваги нативних

інструментів з інноваційними технологіями. Прикладом є Facebook, який використовує гібридну модель з можливістю завантаження на мобільні пристрої, але вимагає доступу до інтернету.

Нативні додатки використовують повноцінний функціонал операційної системи, що робить їх найпродуктивнішими, але й вимогливими до ресурсів. Вони можуть легко взаємодіяти з операційною системою та надають високу швидкість роботи [6].

Графічні бібліотеки: використання бібліотек для роботи з графікою, таких як Canvas або OpenGL ES [7], для відображення візуальних матеріалів у додатку.

Хмарні сервіси (опційно): якщо додаток повинен взаємодіяти з хмарними сервісами, можна використовувати платформи, такі як Firebase або AWS Mobile.

Дизайн і UX-дослідження: використання інструментів для дизайну і прототипування, наприклад, Sketch або Figma, для створення GUI та покращення UX [8].

Обрано конкретні технології для розробки мобільного додатка для забезпечення першої допомоги в умовах відсутності інтернет-підключення з такими обґрунтуваннями:

1. Java для Android:

Широка підтримка та спільнота: Java є основною мовою програмування для Android, забезпечуючи широку підтримку та велику спільноту розробників. Це полегшить розробку та забезпечить доступність великої кількості ресурсів та бібліотек.

Надійність та стабільність: Java відома своєю стабільністю та надійністю, що є критичними якостями для мобільного додатка, призначеного для надання першої допомоги в екстрених ситуаціях.

2. Android Studio:

Офіційне середовище розробки: Android Studio є офіційним інтегрованим середовищем розробки для Android, розробленим компанією Google. Воно забезпечує повноцінні інструменти для розробки та тестування додатків, що сприяє ефективній розробці та підтримці.

Підтримка великої спільноти: Android Studio користується підтримкою великої спільноти розробників, що робить його популярним інструментом з багатьма документаційними ресурсами та розвитковою спільнотою.

3. Canvas:

Графічна можливість для інтерактивності: Canvas [9] дозволяє створювати графічні елементи, що може бути корисно для надання інтерактивних інструкцій та візуалізації процедур першої допомоги.

Локальне відображення графіки: Canvas ідеально підходить для локального відображення графіки, що дозволить додатку працювати ефективно в умовах відсутності інтернет-підключення.

Ці технології обрані через їхню відомість, стабільність, ефективність та можливість оптимального використання в умовах, коли додаток має функціонувати в автономному режимі та надавати користувачам необхідну допомогу в екстрених ситуаціях.

1.2. Ретроспектива мови програмування Java: аналіз особливостей та можливостей

Визначення мети ретроспективи є ключовим етапом, який допомагає чітко зорієнтувати учасників і визначити, що саме очікується від цього дослідження.

Мета ретроспективи мови програмування Java полягає в глибокому аналізі та оцінці усіх аспектів цієї мови програмування. Основні завдання включають вивчення ефективності її використання в різних сферах розробки, розгляд синтаксису та особливостей, які можуть впливати на розробку програм, оцінку зручності для розробників та вивчення інструментів, що використовуються. Також важливо виявити можливі проблеми та шляхи оптимізації використання мови, а також дослідити її роль у сучасній розробці програмного забезпечення та тенденції її розвитку. Підсумково, ретроспектива мови програмування Java має на меті надати повний обзор та зрозуміння цієї мови для кращого використання її потенціалу в майбутніх проектах.

Мова програмування Java була створена та розроблена компанією Sun Microsystems (згодом придбаною компанією Oracle) в кінці 1980-х та на початку 1990-х років. Основна ідея полягала в створенні мови, яка була б портативною (працювала на різних апаратних та операційних платформах), об'єктно-орієнтованою та безпечною [10].

Програмування на Java має свої особливості, які роблять цю мову популярною та ефективною для розробки різноманітних програмних застосунків. Ось аналіз основних особливостей програмування на Java:

1. Кросплатформенність (Write Once, Run Anywhere):

Кросплатформенність Java реалізується за допомогою віртуальної машини Java (JVM). JVM є виконавчим середовищем, яке трансформує байт-код Java, отриманий під час компіляції, в машинний код. Цей машинний код може бути

виконаний на різних платформах, що дозволяє програмам на Java працювати на різноманітних операційних системах без необхідності перекомпіляції.

2. Об'єктно-орієнтований Підхід:

Java побудована на основі об'єктно-орієнтованого підходу, що використовує класи та об'єкти для створення модульного та легко розширюваного коду. Цей підхід дозволяє ефективно організувати програмний код у вигляді невеликих, незалежних компонентів, що полегшує розвиток та підтримку програм. Об'єктно-орієнтована структура Java сприяє вищій зрозумілості, підтримці та розширенню кодової бази.

3. Інкапсуляція та Контроль Доступу:

Java підтримує інкапсуляцію, що дозволяє об'єднувати дані та методи в класах. Використовуючи ключові слова, такі як `'public'`, `'private'`, `'protected'`, можна ефективно контролювати доступ до різних частин класу. Це забезпечує захист від несанкціонованого доступу та сприяє створенню більш безпечних та підтримуваних програм.

4. Автоматичне Управління Пам'яттю:

Java використовує автоматичне управління пам'яттю за допомогою Garbage Collector, що полегшує розробникам відслідковування та вивільнення пам'яті. Garbage Collector автоматично визначає та видаляє об'єкти, які більше не використовуються в програмі, звільняючи розробників від необхідності явної роботи з вивільненням ресурсів. Це сприяє підвищенню продуктивності та забезпечує більшу стійкість програмного забезпечення.

5. Багатопотоковість [11]:

Мова Java володіє вбудованою підтримкою багатопотоковості, що дозволяє виконувати одночасно кілька задач. Ця можливість є корисною для підвищення

продуктивності програм та ефективної обробки паралельних завдань. Використання багатопоточності дозволяє програмам працювати швидше та ефективніше, особливо в умовах, коли необхідно вирішити кілька завдань одночасно..

6. Стандартні Бібліотеки:

Java має розгалужену набір стандартних бібліотек, які включають в себе різноманітні функції, такі як робота з мережами, обробка рядків, робота з файлами та інші. Це розширений інструментарій спрощує процес розробки, оскільки розробнику не обов'язково виготовляти всі компоненти "з нуля". Замість цього він може використовувати готові функції та класи з бібліотек, що полегшує вирішення різних завдань та прискорює розробку програм.

7. Широке Використання:

Java широко використовується для розробки різноманітних застосунків, включаючи корпоративні системи, веб-додатки, мобільні додатки, вбудовані програми та інші. Вона є універсальною мовою програмування, яка підходить для різних сценаріїв та областей застосування. Багатофункціональність Java дозволяє розробникам створювати різноманітні програми, що відповідають різним вимогам та потребам користувачів.

Ці особливості роблять Java потужним інструментом для розробки програмного забезпечення в різних областях.

1.3. Аналіз існуючого програмного забезпечення за тематикою роботи

Аналіз існуючого програмного забезпечення за тематикою роботи, що стосується розробки мобільного додатка для забезпечення першої допомоги в умовах відсутності інтернет-підключення, включає огляд конкурентного середовища та оцінку функціональності інших програмних продуктів.

Під час моніторингу Google Play було виявлено обмежену кількість додатків, що стосуються теми даної кваліфікаційної роботи. Були проведені аналіз позитивних та негативних аспектів цих додатків, головні сторінки яких було висвітлено у (Таблиці 1.1)

Перша допомога - IFRC - офіційний додаток IFRC "Перша допомога" забезпечує миттєвий доступ до інформації, необхідної для надання першої допомоги у найпоширеніших НС та надає поради з безпеки в кризових ситуаціях.

Завдяки інтерактивним вікторинам та простим покроковим сценаріям, вивчення навичок першої допомоги стає легким і доступним у повсякденному житті.

Одним із недоліків програми є те, що вона є в основному навчальним додатком [15]. В силу своєї освітньої спрямованості, використання програми не є ефективним та оперативним засобом надання допомоги в кризових ситуаціях, особливо в умовах відсутності доступу до інтернету.

Instant Aid - Erste Hilfe App - легкий та інтуїтивно зрозумілий додаток для надання першої допомоги допомагає вам ефективно діяти в стресовій ситуації, гарантуючи, що жоден важливий крок у процесі надання допомоги не буде пропущений. Важливо зазначити, що це безкоштовний застосунок [16].

Однак слід відзначити, що одним з недоліків застосунку є його обмежена мовна підтримка, яка обмежується англійською та німецькою мовами.

TacticMedAid - це довідник, який базується на адаптованій версії протоколу TCCC-AMS для військових та протоколу STOP THE BLEED для цивільних [17]. У цих довідниках розглядаються конкретні дії, які може виконати навіть особа без медичної освіти, і вони спрямовані на надання домашньої медичної допомоги.

Обмеження доступу до відео лише в режимі онлайн є суттєвим недоліком цього додатка, оскільки це ускладнює можливість перегляду без наявності інтернет-з'єднання. Крім того, слід відзначити, що додаток є недоробленим у відношенні обмеженої кількості інформації для громадянського населення, оскільки основний акцент зроблений на військовому напрямку. Це обмеження може створювати проблеми для користувачів, які шукають інформацію та інструкції, спрямовані на цивільний сектор.

Рис. 1.3. Існуюча програма першої допомоги. «TacticMedAid»

"iFirstAid" - це мобільний ресурс з безкоштовною інформацією щодо надання першої допомоги, який допоможе вам отримати знання та впевненість у вправному управлінні як невеликими, так і серйозними НС [18].

Головний недолік додатка полягає в обмеженні доступу лише до англійської мови, і відсутності можливості змінити мову локалізації на іншу.

Проаналізувавши існуюче програмне забезпечення за тематикою роботи, ми прийшли до висновку, що при створенні нового додатка необхідно враховувати наступне:

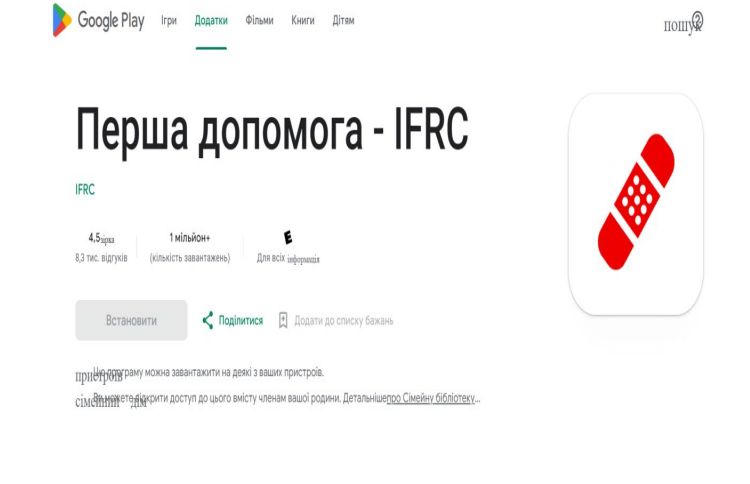
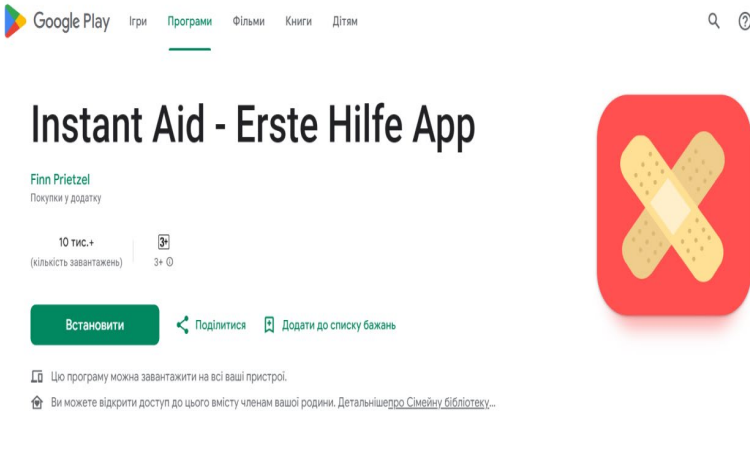
1. Відеоматеріали, які додаються до додатка, повинні працювати навіть у відсутності інтернет-з'єднання. Це забезпечить доступ до важливих відомостей у будь-який час і в будь-якому місці.

