Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Мобільний застосунок для психологічного центру «medART»

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-4
Катерина КЛЕВЦОВА
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп’ютерна інженерія
(повна назва освітньої програми)

Керівник: доц. Наталія БОЛОГОВА
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Клевцовій Катерині Володимирівні
(прізвище, ім’я, по батькові)

1. Тема роботи Мобільний застосунок для психологічного центру «medART»

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи 1) середовище розробки: Android Studio:2) емулятор: AVD; 3) мова програмування: Java; 4) бібліотеки: Android SDK та AppCompat.

4. Перелік питань, що потрібно опрацювати у роботі
1) аналіз проблеми та огляд існуючих рішень;
2) аналіз існуючих мобільних застосунків та їх недоліки;
3) аналіз вимог до створення мобільного застосунку;
4) розробка рішення реалізації застосунку;
5) програмна реалізація застосунку;
6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій - 15 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	27.05.25-30.05.25	
2	Аналіз проблеми мобільних застосунків	31.05.25-03.06.25	
3	Розробка рішення реалізації застосунку	04.06.25-07.06.25	
4	Програмна реалізація застосунку	08.06.25-09.06.25	
5	Оформлення матеріалів кваліфікаційної	10.06.25-11.06.25	
	роботи		
6	Подання кваліфікаційної роботи керівникові та	12.06.25-13.06.25	
	її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	14.06.25-16.06.25	
	вання		

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Наталія БОЛОГОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 64 с., 19 рис., 2 табл., 1 дод., 11 джерел.

МОБІЛЬНИЙ ЗАСТОСУНОК, ANDROID, ANDROID STUDIO, ПСИХОЛОГІЧНЕ ЗДОРОВ'Я, ЦИФРОВІ ТЕХНОЛОГІЇ, ПТСР, МЕДИЧНИЙ ЦЕНТР, JAVA, МОБІЛЬНА РОЗРОБКА, , UX/UI, МОБІЛЬНА ПЛАТФОРМА, ANDROID ART, ЦИФРОВА ТРАНСФОРМАЦІЯ.

Метою кваліфікаційної роботи є розробка мобільного застсоунку для освітнього центру з метою залучення нових клієнтів, інформування наявних користувачів та реалізації практичних навичок Android-програмування в Android Studio. Для досягнення мети поставлено низку завдань: від аналізу ефективності мобільних рішень та вивчення вимог до програмного забезпечення – до створення інтерфейсу та розробки повноцінного функціонального продукту.

У ході виконання кваліфікаційної роботи було створено Android-проєкт в середовищі Android Studio, який розпочинається з меню File → New → New Project, після чого відкривається вікно конфігурації, в якому задаються параметри майбутнього застосунку: Application Name (назва програми), Company Domain (доменне ім'я компанії, наприклад, example.com), Package Name (унікальний ідентифікатор додатку, що формується на основі домену й назви програми, наприклад, com.example.appname), а також Project Location – шлях до директорії зберігання файлів проєкту. Після підтвердження переходить до вибору мінімальної версії Android, та визначення шаблону головної Activity, яка буде слугувати початковим екраном програми. Завершивши ці кроки, середовище автоматично створює проєкт із необхідною структурою – кодом, ресурсами та службовими файлами.

ABSTRACT

Bachelor’s thesis: 64 pages, 19 figures, 2tables, 1 appendices, 11 sources.

MOBILE APPLICATION, ANDROID, ANDROID STUDIO, PSYCHOLOGICAL HEALTH, DIGITAL TECHNOLOGIES, PTSD, MEDICAL CENTER, JAVA, MOBILE DEVELOPMENT, UX/UI, MOBILE PLATFORM, ANDROID ART, DIGITAL TRANSFORMATION.

The major goal of this thesis is to develop a mobile application for an educational center in order to attract new customers, inform existing users, and implement practical Android programming skills in Android Studio. To achieve this goal, a number of tasks were set: from analyzing the effectiveness of mobile solutions and studying software requirements to creating an interface and developing a fully functional product.

In order to the qualification work, an Android project was created in the Android Studio environment, which begins with calling the menu item File → New → New Project, after which a configuration window opens in which the basic parameters of the future application are set: Application Name (application name), Company Domain (company domain name, for example, example.com), Package Name (unique application identifier based on the domain and application name, for example, com.example.appname), and Project Location - the path to the project file storage directory. After confirming the data, the user proceeds to select the minimum Android version and define the template for the main Activity, which will serve as the application's home screen. After completing these steps, the environment automatically creates a project with the necessary structure: code, resources, and service files.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПТСР – посттравматичний стресовий розлад

AOSP – офіційний відкритий проєкт розробки операційної системи Android (англ., Android Open Source Project)

ART – це середовище виконання для операційної системи Android (англ., Android Runtime)

АРМ – автоматизоване робоче місце

ІАЦ – інформаційно-аналітичний центр

WDS – бездротова розподільча система (англ., Wireless Distribution System)

ВСТУП

Актуальність теми дослідження обумовлена потребою у збереженні соціально-психологічної стійкості особистості, її здатності до адаптації та активного функціонування в умовах глибоких суспільних змін. Сучасна епоха характеризується стрімкими трансформаціями, нестабільністю та невизначеністю, що супроводжуються кризовими явищами та значними змінами соціальних орієнтирів. Такі обставини чинять значний і часто суперечливий вплив на психологічний стан людини, її емоційне благополуччя та відчуття безпеки.