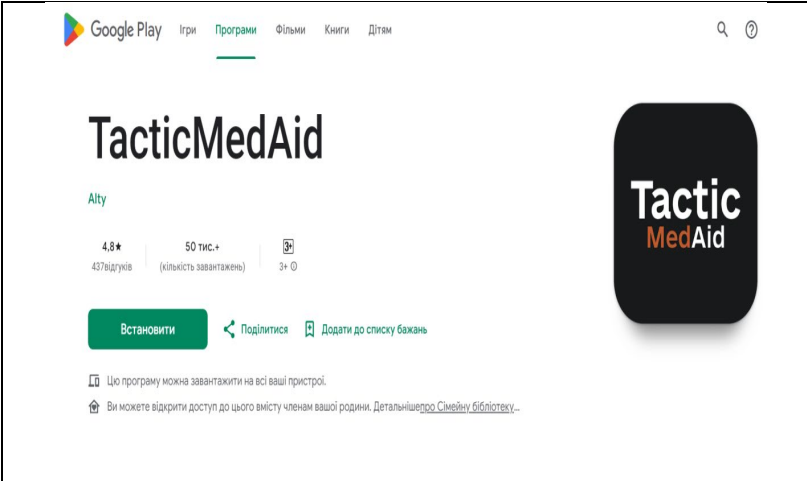
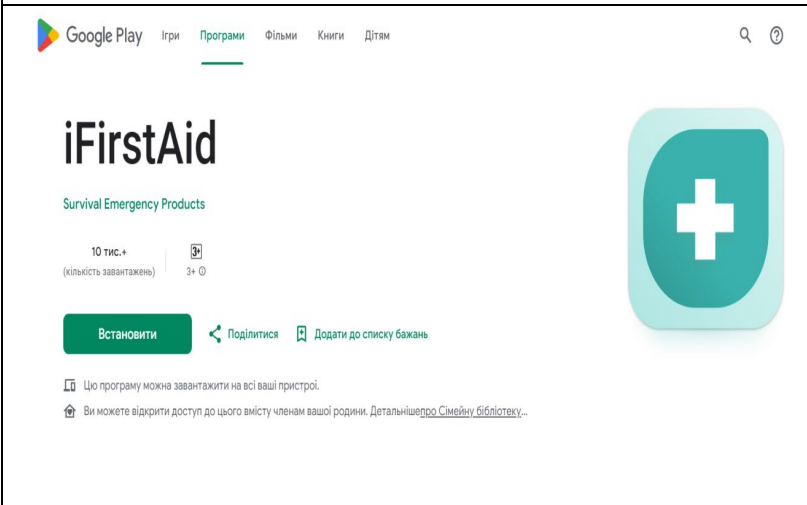
2. Основна мета додатка повинна полягати в наданні користувачам інструкцій та знань, спрямованих на вміння надавати допомогу в невідкладних ситуаціях, замість виключно навчання.

3. GUI додатка повинен бути зрозумілим для простого користувача, щоб надати можливість швидко та ефективно знаходити необхідні інструкції та функції.

Таблиця 1.1

Вже існуючі програмні застосунки першої допомоги

Головна сторінка додатку	Опис
	Існуюча програма першої допомоги. «Перша допомога - IFRC» - додаток спрямований на навчання матеріалу.
Продовження таблиці 1.1	
Головна сторінка додатку	Опис
	«Instant Aid - Erste Hilfe App» - додаток не підтримує українську мову та роботу без інтернет-підключення.

	Існуюча програма першої допомоги. «TacticMedAid» - фіксування контенту на військових тематиках.
	Існуюча програма першої допомоги. «iFirstAid» - додаток не підтримує жодної мови, крім англійської.

1.4. Огляд середовища розробки Android Studio

Android Studio є офіційним інтегрованим середовищем розробки (IDE) для платформи Android та абсолютно безкоштовним завдяки ліцензії Apache 2.0. Це кросплатформне рішення, підтримується на операційних системах Microsoft Windows, macOS та GNU/Linux. Розроблене на базі програмного забезпечення IntelliJ IDEA від компанії JetBrains, Android Studio було випущено з метою заміни Eclipse як офіційного інструменту розробки додатків для платформи Android [12].

Це комплексне середовище надає повний набір інструментів для розробки та налагодження програм для мобільної операційної системи Google.

IDE Android Studio надає розробникам інтуїтивно зрозумілий та потужний інструментарій для створення мобільних додатків під управлінням платформи Android. Головне робоче вікно має зручний GUI з меню та панелями інструментів, що полегшує навігацію та доступ до основних функцій [13].

Відкриття та створення проектів в Android Studio є невимушеним процесом. Розробники можуть легко відкривати існуючі проекти або створювати нові, налаштовуючи їх параметри за своїми потребами.

Навігація по проекту включає в себе швидкий доступ до файлів та папок, а також інструменти для ефективного пошуку та переміщення між частинами коду.

Відмінністю редактора коду в Android Studio є зручне оформлення, кольорова розмітка та вбудовані інструменти автодоповнення та визначення, що роблять процес написання та редагування коду простішим та ефективнішим для розробників.

Огляд редактора коду та його функціоналу в Android Studio надає розробникам повноцінне середовище для комфортної та продуктивної роботи при створенні Android-додатків.

Android Studio підтримує кілька мов програмування [14] для розробки мобільних додатків під управлінням платформи Android. Основні мови програмування, які використовуються в Android Studio, включають:

1. Kotlin:

Починаючи з випуску Android Studio 3.0, Kotlin стала офіційною мовою програмування для розробки Android-додатків. Забезпечуючи сучасні можливості та лаконічний синтаксис, Kotlin став популярним вибором серед молодих розробників для створення Android-застосунків.

2. C++:

Android Studio також підтримує розробку частини коду на мові програмування C++. Це особливо корисно для реалізації високопродуктивних операцій або використання існуючих бібліотек на C++. Розширена підтримка C++ в Android Studio дозволяє розробникам ефективно поєднувати можливості цієї мови з перевагами інших мов програмування, таких як Kotlin та Java.

3. Groovy:

Groovy може бути використаний як мова сценаріїв для деяких аспектів розробки Android-додатків. Його гнучкість та можливості використання у скриптах роблять його привабливим вибором для розробників, які шукають альтернативу в рамках розширених можливостей програмування на платформі Android..

4. XML:

Навіть як мова розмітки, XML використовується для опису GUI та ресурсів в Android Studio. Його зручність у визначенні структури GUI та відокремлення даних від коду додає ефективності у розробці Android-додатків.

5. Java:

Java є стандартною мовою програмування для розробки Android-додатків. Вона має обширну кількість ресурсів та бібліотек, спрямованих на полегшення процесу створення застосунків на велике різноманіття платформ (Android включно). Ця підтримка дозволяє розробникам вибирати мову програмування відповідно до їхніх потреб та вподобань. Kotlin набуває все більшої популярності як основна мова програмування для розробки Android-додатків, але Java залишається широко використовуваною.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Проектування загального алгоритму роботи програми

Після успішного освоєння теоретичних аспектів розробки програмного продукту ми можемо перейти до етапу проектування. Розглянувши загальний алгоритм функціонування програми [19], ми можемо визначити послідовність основних етапів та кроків, які вона виконує для досягнення своїх цілей, як можна бачити на схемі (Рис.2.1.)

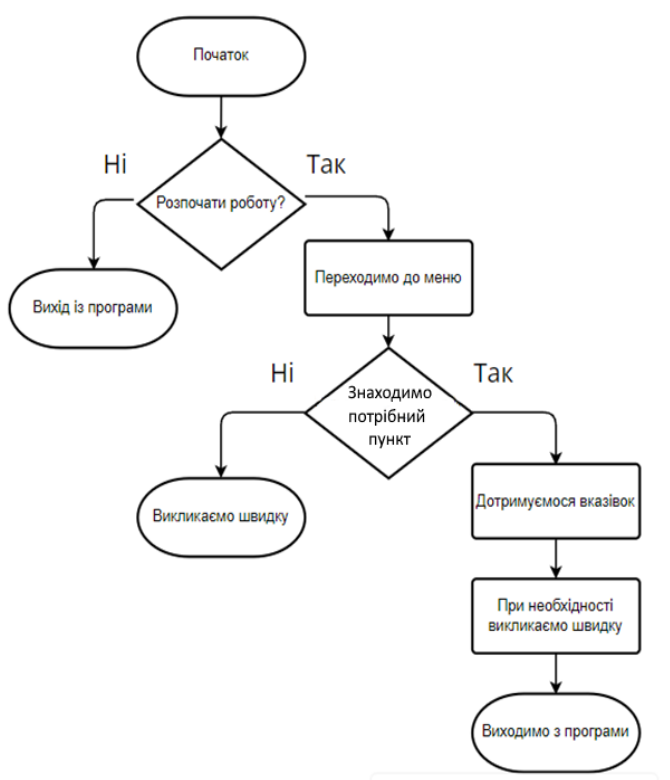


Рис. 2.1. Загальний алгоритм роботи програми

Цільова аудиторія мобільного додатка для надання першої допомоги включає людей без медичної освіти, які бажають отримати базові навички, а також професіоналів у галузі медицини, батьків, доглядальників та подорожуючих.

Вхідні дані для мобільного додатка [20] включають інформацію про тип невідкладного випадку або медичної ситуації, а також характеристики потерпілого (якщо відомо), такі як вік, стать, можливі травми чи симптоми.

Вихідні дані додатка полягатимуть у наданні послідовних інструкцій щодо надання першої допомоги в конкретній ситуації. Це може включати крок за кроком дії, візуалізацію або відео-матеріали, щоб допомогти користувачеві правильно та ефективно реагувати на невідкладну ситуацію.

2.2. Розробка програмного забезпечення

При розробці програмного забезпечення необхідно враховувати як візуальні, так і функціональні аспекти. Візуальна складова програми охоплює основні сторінки додатку, такі як головна, друга сторінка із списком, та сторінки з основною інформацією. Дизайн програми відзначається мінімалістичним підходом [21].

Головна сторінка, що зображена на рис. 2.2, спроектована з метою максимальної доступності та зрозумілості для будь-якого користувача. В ньому розташовані дві ключові кнопки: "Увійти" (Розпочати роботу) та "Вийти". Ця мінімалістична структура головної сторінки дозволяє користувачеві легко орієнтуватися та вибирати бажані опції, підкреслюючи простоту використання програми.

Для дизайну головного екрану була використана не комерційна 3D-модель лікаря [22], що у найкращий спосіб передає суть додатка. Вибраний шрифт є простим і лаконічним, додаючи загальну зрозумілість та чіткість GUI. Основна кнопка виконана у вигляді аптечного набору, підкреслюючи його призначення, а кнопка виходу відображена як двері, для більшої зрозумілості та зручності користувача.

Вибрані кольори сприяють заспокоєнню та не викликають відчуття паніки [23], додаючи важливий аспект емоційної стабільності до вигляду програми. Пастельні та теплі відтінки використовуються для створення приємного візуального UX, а їх комбінація підтримує сприятливий фон для користувача.

Основою на цих принципах, нами було створено програмний код (Додаток А.1) головної сторінки меню та її розмітку, або верстку (Додаток А.2)

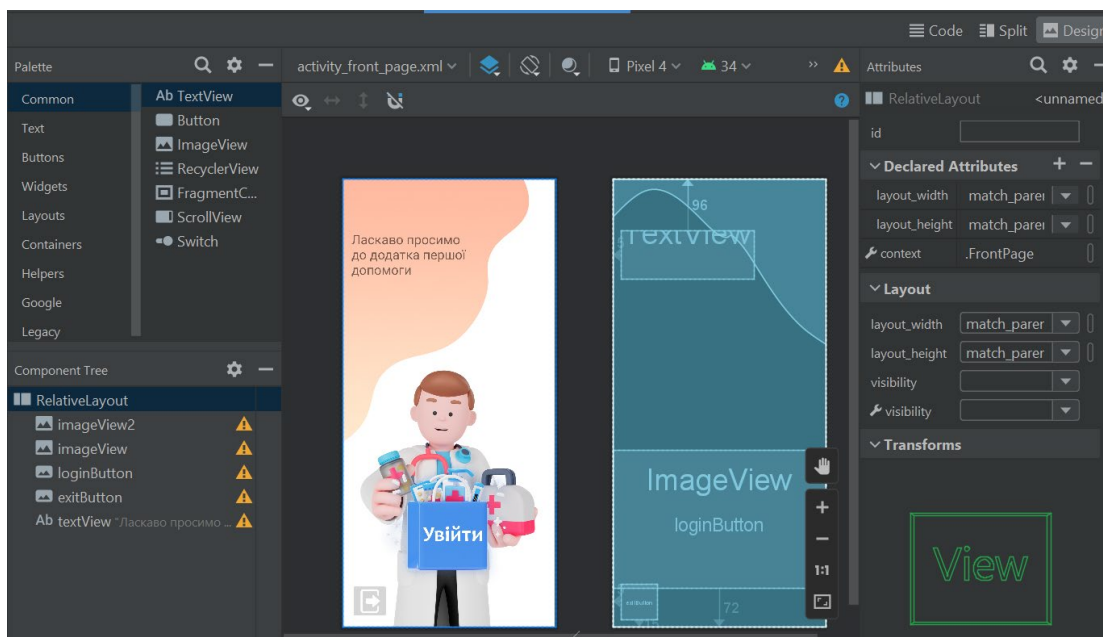


Рис. 2.2. GUI сторінки, який є зручним для користувача

Розглянемо тепер код та функціональну частину головного екрану додатка, що включає в себе створення ключових алгоритмів та взаємодії між користувачем і програмою з урахуванням особливостей Android Studio [24], поадну у вигляді таблиці 2.1;

Таблиця 2.1.

Основний функціонал головного екрану додатка

Код	Опис
<pre> ImageButton exitButton = findViewById(R.id.e xitButton); ImageButton loginButton = findViewById(R.id.l oginButton); </pre>	У даному коді здійснюється пошук двох елементів GUI за їх ідентифікаторами: ImageButton [25] exitButton = findViewById(R.id.exitButton);: Знаходить ImageButton за ідентифікатором R.id.exitButton і прив'язує його до змінної exitButton. ImageButton loginButton = findViewById(R.id.loginButton);: Знаходить ще один ImageButton за ідентифікатором R.id.loginButton і прив'язує його до змінної loginButton.
<pre> ImageButton exitButton = findViewById(R.id.e xitButton); ImageButton loginButton = findViewById(R.id.l oginButton); </pre>	У даному коді здійснюється пошук двох елементів GUI за їх ідентифікаторами: ImageButton [25] exitButton = findViewById(R.id.exitButton);: Знаходить ImageButton за ідентифікатором R.id.exitButton і прив'язує його до змінної exitButton. ImageButton loginButton = findViewById(R.id.loginButton);: Знаходить ще один ImageButton за ідентифікатором R.id.loginButton і прив'язує його до змінної loginButton.

Продовження таблиці 2.1

Код	Опис
<pre> exitButton.setOn ClickListener(new View.OnClickListener() { @Override public void onClick(View view) { runOnUiThread(new Runnable() { @Override public void run() { finishAffinity(); } }); } }); </pre>	У даному коді встановлюється обробник кліку для кнопки `exitButton`. При натисканні на цю кнопку викликається метод `onClick` [26], який містить логіку для виходу з додатка. Детальний опис кожного рядка: 1. `exitButton.setOnClickListener (new View.OnClickListener() {...}`: Встановлює обробник подій для кнопки `exitButton`. 2. `public void onClick(View view) {...}`: Перевизначає метод `onClick` інтерфейсу `View.OnClickListener`. 3. `runOnUiThread(new Runnable() {...}`: Запускає виконання коду на основному потоці GUI. 4. `public void run() {...}`: Перевизначає метод `run` інтерфейсу `Runnable`. У цьому методі міститься код логіки для виходу [27]. 5. `finishAffinity();`: Метод `finishAffinity()` завершує поточну активність та всі активності, пов'язані з нею, і виходить з додатка. Отже, цей код забезпечує вихід з додатка при натисканні на кнопку `exitButton`.

Продовження таблиці 2.1

Код	Опис
<pre>loginButton.setOnClickListner(new View.OnClickListener() { @Override public void onClick(View view) { Intent intent = new Intent(FrontPage.this, MenuPage.class); startActivity(intent); } });</pre>	В даному коді встановлюється обробник кліку для кнопки loginButton. При натисканні на цю кнопку викликається метод onClick, який містить логіку для переходу на іншу активність. Детальний опис кожного рядка: loginButton.setOnClickListener(new View.OnClickListener() {...}): Встановлює обробник подій для кнопки loginButton. public void onClick(View view) {...}: Перевизначає метод onClick інтерфейсу View.OnClickListener. Весь код логіки для переходу на іншу активність знаходиться в середині цього методу. Intent intent = new Intent(FrontPage.this, MenuPage.class);: Створює новий об'єкт Intent, який вказує на перехід з поточної активності (FrontPage.this) до активності MenuPage.class. startActivity(intent);: Запускає нову активність, яку вказано у Intent.

На другій сторінці додатка розташовано список кнопок (рис.2.3). У верхній панелі розташовані чотири кнопки для керування додатком: перехід на початковий екран, перехід до сторінки із інформацією про аптечку та кнопки для зміни розміру шрифту, було враховано можливість проблем із зором користувача, дозволяючи змінювати розмір шрифту для покращення читабельності. (Додаток Б.1), (Додаток Б.2)

Далі наведено список кнопок із назвами тем і можливістю переходу до сторінки із відповідною інформацією та код, відповідаючий а керування даною сторінкою додатку на таблиці 2.2.

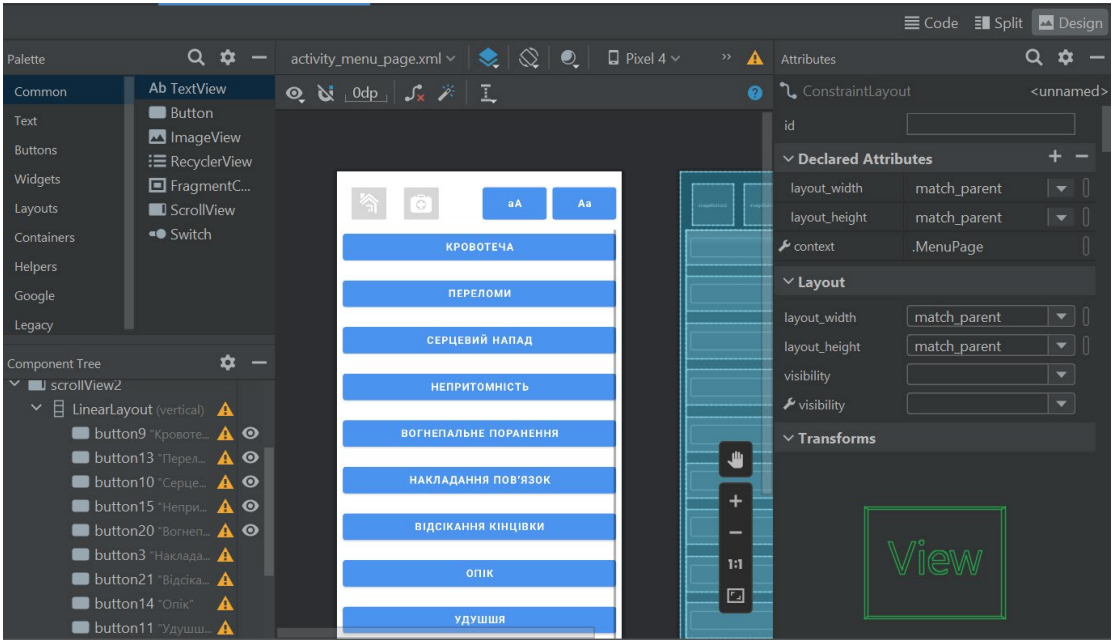


Рис. 2.3. Друга сторінка додатку зі списком розділів та кнопками налаштування

Таблиця 2.2.

Опис основного функціоналу другої сторінки додатка

Код	Опис
<pre>button3 = findViewById(R.id.button3); button9 = findViewById(R.id.button9); button10 = findViewById(R.id.button10); button11 = findViewById(R.id.button11); button12 = findViewById(R.id.button12); button13 = findViewById(R.id.button13); та інші.</pre>	Цей код здійснює пошук кнопок в GUI Android-додатка [28] за їх ідентифікаторами та прив'язує їх до змінних для подальшого використання в коді. Вибірка ідентифікаторів кнопок виглядає так: R.id.button3, R.id.button9, і так далі.

Продовження таблиці 2.2.

Код	Опис
<pre> button3.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent intent = new Intent(MenuPage.this, ApplyingBandages.class); startActivity(intent); } }); </pre>	Цей код встановлює обробник кліку для кнопки button3 на сторінці меню вашого Android-додатка. При натисканні на цю кнопку викликається метод onClick, який включає логіку для переходу на іншу активність. Цей метод повторюється для кожної кнопки.
<pre> private void setTextSize(int textSize) { Button[] buttons = { button3, button9, button10, button11, button12, button13, button14, button15, button16, button17, button18, button20, button21, button22, button23 }; for (Button button : buttons) { button.setTextSize(textSize); } } </pre>	Цей код визначає приватний метод setTextSize, який встановлює розмір тексту для групи кнопок у Android-додатку. Кількість кнопок та самі кнопки перераховуються у масиві buttons. Цей метод дозволяє однаково змінити розмір тексту для групи зазначених кнопок.

Наступною сторінкою є розділ із зображенням аптечки та кількома кнопками, зокрема "Назад в меню" та "Повернутися до попереднього розміру зображення" із головним інформаційним зображенням відображеним на повний екран для зручності (рис. 2.4). Нами було написано код для обробки жестів на зображеннях для інформаційних сторінок (Додаток В.1), тож користувач може зручно масштабувати та переміщувати зображення за своїм власним бажанням, щоб полегшити його сприйняття. Зображення [29], включає в себе перелік необхідних предметів для аптечки.

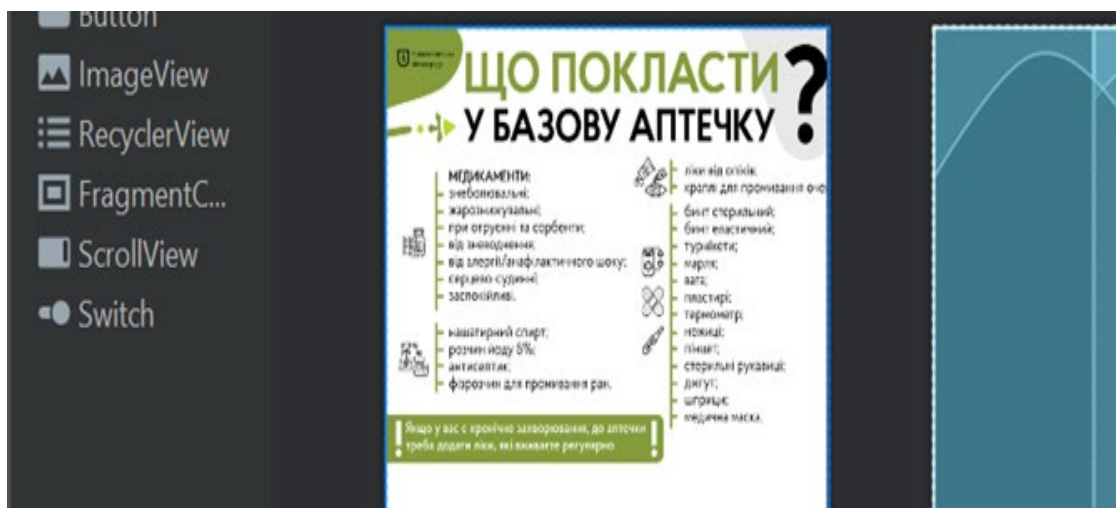


Рис. 2.4. Сторінка аптечки

Код даної сторінки представлено та пояснено нижче у таблиці 2.3.

Таблиця 2.3.

Опис основного функціоналу сторінки аптечки

Код	Опис
<pre> @Override protected void onCreate(@Nullable Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_image_page); backButton = findViewById(R.id.imageButton); resetButton = findViewById(R.id.resetButton); imageView = findViewById(R.id.imageView); scaleGestureDetector = new ScaleGestureDetector(this, new ScaleListener()); backButton.setOnClickListener(view -> finish()); resetButton.setOnClickListener(view -> resetImage()); imageView.setOnTouchListener((v, event) -> { scaleGestureDetector.onTouchEvent(event); handleMoveEvent(event); return true; }); } </pre>	Цей код встановлює функціональність для сторінки із зображенням, включаючи кнопки "Назад" і "Скинути", а також підключає метод для можливості масштабування та переміщення зображення за допомогою жестів.

Продовження таблиці 2.3.

<pre> private void resetImage() { scaleFactor = 1.0f; imageView.setScaleX(scaleFactor); imageView.setScaleY(scaleFactor); imageView.setX(0); imageView.setY(0); } </pre>	Цей метод скидає розмір та позицію [30] зображення на початкові значення.
<pre> private class ScaleListener extends ScaleGestureDetector.SimpleOnScale GestureListener { @Override public boolean onScale(ScaleGestureDetector detector) { scaleFactor *= detector.getScaleFactor(); scaleFactor = Math.max(0.5f, Math.min(scaleFactor, 2.0f)); imageView.setScaleX(scaleFactor); imageView.setScaleY(scaleFactor); return true; } } </pre>	Цей код визначає приватний клас `ScaleListener`, який розширює `ScaleGestureDetector.SimpleOnScaleGestureListener`. У методі `onScale` обробляється жест масштабування (як зумування на сенсорному екрані). Код використовує `detector.getScaleFactor()` для визначення зміни масштабу та обмежує його значення в діапазоні від 0.5 до 2.0. Потім цей масштабний коефіцієнт застосовується до зображення через встановлення масштабу по осях X та Y. Цей код відповідає за обробку жестів масштабування для забезпечення зручного змінення розміру зображення на сторінці [31].

Продовження таблиці 2.3.

<pre> private void handleMoveEvent(MotionEvent event) { switch (event.getAction()) { case MotionEvent.ACTION_DOWN: previousX = event.getX(); previousY = event.getY(); break; case MotionEvent.ACTION_MOVE: float dx = event.getX() - previousX; float dy = event.getY() - previousY; imageView.setX(imageView.getX() + dx); imageView.setY(imageView.getY() + dy); previousX = event.getX(); previousY = event.getY(); break; } } </pre>	Цей метод визначає обробник подій для переміщення зображення на сторінці залежно від руху пальця користувача на сенсорному екрані. 1. У випадку <code>MotionEvent.ACTION_DOWN</code> (початок касання), фіксуються початкові координати. 2. У випадку <code>MotionEvent.ACTION_MOVE</code> (рух пальця), визначається зміщення (<code>dx</code> і <code>dy</code>) відносно попередніх координат, і зображення переміщується на відповідну дельту. 3. По завершенні події, оновлюються початкові координати для наступного кроку. Цей код забезпечує можливість переміщати зображення на сторінці відповідно до руху пальця користувача.
---	---

Сторінки з текстовою інформацією створені відповідно до єдиного принципу, використовуючи дані з офіційних медичних статей [32]. Кожна сторінка цього ресурсу спрямована на надання конкретних термінів та практичних вказівок для дій під час НС.

Основний акцент зроблено на важливих термінах, які стосуються НС та заходів, які слід вживати для ефективної реакції та допомоги.

Кожна сторінка обладнана кнопками, що забезпечують зручний UX. Кнопка "Назад" дозволяє повертатися до попередньої сторінки, та кнопки для зміни розміру шрифту, усі вони реалізовані за допомогою Java [33]. Це забезпечує можливість налаштовувати зручний розмір шрифту відповідно до потреб користувача – як показано на рис. 2.5.

Основаючись на цих принципах, нами було створено програмний код (Додаток Д.1) інформаційної сторінки про кровотечу, та розмітку, або верстку (Додаток Д.2)

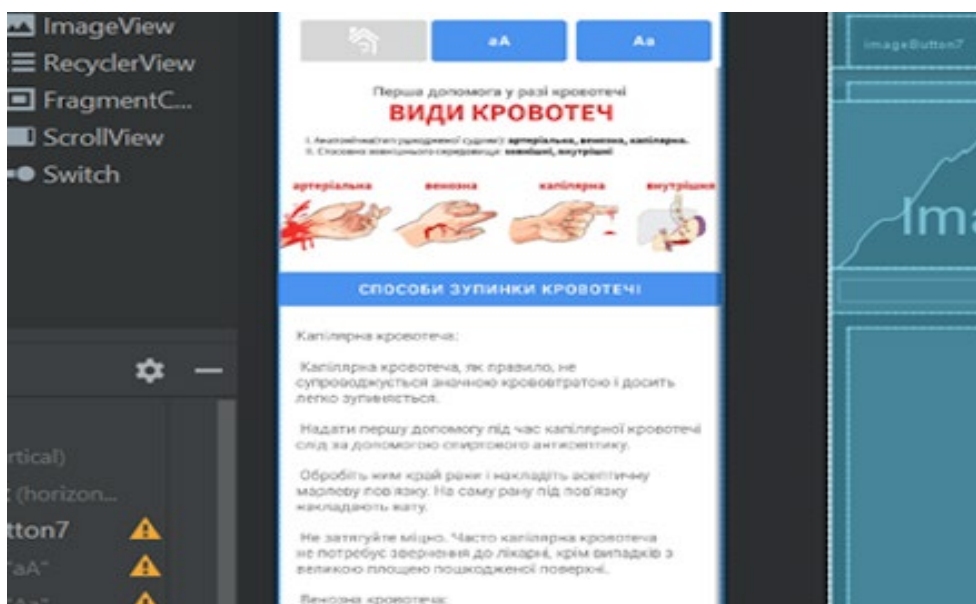


Рис. 2.5. Сторінка присвячена першій допомозі при кровотечі

Код даної сторінки представлено та описано нижче на таблиці 2.4.

Таблиця 2.4.

Опис основного функціоналу сторінок з інформацією

Код	Опис
<pre> ImageButtons imageButtons = findViewById(R.id.imageButtons); Button button7 = findViewById(R.id.button7); Button button4 = findViewById(R.id.button4); Button button5 = findViewById(R.id.button5); textView = findViewById(R.id.textView); textView3 = findViewById(R.id.textView3); button4.setOnClickListener(new View.OnClickListener() { public void onClick(View v) { textView3.setTextSize(28); textView.setTextSize(32); } }); button5.setOnClickListener(new View.OnClickListener() { public void onClick(View v) { textView3.setTextSize(14); textView.setTextSize(12); } }); </pre>	Код встановлює обробники подій для кнопок `button4` і `button5`. При натисканні на `button4` текстові поля `textView` і `textView3` отримують величину шрифту 32 та 28 відповідно. При натисканні на `button5` вони отримують величину шрифту 12 та 14. Цей код виконує зміну розміру шрифту для двох текстових полів (`textView` і `textView3`) в залежності від того, яка кнопка була натиснута.

Продовження таблиці 2.4	
Код	Опис
<pre> imageButton3.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent intent = new Intent(Bleeding.this, MenuPage.class); startActivity(intent); } }); </pre>	Цей код встановлює обробник подій для `ImageButton` [34] з ідентифікатором `imageButton3`. При натисканні на цей `ImageButton` відбувається перехід до активності `MenuPage` в додатку Android.

Додаток також пропонує для користувачів функцію – сторінку з відеоматеріалами (Рис.2.6).

Важливим аспектом є естетичний та легкий у використанні GUI сторінки відео-матеріалів. Необхідною характеристикою є зручність навігації, що дозволяє користувачу швидко знаходити необхідний вміст, коли необхідно. Додатково, сторінка відображає інформацію про те, як правильно та швидко надавати першу допомогу в різних ситуаціях – доступ до такої інформації повинен бути швидким та зручним.

Загальна концепція сторінки відео-матеріалів відображає важливість забезпечення доступу до надійної медичної інформації у НС, коли доступ до інтернету може бути обмеженим.

Дана сторінка надає надійний доступ до локальних інформаційних відеофайлів навіть під час відсутності підключення до мережі.

Основоючись на цих принципах, нами було створено програмний код (Додаток Е.1) інформаційної сторінки про непритомність, та розмітку, або верстку (Додаток Е.2)

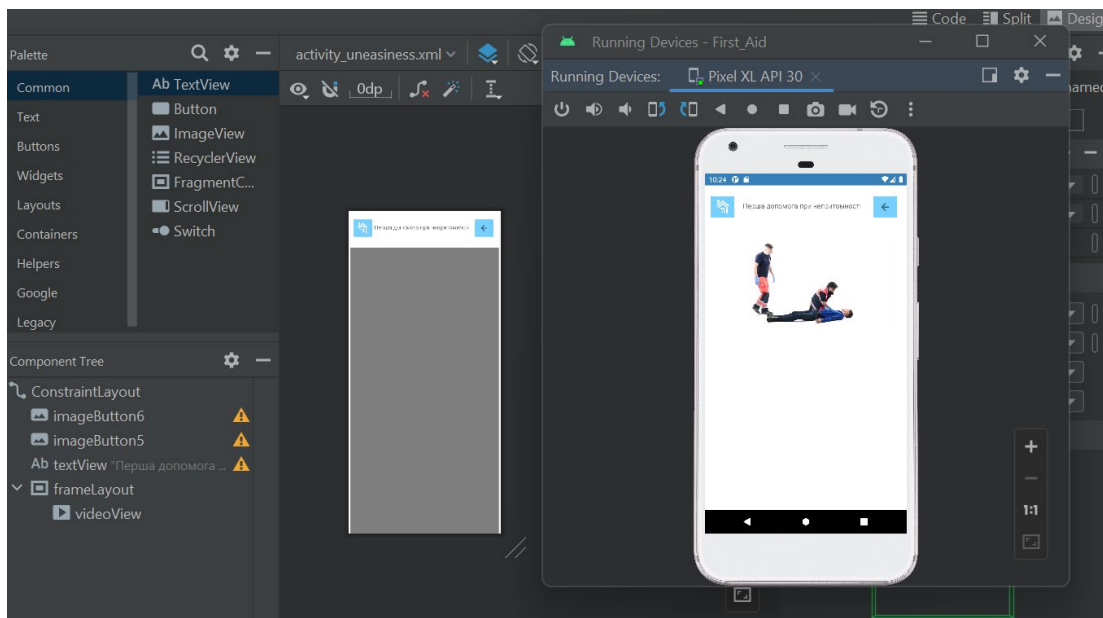


Рис. 2.6. Проста та естетична сторінка для відеоматеріалів

Код для сторінки із відеоматеріалами подаємо та описуємо нижче у таблиці 2.5.

Таблиця 2.5.

Опис основного функціоналу сторінок з відеоматеріалами

Код	Опис
<pre> @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_uneasiness); VideoView videoView = (VideoView) findViewById(R.id.videoView); String videoPath = "android.resource://" + getPackageName() + "/" + R.raw.v1; uri = Uri.parse(videoPath); videoView.setVideoURI(uri); MediaController mediaController = new MediaController(this); videoView.setMediaController(mediaController); mediaController.setAnchorView(videoView); videoView.start(); if (savedInstanceState != null){ uri = savedInstanceState.getParcelable(KEY_VIDEO); } if (uri!=null){ videoView.setVideoURI(uri); } } </pre>	У методі `onCreate` цей код встановлює контент активності на основі макету `activity_uneasiness`. Далі, він ініціалізує `VideoView` [35] та встановлює шлях до відео, яке розташоване в ресурсах додатка. Створюється об'єкт `MediaController` [36] для керування відтворенням відео, і він асоціюється з `VideoView`. Відео розпочинає відтворення автоматично. Також, код перевіряє наявність збереженого стану при повороті екрану. Якщо він не порожній, то відновлюється URI відео. Це забезпечує плавний перехід між станами активності при різних подіях, таких як поворот екрану.

Продовження таблиці 2.5.

Код	Опис
<pre> @Override protected void onSaveInstanceState(@NonNull Bundle outState) { outState.putParcelable(KEY_VIDEO, uri); super.onSaveInstanceState(outState); } public void startActivitiHome(View v){ Intent intent = new Intent(this,FrontPage.class); startActivity(intent); } public void startActivitiHome2(View v){ Intent intent = new Intent(this,MenuPage.class); startActivity(intent); } </pre>	Цей код визначає два методи в активності Android: onSaveInstanceState: Перевизначає метод для збереження стану активності під час її знищення (наприклад, при повороті екрану). В даному випадку, URI відео зберігається у бандлі для подальшого відновлення. startActivitiHome: Запускає нову активність (FrontPage) за допомогою інтента при натисканні на відповідний елемент GUI. startActivitiHome2: Запускає іншу активність (MenuPage) за допомогою інтента при натисканні на інший елемент GUI.

Для поліпшення функціональності додатка та забезпечення більшої зручності користувачу, у процесі розробки додатка нами була внесена важлива

інновація - тепер на головному екрані доступна спеціальна кнопка для негайного виклику екстреної допомоги.

Ця кнопка (Рис.2.7.) призначена для виклику служб оперативного реагування у випадках НС та надає користувачам можливість швидкого доступу до допомоги та підтримки. Після натискання на кнопку, додаток автоматично ініціює виклик екстреним службам.

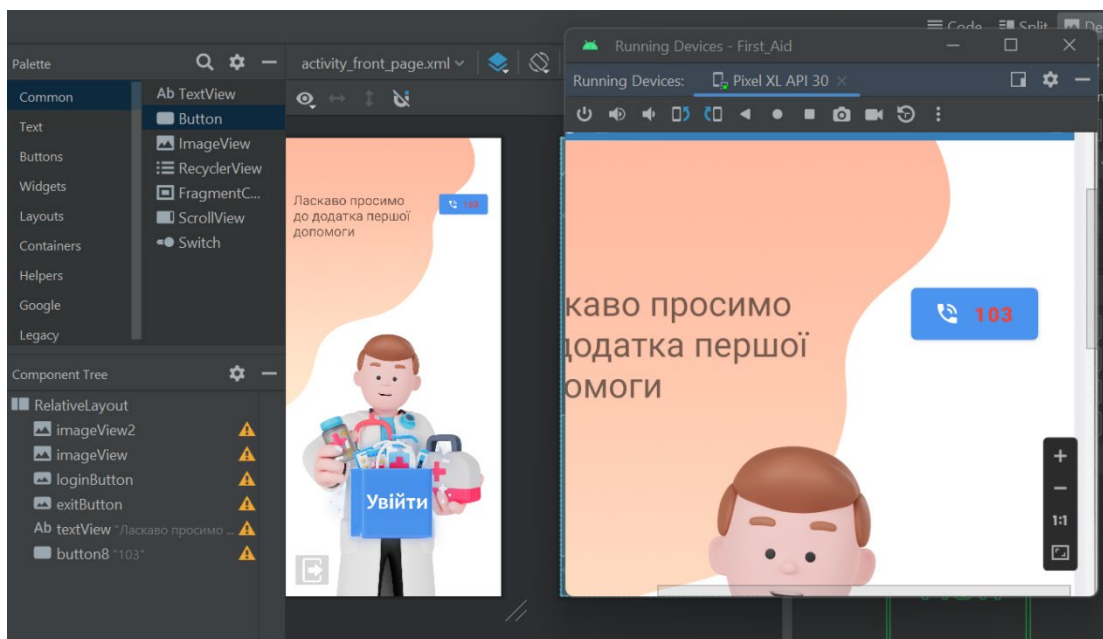


Рис. 2.7. Удосконалення стартової сторінки програми (у текст:кнопкою екстреної допомоги)

Удосконалюючи функціонал додатка, було вжито ключовий крок - додано дозвіл у маніфест-файл [37]. Шляхом включення наступного рядка в блок <manifest>, надається можливість додатку ініціювати телефонні виклики:

```
<uses-permission android:name = "android.permission. CALL_PHONE" />
```

Крім того, для повноцінної реалізації функціоналу, було введено обробник подій кнопки. Цей обробник не лише активується при натисканні на кнопку, а й відповідає за відправлення виклику до екстреної служби. Код, відповідальний за цю роботу, подано та описано нижче на таблиці 2.6.

Таблиця 2.6.

Функціонал обробника подій кнопки виклику до екстреної служби

Код	Опис
<pre> findViewById(R.id.button8).setOnClickListener { @Override public void onClick(View v) { // Виклик екстрених служб String phoneNumber = "103"; Intent dialIntent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" + phoneNumber)); // Перевірка дозволу на дзвінок if (checkSelfPermission(android.Manifest.permission.CALL_PHONE) == android.content.pm.PackageManager.PERMISSION_GRANTED) { startActivity(dialIntent); } else { // Запит дозволу на дзвінок requestPermissions(new String[]{android.Manifest.permission.CALL_PHONE}, 1); } } }); </pre>	Цей код встановлює обробник подій для кнопки з ідентифікатором `button8`. При клацанні на цю кнопку викликається екстрена служба ("103"). Код перевіряє, чи є дозвіл на виклик телефону. Якщо дозвіл наданий (`PERMISSION_GRANTED` [38]), то відбувається ініціація дзвінка за допомогою `Intent.ACTION_CALL`. У протилежному випадку ініціюється запит на отримання дозволу на дзвінок за допомогою `requestPermissions`.

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ, ЙОГО ТЕСТУВАННЯ ТА ВАРІАНТИ ПОКРАЩЕННЯ

3.1. Опис роботи з програмним додатком та експлуатація

Після розробки програмного коду (див. додатки) нашої програми першої допомоги та його компіляції за допомогою Android Studio, ми отримали функціонуючий додаток на платформу Android, який можемо інсталювати на мобільний девайс та користуватись.

Після запуску програмного додатку, користувач отримує доступ до головного меню програми, де можна знайти такі опції, як «Увійти», «Дзвінок у 103» та «Вихід» (рис. 3.1).

GUI програми [39] грає ключову роль у створенні зручного та ефективного способу взаємодії з користувачем, надаючи йому можливість ефективно використовувати різноманітні функції додатку. Головне меню стає візуальною точкою входу, де користувач може обирати необхідні опції та виконувати відповідні дії взаємодіючи з інтерактивними елементами GUI.

Натискання на кнопку "Увійти" дозволяє користувачеві переглядати список усіх доступних тем з інформацією і напрямку переходити до їх перегляду для зручності навігації. (рис. 3.2.).

При виборі однієї з тем і натисканні відповідної кнопки користувач переходить на окрему сторінку, де відображається відповідна інформація, як зображено на (рис.3.3.).

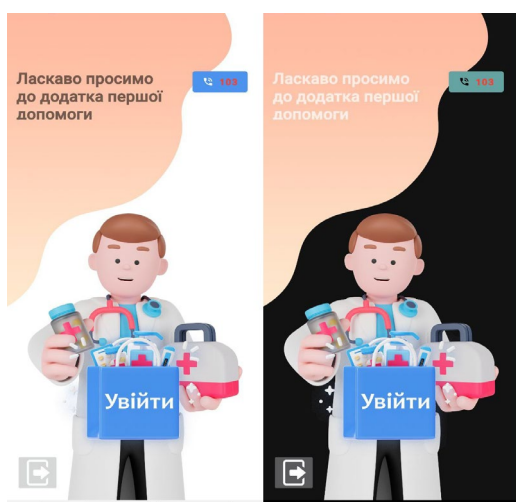


Рис. 3.1. Зручне головне меню програми

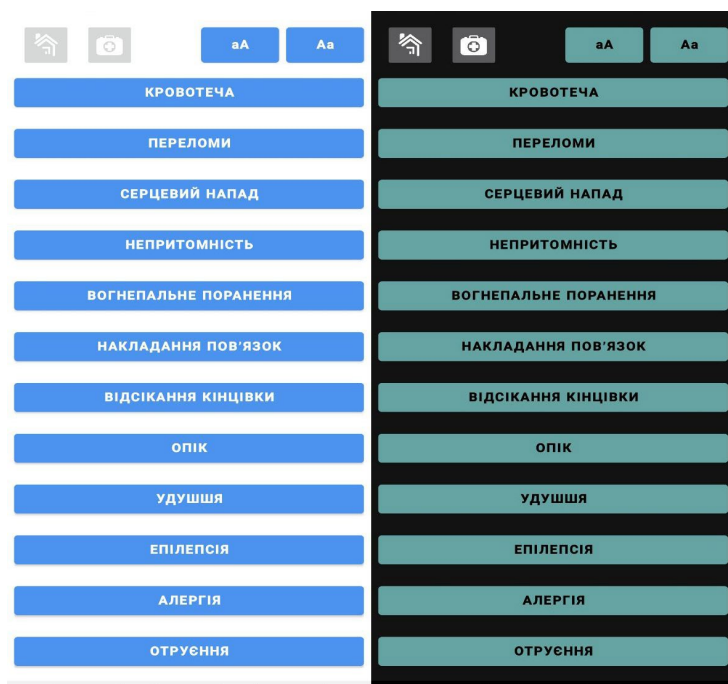


Рис. 3.2. Мінімалістична та зручна сторінка зі списком тем.

При розробці GUI ми слідуємо принципам мінімалізму та намагаємося досягнути приємного UX при використанні.

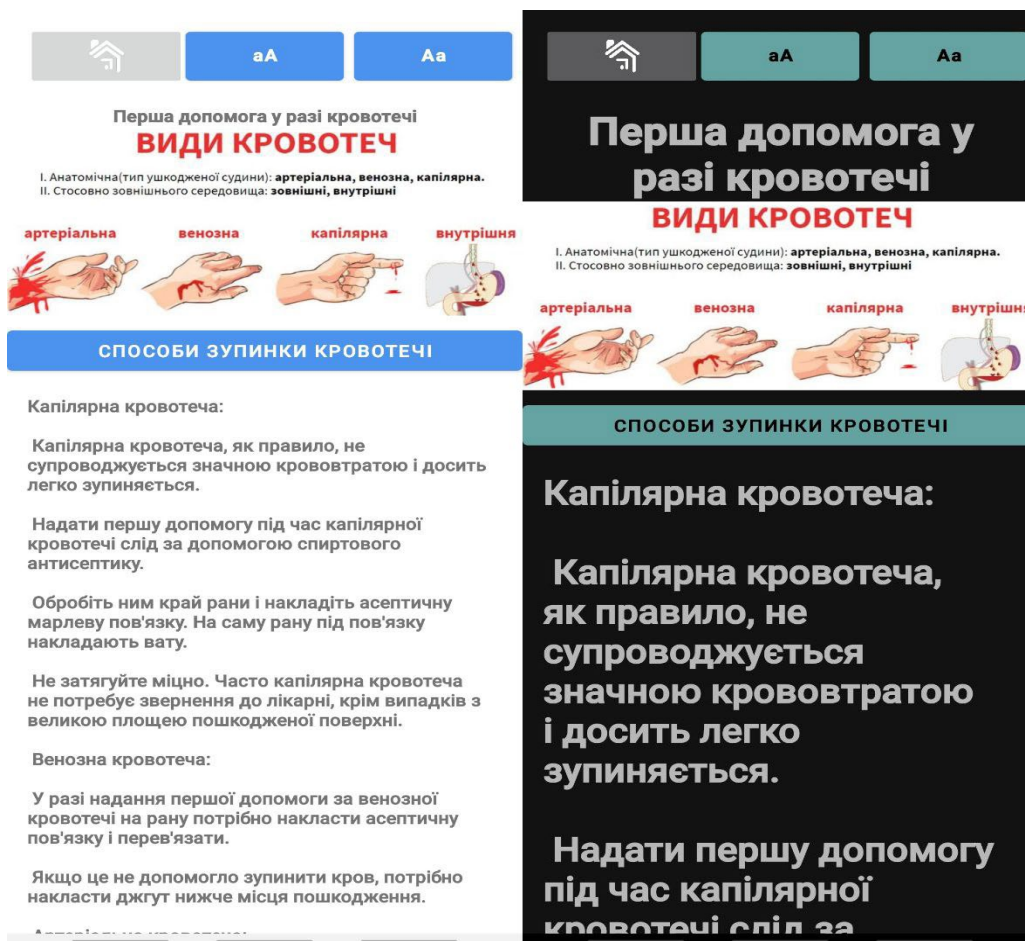


Рис. 3.3. Просто налаштовувана під користувача сторінка із інформацією.

3.2. Тестування роботи програмного забезпечення

Впровадження тестування додатку [40] "Розробка та впровадження програми першої допомоги" є критичним етапом у розробці, спрямованим на забезпечення високої якості та відповідності стандартам. Цей процес включає в себе проведення різноманітних тестів, спрямованих на перевірку функціональності, ефективності та стабільності додатку. Нижче подано опис функціоналу у тесті: (переробити у таблицю)

Тест №1. Встановлення програми:

Копіювання програми разом із її супутніми файлами, активація програми та виконання тестування дали позитивний результат, при цьому жодних помилок

не виявлено. Цей успішний виконавчий процес підтверджує, що програма інтегрується належним чином, а її функціональність відповідає очікуванням.

Тест №2. Взаємодія з картинкою:

Демонстрація виконання цього завдання недоступна, однак можна представити фрагмент програмного коду, який втілює в собі вказану задачу. Більш детальну інформацію щодо коду можна знайти в таблиці 2.3.

Тест №3. Виклик 103:

Під час натискання на кнопку "103" програма запитує дозвіл на виклик. Згідно з результатами тестування підтверджено, що після підтвердження дозволу здійснюється дзвінок до служби 103.

Тест №4. Поворот екрану:

Під час перевірки виявлено, що під час повороту екрану з вертикального (рис.3.4.) в горизонтальний режим (рис.3.5.) основні елементи залишаються незмінними, інформація подається так само зручно, як і в вертикальному режимі.

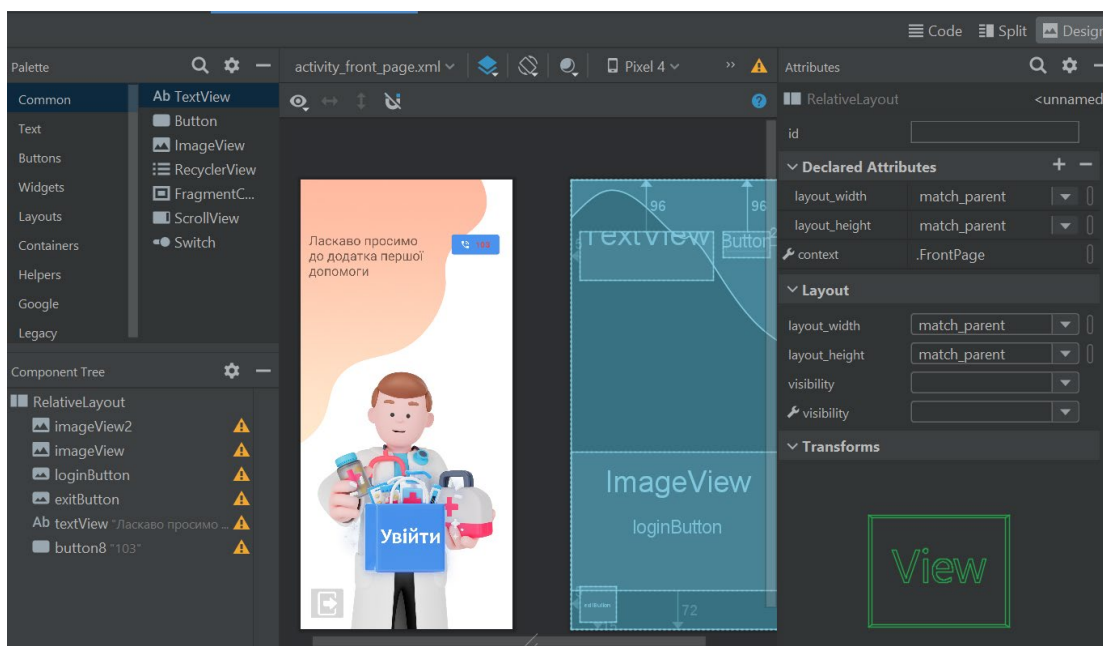


Рис. 3.4. Робота додатка у вертикальному режимі

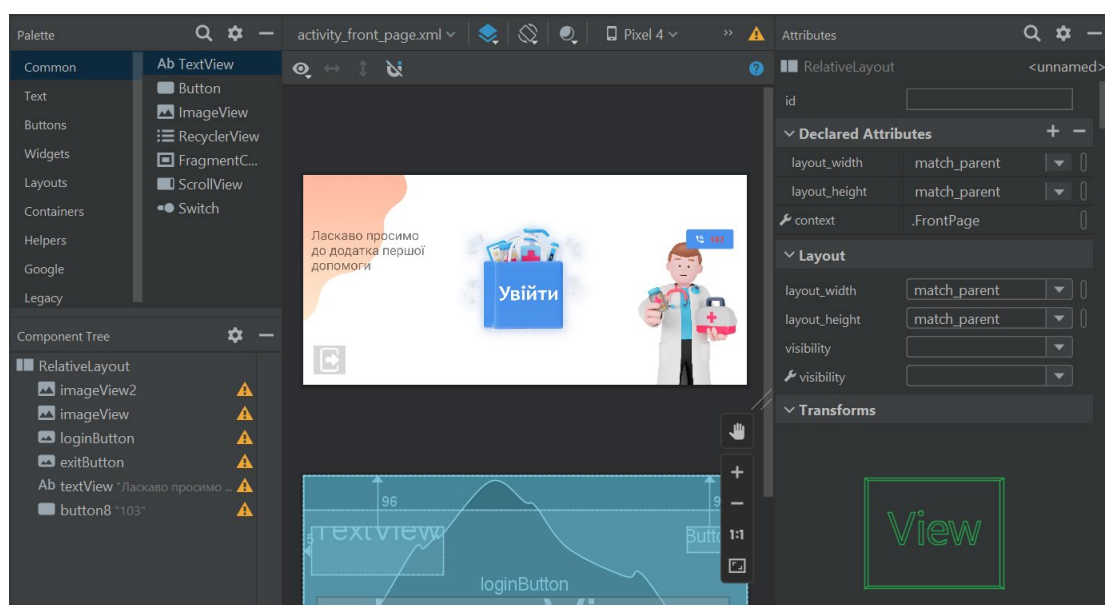


Рис. 3.5. Робота додатка у горизонтальному режимі

ВИСНОВКИ

Кваліфікаційна робота зосереджувалася на розробці повноцінного мобільного додатка для надання першої допомоги в умовах, де доступ до інтернету відсутній. За рахунок того, що додаток було створено в середовищі Android Studio з використанням мови програмування Java, додаток має широкі можливості для реалізації функціональності та забезпечення сумісності з платформою Android.

Однією з ключових задач розробки було забезпечення автономності та відсутності залежності від інтернет-підключення. Це було досягнуто через реалізацію локального сховища даних, що дозволяє додатку ефективно працювати, навіть коли зв'язок з інтернетом обмежений.

Нами детально проаналізовано задачу створення мобільного додатку для першої допомоги в умовах відсутності інтернет-підключення. Визначено вимоги до розробки та розглянуті різні методи та інструменти для її вирішення. Розглянута історія та особливості мови програмування Java, яка використовуватиметься для створення додатку для платформи Android. Проведено огляд інтегрованого середовища розробки Android Studio як основного інструменту для реалізації проекту. Надано аналіз існуючих програмних рішень у сфері першої допомоги, визначено їхні переваги та недоліки, що визначає унікальні особливості та функціональні вимоги до розроблюваного додатку.

1. Нами було розроблено програмне забезпечення та проведено аналіз технічних можливостей та вимог для розробки програми першої допомоги, яка функціонує в умовах обмеженого чи відсутнього інтернет-підключення.

2. Нами була проконтрольована валідація медичних аспектів: систематично переглянутий та валідований медичний зміст програми, забезпечуючи його відповідність найвищим медичним стандартам та рекомендаціям.

3. Нами було Психологічно адаптовано GUI: розроблено інтуїтивно зрозумілий та емпатичний інтерфейс, який підтримує користувачів у стресових ситуаціях та сприяє швидкій реакції.

4. Нами було протестовно ефективності додатку: було проведено серію тестів та сценаріїв в реальних умовах для перевірки функціоналу та ефективності програми, звертаючи увагу на її реальний вплив у різних ситуаціях.

5. Нами було детально задокументовано всі етапи розробки, тестування та вдосконалення програми для створення звіту, який буде служити джерелом інформації для подальших досліджень та розвитку.

Був визначений та розроблений загальний алгоритм функціонування програми, а також виготовлений програмний код, який включає в себе розробку як візуальної, так і функціональної складових. Під час проектування та створення програмного додатку було реалізовано відповідні елементи GUI та програмну логіку, забезпечуючи інтуїтивність та ефективність у користуванні додатком, що сприяє швидкій реакції на критичні ситуації та підтримує користувача. У результаті цього етапу вироблено функціонально повноцінний програмний продукт, готовий до подальшого тестування та вдосконалення.

Були представлені ілюстрації, які докладно розкривають взаємодію з програмним додатком, забезпечуючи зручний спосіб для звичайного користувача ознайомитися з принципом його функціонування. Також проведено аналіз та тестування додатку з метою забезпечення його плавної та ефективної роботи. Це дозволило нам мінімізувати виникнення ускладнень при використанні програми. Під час тестування нам вдалося підтвердити, що розроблений програмний продукт відповідає всім поставленим наперед технічним вимогам та виконує свої функції з високою ефективністю. Нами задокументовані всі етапи розробки, тестування та вдосконалення програми для

створення звіту, який буде служити джерелом інформації для подальших досліджень та розвитку у даному напрямку.

Розроблений додаток забезпечує надійну та доступну платформу для надання першої допомоги в екстрених ситуаціях, навіть при відсутності інтернет-підключення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мельник Т., Дудко В. Мільйони українців пересіли на сервіси, які критично залежать від інтернету. Як працюватимуть моно, «Дія», Glovo, Uklon та інші під час блекауту : Forbes.ua. Forbes.ua | Бізнес, мільярдери, новини, фінанси, інвестиції, компанії.
URL: <https://forbes.ua/innovations/zruchni-y-zvichni-onlayn-servisi-kritichno-zalezhat-vid-internetu-yak-pratsyuvatimut-mono-diya-glovo-lik24com-bolt-i-ukon-pid-chas-naygirshogo-stsenariyu-blekautu-30112022-10134> (дата звернення: 06.01.2024).
2. Інтерфейс мобільних додатків: ключові принципи, критерії оцінки та помилки. : ІТ-компанія повного циклу розробки програмних продуктів WEZOM - Київ, Україна. URL: <https://wezom.com.ua/ua/blog/dizajn-interfejsov-mobilnyh-prilozhenij> (дата звернення: 04.01.2024).
3. Meier R. Professional Android 4 Application Development. : Wiley & Sons, Incorporated, John, 2012.
4. Чи працюють мобільні додатки в автономному режимі? : Knowledge Base | Pipedrive. Knowledge Base | Pipedrive.
URL: <https://support.pipedrive.com/uk/article/do-the-mobile-apps-work-offline> (дата звернення: 08.01.2024).
5. Топ-5 мов програмування для розробників мобільних додатків : UASpectr. Новини про технології та бізнес | UASpectr.
URL: <https://uaspectr.com/2020/10/20/top-5-mov-programuvannya-dlya-rozrobnykiv-mobilnyh-dodatktiv/> (дата звернення: 06.01.2024).
6. Розробка мобільних додатків і їх види : Кафедра АПЕПС ТЕФ КПІ.
URL: <https://apeps.kpi.ua/rozrobka-mobilnykh-dodatktiv-i-yii-vidy> (дата звернення: 09.01.2024).
7. Hearn D. Computer graphics with OpenGL. 3-тє вид. Upper Saddle River, NJ : Pearson Prentice Hall, 2004. 857 с.

8. Nudelman G. Android Design Patterns: Interaction Design Solutions for Developers. : Wiley & Sons, Incorporated, John, 2013.

9. Canvas : Android Developers.

URL: <https://developer.android.com/reference/android/graphics/Canvas> (дата звернення: 11.01.2024).

10. Як народилася мова Java. Курси Java Львів : Комплексний курс Java від Java Academy. URL: <https://java.lviv.ua/yak-narodylasya-mova-java> (дата звернення: 10.01.2024).

11. Plynko P. Java. : JavaRush.

URL: <https://javarush.com/ua/groups/posts/uk.1878.bagatopotochnstjh-u-java> (дата звернення: 11.01.2024).

12. Android Studio - офіційне інтегроване середовище розробки Android. : Desde Linux.

URL: <https://blog.desdelinux.net/uk/android-studio-офіційна-інтегрована-середовище-розробки-android/> (дата звернення: 12.01.2024).

13. Hagos T. Android Studio IDE Quick Reference: A Pocket Guide to Android Studio Development : Apress, 2019. 200 с.

14. C. Stewart та ін. Android Programming: The Big Nerd Ranch Guide : Pearson Education, Limited, 2021.

15. IFRC. First Aid - IFRC : Google Play.

URL: <https://play.google.com/store/apps/details?id=com.cube.gdpc.fa&hl=ua&amr;gl=US> (дата звернення: 12.01.2024).

16. Prietzel F. Instant Aid - First Aid App : Google Play.

URL: https://play.google.com/store/apps/details?id=finn.prietzel.instant_aid (дата звернення: 13.01.2024).

17. Alty. TacticMedAid : Google Play.

URL: <https://play.google.com/store/apps/details?id=app.TacticMedAid> (дата звернення: 14.01.2024).

18.iFirstAid. : Google Play.

URL: <https://play.google.com/store/apps/details?id=com.ifirstaid> (дата звернення: 15.01.2024).

19.Page 4. Головна : Elib LNTU.

URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/ЕНП_АП%20готовий/page4.html (дата звернення: 14.01.2024).

20.Кобзар, М. С., Єрьоміна Н. С. Методи розробки мобільних додатків / Кобзар М. С., Єрьоміна Н. С. // Журнал «Проблеми інформатизації». – 2020.

21.Мобільна розробка: сучасні технології, інструменти та тренди. : Онлайн навчання ІТ спеціальностям | ІТ курси онлайн | ІТ STEP.

URL: https://cloud.itstep.org/blog_3/the-latest-technologies-in-mobile-application-development-top-tools-and-trends (дата звернення: 14.01.2024).

22.Медицинский работник позирует со стетоскопом | Бесплатно PSD Файл. Freepik. URL: https://ru.freepik.com/free-psd/healthcare-professional-posing-with-stethoscope_37222200.htm#from_view=detail_serie (дата звернення: 13.01.2024).

23.Як кольори впливають на людину : Запорозька Січ.

URL: <https://sich.zp.ua/iak-kolory-vplyvaiut-na-liudynu> (дата звернення: 15.01.2024).

24.Drongelen M. v. Android Studio Cookbook : Packt Publishing, Limited, 2015.

25.ImageButton : Android Developers.

URL: <https://developer.android.com/reference/android/widget/ImageButton> (дата звернення: 14.01.2024).

26.Android Button Clicks : Free Beginners Computer Tutorials and Lessons.

URL: https://www.homeandlearn.co.uk/android/android_button_clicks.html (дата звернення: 15.01.2024).

27.Naughton P., Schildt H. Java: The Complete Reference (Complete Reference Series) : Mcgraw-Hill Osborne Media, 1996. 886 с.

- 28.Розробити додаток на андроїд: інструкція зі створення : FoxmindEd.
URL: <https://foxminded.ua/rozrobyty-dodatok-na-android/> (дата звернення: 16.01.2024).
- 29.Базова аптечка : nov-rada. URL: <https://nov-rada.gov.ua/2023/01/18/bazova-aptechka-perelik-najneobkhidnishykh-preparativ-ta-medychnykh-pryladiv/> (дата звернення: 12.01.2024).
- 30.Clean Code: A Handbook of Agile Software Craftsmanship : Prentice Hall, 2008.
- 31.Дворецький М. Л., Нездолій Ю. О., Дворецька С. В., Кандиба І. О.
Розробка мобільних застосунків для OS Android : навч. посіб. : Вид-во ЧНУ ім. Петра Могили, 2021. – 140 с.
- 32.Порядок надання допомоги при зовнішній кровотечі : Автомобільна література | Видавництво Моноліт. URL: <https://xn--h1afceeb4a.xn--j1amh/pomosh-pri-dtp/okazanie-pervoj-pomoshhi-pri-naruzhnom-krovotechenii/> (дата звернення: 14.01.2024).
- 33.Bloch J. Effective Java : Pearson Education, Limited, 2017.
- 34.Події і їх обробка в Java : um.co.ua. URL: <http://um.co.ua/8/8-2/8-213617.html> (дата звернення: 15.01.2024).
- 35.VideoView Tutorial With Example In Android Studio : Abhi Android.
URL: https://abhiandroid.com/ui/videoview#google_vignette&gsc.tab=0 (дата звернення: 14.01.2024).
- 36.MediaController : Android Developers.
URL: <https://developer.android.com/reference/android/media/session/MediaController> (дата звернення: 14.01.2024).
- 37.Manifest.permission : Android Developers.
URL: <https://developer.android.com/reference/android/Manifest.permission> (дата звернення: 14.01.2024).

38.PackageManager : Android Developers.

URL: <https://developer.android.com/reference/android/content/pm/PackageManager> (дата звернення: 15.01.2024).

39.Редакція. Посібник із дизайну мобільних додатків під iOS та Android.

Основні відмінності платформ : UX PUB у Дизайн-спільнота.

URL: <https://ux.pub/editorial/posibnik-iz-dizainu-mobilnikh-dodatkov-pid-ios-ta-android-osnovni-vidminnosti-platform-1a5d> (дата звернення: 14.01.2024).

40.Принципи тестування : QALight.

URL: <https://qalight.ua/baza-znaniy/printsipi-testuvannya/> (дата звернення: 15.01.2024).

ДОДАТКИ

Додаток А.1

Програмний код головного меню додатку (FrontPage)

```
package com.example.first_aid;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;

public class FrontPage extends AppCompatActivity {

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

        ImageButton exitButton = findViewById(R.id.exitButton);
        ImageButton loginButton = findViewById(R.id.loginButton);

        exitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                runOnUiThread(new Runnable() {
                    @Override
```

```

        public void run() {

            finishAffinity();

        }

    });

}

});

```

```

loginButton.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

```

```

        Intent intent = new Intent(FrontPage.this, MenuPage.class);

```

```

        startActivity(intent);

    }

});

```

```

findViewById(R.id.button8).setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        // Виклик екстрених служб

        String phoneNumber = "103";

        Intent dialIntent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" +
phoneNumber));

```

```

        // Перевірка дозволу на дзвінок

        if (checkSelfPermission(android.Manifest.permission.CALL_PHONE) ==
android.content.pm.PackageManager.PERMISSION_GRANTED) {

```

```
        startActivity(dialIntent);  
    } else {  
        // Запит дозволу на дзвінок  
        requestPermissions(new  
String[]{android.Manifest.permission.CALL_PHONE}, 1);  
    }  
}  
});  
  
}  
}
```

Верстка головного меню (Layout)

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FrontPage">
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="0dp"
        android:scaleType="fitStart"
        app:srcCompat="@drawable/bekimeg" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="0dp"
        android:scaleType="fitEnd"
        app:srcCompat="@drawable/doctor1" />
```

<ImageButton

```
    android:id="@+id/loginButton"  
    android:layout_width="match_parent"  
    android:layout_height="256dp"  
    android:layout_alignParentBottom="true"  
    android:layout_marginBottom="72dp"  
    android:background="@android:color/transparent"  
    android:contentDescription="Login Button"  
    android:scaleType="centerInside"  
    android:src="@drawable/login_image" />
```

<ImageButton

```
    android:id="@+id/exitButton"  
    android:layout_width="68dp"  
    android:layout_height="65dp"  
    android:layout_alignParentStart="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginStart="15dp"  
    android:layout_marginBottom="15dp"  
    android:contentDescription="Exit Button"  
    android:scaleType="centerInside"  
    android:src="@drawable/exit_image" />
```

<TextView

```
    android:id="@+id/textView"  
    android:layout_width="247dp"  
    android:layout_height="90dp"  
    android:layout_alignParentStart="true"
```

```

    android:layout_alignParentTop="true"
    android:layout_marginStart="15dp"
    android:layout_marginTop="96dp"
    android:text="Паскаво просимо до додатка непуої допомоги"
    android:textSize="24sp" />

```

```

<Button

```

```

    android:id="@+id/button8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="96dp"
    android:layout_marginEnd="25dp"
    android:text="103"
    android:textColor="#F44336"
    android:textStyle="bold"
    app:icon="@android:drawable/stat_sys_speakerphone" />

```

```

</RelativeLayout>

```

Код сторінки меню із переліком тем (MenuPage)

```
package com.example.first_aid;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

import androidx.appcompat.app.AppCompatActivity;

public class MenuPage extends AppCompatActivity {

    private Button button3, button9, button10, button11, button12, button13, button14,
button15, button16, button17, button18, button20, button21, button22, button23;
    private Button button, button2;

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

        button3 = findViewById(R.id.button3);
        button9 = findViewById(R.id.button9);
        button10 = findViewById(R.id.button10);
        button11 = findViewById(R.id.button11);
```

```

button12 = findViewById(R.id.button12);
button13 = findViewById(R.id.button13);
button14 = findViewById(R.id.button14);
button15 = findViewById(R.id.button15);
button16 = findViewById(R.id.button16);
button18 = findViewById(R.id.button18);
button20 = findViewById(R.id.button20);
button21 = findViewById(R.id.button21);
button22 = findViewById(R.id.button22);
button23 = findViewById(R.id.button23);

```

```

button = findViewById(R.id.button);
button2 = findViewById(R.id.button2);

```

```

ImageButton imageButton2 = findViewById(R.id.imageButton2);
imageButton2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

```

```

        Intent intent = new Intent(MenuPage.this, FrontPage.class);

```

```

        startActivity(intent);
    }
});

```

```

button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

```

```
Intent intent = new Intent(MenuPage.this, ApplyingBandages.class);

startActivity(intent);
}
});
```

```
button9.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
```

```
Intent intent = new Intent(MenuPage.this, Bleeding.class);

startActivity(intent);
}
});
```

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        setTextSize(24);
    }
});
```

```
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        setTextSize(14);
```

```

        }
    });
}

private void setTextSize(int textSize) {
    Button[] buttons = {
        button3, button9, button10, button11, button12, button13, button14,
button15,
        button16, button18, button20, button21, button22, button23
    };

    for (Button button : buttons) {
        button.setTextSize(textSize);
    }
}

public void startFrontRage(View v) {
    startNewActivity(FrontPage.class);
}

public void startMedicinechestActiviti(View v) {
    startNewActivity(ImagePageActivity.class);
}

public void startBlood(View v) {
    startNewActivity(Bleeding.class);
}

public void startUneasinessActiviti(View v) {
    startNewActivity(Uneasiness.class);
}

public void startTraumaActiviti(View v) {
    startNewActivity(Trauma.class);
}

```

```

}
public void startInfarctActiviti(View v) {
    startNewActivity(Infarct.class);
}
public void startEpliepsiaActiviti(View v) {
    startNewActivity(Epliepsia.class);
}
public void startInjuryActiviti(View v) {
    startNewActivity(Injury.class);
}
public void startApputActiviti(View v) {
    startNewActivity(Apput.class);
}
public void startBurnsActiviti(View v) {
    startNewActivity(Burns.class);
}
public void startSuffocationActiviti(View v) {
    startNewActivity(Suffocation.class);
}
public void startAllergyActiviti(View v) {
    startNewActivity(Allergy.class);
}
public void startPoisoningActiviti(View v) {
    startNewActivity(Poisoning.class);
}
public void startFrostbiteActiviti(View v) {
    startNewActivity(Frostbite.class);
}
public void startSunstrokeActiviti(View v) {

```

```
        startNewActivity(Sunstroke.class);
    }
    private void startNewActivity(Class<?> cls) {
        Intent intent = new Intent(this, cls);
        startActivity(intent);
    }
}
```

Верстка екрану списку тем (Layout)

```

<?xml version="1.0" encoding="utf-8"?>
    <androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MenuPage">

        <ImageButton
            android:id="@+id/imageButton9"
            android:layout_width="56dp"
            android:layout_height="56dp"
            android:layout_marginStart="16dp"
            android:layout_marginTop="16dp"
            android:contentDescription="Перша домога"
            android:onClick="startMedicinechestActiviti"
            android:scaleType="centerInside"
            app:layout_constraintStart_toEndOf="@+id/imageButton2"
            app:layout_constraintTop_toTopOf="parent"
            app:srcCompat="@drawable/aptechca" />

        <ImageButton
            android:id="@+id/imageButton2"
            android:layout_width="56dp"

```

```

    android:layout_height="56dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:contentDescription="Περιοχή donomosa"
    android:onClick="startFrontRage"
    android:scaleType="centerInside"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/home" />

```

<Button

```

    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:fontFamily="sans-serif-smallcaps"
    android:text="αΑ"
    android:textAllCaps="false"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/imageButton9"
    app:layout_constraintEnd_toStartOf="@+id/button2"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

```

<Button

```

    android:id="@+id/button2"

```

```

    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:fontFamily="sans-serif-smallcaps"
    android:text="Aa"
    android:textAllCaps="false"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/imageButton9"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

```

<ScrollView

```

    android:id="@+id/scrollView2"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageButton9">

```

<LinearLayout

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

android:orientation="vertical">

<Button

android:id="@+id/button9"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fontFamily="sans-serif-smallcaps"
android:onClick="startBlood"
android:text="Кровомеча"
android:textStyle="bold"
android:visibility="visible" />

<Button

android:id="@+id/button13"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="16dp"
android:fontFamily="sans-serif-smallcaps"
android:onClick="startTraumaActiviti"
android:text="Перелому"
android:textStyle="bold"
android:visibility="visible" />

<Button

android:id="@+id/button10"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="16dp"
android:fontFamily="sans-serif-smallcaps"

```

    android:onClick="startInfarctActiviti"
    android:text="Серцевий напад"
    android:textStyle="bold"
    android:visibility="visible" />

```

```

<Button

```

```

    android:id="@+id/button15"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startUneasinessActiviti"
    android:text="Непритомність"
    android:textStyle="bold"
    android:visibility="visible" />

```

```

<Button

```

```

    android:id="@+id/button20"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startInjuryActiviti"
    android:text="Вогнепальне поранення"
    android:textStyle="bold"
    android:visibility="visible" />

```

```

<Button

```

```

    android:id="@+id/button3"

```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startBandagActiviti"
    android:text="Накладання пов'язок"
    android:textStyle="bold" />

```

<Button

```

    android:id="@+id/button21"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startApputActiviti"
    android:text="Відсікання кінцівки"
    android:textStyle="bold" />

```

<Button

```

    android:id="@+id/button14"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startBurnsActiviti"
    android:text="Onik"
    android:textStyle="bold" />

```

<Button

```

    android:id="@+id/button11"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startSuffocationActiviti"
    android:text="Удушся"
    android:textAlignment="center"
    android:textStyle="bold" />

```

<Button

```

    android:id="@+id/button23"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startEpliepsiaActiviti"
    android:text="Епілесія"
    android:textStyle="bold"
    android:visibility="visible" />

```

<Button

```

    android:id="@+id/button22"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp"
    android:fontFamily="sans-serif-smallcaps"
    android:onClick="startAllergyActiviti"
    android:text="Алергія"

```

android:textStyle="bold" />

<Button

android:id="@+id/button18"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="16dp"
android:fontFamily="sans-serif-smallcaps"
android:onClick="startPoisoningActiviti"
android:text="Отруєння"
android:textStyle="bold" />

<Button

android:id="@+id/button12"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="16dp"
android:fontFamily="sans-serif-smallcaps"
android:onClick="startFrostbiteActiviti"
android:text="Обмороження"
android:textStyle="bold" />

<Button

android:id="@+id/button16"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="16dp"
android:fontFamily="sans-serif-smallcaps"
android:onClick="startSunstrokeActiviti"

```
        android:text="Сонячный удар"  
        android:textStyle="bold" />
```

```
    </LinearLayout>
```

```
</ScrollView>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Код підтримки жестів масштабування для зображень

```
package com.example.first_aid;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.ScaleGestureDetector;
import android.widget.ImageButton;
import android.widget.ImageView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class ImagePageActivity extends AppCompatActivity {
    private ImageView imageView;
    private ImageButton backButton;
    private ImageButton resetButton;
    private ScaleGestureDetector scaleGestureDetector;
    private float scaleFactor = 1.0f;
    private float previousX, previousY;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_image_page);
        backButton = findViewById(R.id.imageButton);
        resetButton = findViewById(R.id.resetButton);
        imageView = findViewById(R.id.imageView);
        scaleGestureDetector = new ScaleGestureDetector(this, new ScaleListener());
        backButton.setOnClickListener(view -> finish());
        resetButton.setOnClickListener(view -> resetImage());
    }
}
```

```

    imageView.setOnTouchListener((v, event) -> {
        scaleGestureDetector.onTouchEvent(event);
        handleMoveEvent(event);
        return true;
    });
}

```

```

private void resetImage() {
    scaleFactor = 1.0f;
    imageView.setScaleX(scaleFactor);
    imageView.setScaleY(scaleFactor);
    imageView.setX(0);
    imageView.setY(0);
}

```

```

private class ScaleListener extends
ScaleGestureDetector.SimpleOnScaleGestureListener {
    @Override
    public boolean onScale(ScaleGestureDetector detector) {
        scaleFactor *= detector.getScaleFactor();
        scaleFactor = Math.max(0.5f, Math.min(scaleFactor, 2.0f));
        imageView.setScaleX(scaleFactor);
        imageView.setScaleY(scaleFactor);
        return true;
    }
}

```

```

private void handleMoveEvent(MotionEvent event) {
    switch (event.getAction()) {

```

```
case MotionEvent.ACTION_DOWN:
    previousX = event.getX();
    previousY = event.getY();
    break;
case MotionEvent.ACTION_MOVE:
    float dx = event.getX() - previousX;
    float dy = event.getY() - previousY;

    imageView.setX(imageView.getX() + dx);
    imageView.setY(imageView.getY() + dy);

    previousX = event.getX();
    previousY = event.getY();
    break;
}
}
}
```

Верстка сторінки із зображенням про першу допомогу

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/imageView3"
            android:layout_width="350dp"
            android:layout_height="wrap_content"
            android:layout_marginEnd="-100dp"
            android:scaleType="fitEnd"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:srcCompat="@drawable/doctor2" />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="411dp"
            android:layout_height="wrap_content"
            android:scaleType="fitStart"
```

```

    android:src="@drawable/aptechka"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```
<ImageButton
```

```

    android:id="@+id/imageButton"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_marginStart="16dp"
    android:layout_marginBottom="16dp"
    android:contentDescription="Пепуа дономога"
    android:src="?attr/homeAsUpIndicator"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

```

```
<ImageButton
```

```

    android:id="@+id/resetButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:contentDescription="Пепуа дономога"
    android:src="@android:drawable/ic_menu_revert"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
</RelativeLayout>
```

Код сторінки із текстом про кровотечу

```
package com.example.first_aid;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class Bleeding extends AppCompatActivity {

    private TextView textView;
    private TextView textView3;

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

        ImageButton imageButton3 = findViewById(R.id.imageButton7);
        Button button7 = findViewById(R.id.button7);
        Button button4 = findViewById(R.id.button4);
        Button button5 = findViewById(R.id.button5);
```

```

textView = findViewById(R.id.textView);
textView3 = findViewById(R.id.textView3);

```

```

button4.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        textView3.setTextSize(28);
        textView.setTextSize(32);
    }
});

```

```

button5.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        textView3.setTextSize(14);
        textView.setTextSize(12);
    }
});

```

```

imageButton3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent intent = new Intent(Bleeding.this, MenuPage.class);

        startActivity(intent);
    }
});

```

```

button7.setOnClickListener(new View.OnClickListener() {
    @Override

```

```
public void onClick(View view) {  
  
    Intent intent = new Intent(Bleeding.this, Bleeding2.class);  
  
    startActivity(intent);  
    }  
    });  
  
    }  
}
```

Верстка сторінки із текстом про кровотечу (Layout)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Bleeding">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

```
<ImageButton
    android:id="@+id/imageButton7"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_weight="1"
```

```

    android:contentDescription="Πεπουα donomoza"
    android:onClick="startFrontRage"
    android:scaleType="centerInside"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/home" />

```

```
<Button
```

```

    android:id="@+id/button4"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:layout_weight="1"
    android:fontFamily="sans-serif-smallcaps"
    android:text="aA"
    android:textAllCaps="false"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/imageButton9"
    app:layout_constraintEnd_toStartOf="@+id/button2"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

```

```
<Button
```

```

    android:id="@+id/button5"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"

```

```

    android:layout_weight="1"
    android:fontFamily="sans-serif-smallcaps"
    android:text="Aa"
    android:textAllCaps="false"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/imageButton9"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

```

```
</LinearLayout>
```

```
<ScrollView
```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent">

```

```
<LinearLayout
```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

```

```
<TextView
```

```

    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:text="Перша допомога у разі кровотечі"

```

```

    android:textAlignment="center"
    android:textStyle="bold" />

```

```

<ImageView

```

```

    android:id="@+id/imageView4"
    android:layout_width="match_parent"
    android:layout_height="180dp"
    android:scaleType="fitStart"
    app:srcCompat="@drawable/blood1" />

```

```

<Button

```

```

    android:id="@+id/button7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="СПОСОБИ ЗУПИНКИ КРОВОТЕЧІ" />

```

```

<TextView

```

```

    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"

```

android:text="Капілярна кровотеча: \n\n Капілярна кровотеча, як
 правило, не супроводжується значною крововтратою і досить легко
 зупиняється.\n\n Надати першу допомогу під час капілярної кровотечі слід за
 допомогою спиртового антисептику.

\n\n Обробіть ним край рани і накладіть асептичну марлеву

пов'язку. На саму рану під пов'язку накладають вату. \n\n Не затягуйте міцно. Часто капілярна кровотеча не потребує звернення до лікарні, крім випадків з великою площею пошкодженої поверхні.

\n\n Венозна кровотеча: \n\n У разі надання першої допомоги за венозної кровотечі на рану потрібно накласти асептичну пов'язку і перев'язати.

\n\n Якщо це не допомогло зупинити кров, потрібно накласти джгут нижче місця пошкодження.

\n\n Артеріальна кровотеча: \n\n Під час надання першої допомоги потрібно підняти кінцівку (якщо немає переломів) і накласти джгут вище місця пошкодження (ближче до тулуба).

\n\n Якщо джгута немає, притисніть пальцями артерію вище пошкодженої ділянки (у точці пульсації)." />

</LinearLayout>

</ScrollView>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

Код сторінки із відео про непритомність

```
package com.example.first_aid;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.MediaController;
import android.widget.VideoView;

public class Uneasiness extends AppCompatActivity {

    private static final String KEY_VIDEO = "uneasiness.KEY_VIDEO";
    Uri uri;

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

        VideoView videoView = (VideoView) findViewById(R.id.videoView);
        String videoPath = "android.resource://" + getPackageName() + "/" +
R.raw.v1;
        uri = Uri.parse(videoPath);
```

```
videoView.setVideoURI(uri);
```

```
MediaController mediaController = new MediaController(this);
```

```
videoView.setMediaController(mediaController);
```

```
mediaController.setAnchorView(videoView);
```

```
videoView.start();
```

```
if (savedInstanceState != null){
```

```
    uri = savedInstanceState.getParcelable(KEY_VIDEO);
```

```
}
```

```
if (uri!=null){
```

```
    videoView.setVideoURI(uri);
```

```
}
```

```
}
```

```
@Override
```

```
protected void onSaveInstanceState(@NonNull Bundle outState) {
```

```
    outState.putParcelable(KEY_VIDEO, uri);
```

```
    super.onSaveInstanceState(outState);
```

```
}
```

```
public void startActivitiHome(View v){
```

```
    Intent intent = new Intent(this,FrontPage.class);
```

```
    startActivity(intent);
```

```
}
```

```
public void startActivitiHome2(View v){
```

```
    Intent intent = new Intent(this,MenuPage.class);
```

```
    startActivity(intent);
```

```

    }
}

```

Додаток E.2

Верстка сторінки із відео про неприємність (Layout)

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Uneasiness">

<ImageButton
    android:id="@+id/imageButton6"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:backgroundTint="#78D0FF"
    android:contentDescription="Перша допомога"
    android:onClick="startActivitiHome2"
    android:scaleType="centerInside"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="?attr/actionModeCloseDrawable" />

```

<ImageButton

```

    android:id="@+id/imageButton5"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:backgroundTint="#78D0FF"
    android:contentDescription="Перша допомога"
    android:onClick="startActivitiHome"
    android:scaleType="centerInside"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/home" />

```

<TextView

```

    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="8dp"
    android:text="Перша допомога при непритомності"
    android:textAlignment="center"
    app:layout_constraintEnd_toStartOf="@+id/imageButton6"
    app:layout_constraintStart_toEndOf="@+id/imageButton5"
    app:layout_constraintTop_toTopOf="parent" />

```

<FrameLayout

```

    android:id="@+id/frameLayout"
    android:layout_width="wrap_content"

```

```

    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="32dp"

```

```

    android:configChanges="orientation|screenSize|screenLayout|smallestScreenSize"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView">

```

```

<VideoView
    android:id="@+id/videoView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp"

```

```

    android:configChanges="orientation|screenSize|screenLayout|smallestScreenSize"
    android:rotation="180.0" />

```

```

</FrameLayout>

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```