У цих умовах зростає запит суспільства на ефективні підходи до підтримки та відновлення психологічного здоров'я громадян, особливо тих, хто зазнав впливу гібридної війни, тому розробка мобільного застосунку для психологічного центру medART з метою підтримки психологічного благополуччя користувачів в умовах соціальної нестабільності, зараз дуже необхідна та актуальна.

Метою кваліфікаційної роботи є дослідження можливостей цифрових рішень, зокрема мобільного застосунку medART, для виявлення та розвитку внутрішнього потенціалу особистості, а також підвищення рівня її психологічної стійкості та захищеності в умовах суспільних трансформацій – від глобальних змін до індивідуальних міжособистісних викликів, зумовлених сучасними реаліями, включаючи гібридну війну проти України.

Об'єкт дослідження – процес цифрової підтримки психологічного здоров'я людини за допомогою мобільного застосунку в контексті сучасних соціальних викликів.

Завдання роботи:

- провести аналіз діяльності психологічного центру medART та цільової аудиторії його послуг;

- визначити основні психологічні потреби користувачів, зокрема в умовах війни, реформ і соціальної нестабільності;

- оцінити існуючі рішення на ринку мобільних застосунків для психологічної підтримки;

- сформулювати вимоги до мобільного застосунку з точки зору функціональності та зручності для користувача;

- розробити прототип мобільного застосунку medART, який сприятиме підвищенню психологічної стійкості та розвитку особистості;

- провести тестування застосунку та оцінити його ефективність з точки зору психологічного впливу на користувачів.

- запропонувати рекомендації щодо подальшого вдосконалення цифрових інструментів психологічної підтримки;

- провести аналіз діяльності психологічного центру medART та цільової аудиторії його послуг.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО МОБІЛЬНОГО ЗАСТОСУНКУ

1.1 Актуальність теми дослідження

Актуальність теми дослідження обумовлена потребою у збереженні соціально-психологічної стійкості особистості, її здатності до адаптації та активного функціонування в умовах глибоких суспільних змін. Сучасна епоха характеризується стрімкими трансформаціями, нестабільністю та невизначеністю, що супроводжуються кризовими явищами та значними змінами соціальних орієнтирів. Такі обставини чинять значний і часто суперечливий вплив на психологічний стан людини, її емоційне благополуччя та відчуття безпеки. Часто це призводить до психотравмуючих ситуацій, що потребують належної уваги з боку фахівців.

Нинішні виклики, які постають перед сучасною особистістю, вимагають глибокого соціально-психологічного аналізу. Зокрема, реформи в економіці, перебудова інституційної структури суспільства та вплив глобальних кризових процесів зумовлюють потребу у формуванні нових підходів до психологічної підтримки.

Особливої гостроти проблема набула у зв’язку з повномасштабною збройною агресією проти України. Наслідки воєнних дій спричинили серйозні загрози: зростання рівня стресу, поширення посттравматичних стресових розладів, виснаження особистісних ресурсів, погіршення загального стану здоров’я населення та зниження якості життя. Молодь, зокрема, стикається з труднощами у формуванні життєвих орієнтирів і планів.

У цих умовах зростає запит суспільства на ефективні підходи до підтримки та відновлення психологічного здоров’я громадян, особливо тих,

хто зазнав впливу гібридної війни. Важливим є не лише теоретичне обґрунтування, але й практична реалізація методів, спрямованих на сприяння особистісним трансформаціям, повернення до мирного життя, формування адаптивності, самостійності та психологічної автономії. Це вимагає створення комплексної теоретико-методичної основи для підготовки психологів, соціальних працівників, волонтерів та інших фахівців, що працюють з людьми, які постраждали від воєнних подій.

Вивчення шляхів гармонізації особистісного життєвого простору як важливого чинника формування людського потенціалу країни охоплює різноманітні аспекти від самопізнання та індивідуального життєвого проектування до розбудови власного майбутнього в умовах активної взаємодії з соціальним середовищем. Це включає залучення громадян до процесів демократизації, розвиток громадянської активності, впровадження новітніх підходів у соціальну взаємодію, ефективну співпрацю в колективах, формування наукового та прикладного соціально-психологічного світогляду. У фокусі також перебуває розвиток культури соціального діалогу, зниження рівня соціального напруження, запобігання негативним суспільним явищам та деструктивному впливу інформаційного простору. Окрема увага приділяється засобам психологічної підтримки осіб, які пережили тривалу травматизацію внаслідок воєнних дій.

Згідно з даними дослідження «Is the global prevalence rate of adult mental illness increasing? Systematic review and meta-analysis» (2019), проведеного командою дослідників, включаючи Вітсе А. Тола та Марка ван Оммерена, було встановлено, що глобальна поширеність психічних розладів серед дорослого населення зросла на 17,9% у період з 1978 по 2015 рік. Це зростання, хоча й незначне, вважається переважно наслідком демографічних змін, таких як старіння населення та урбанізація (рисунок 1.1) [1].

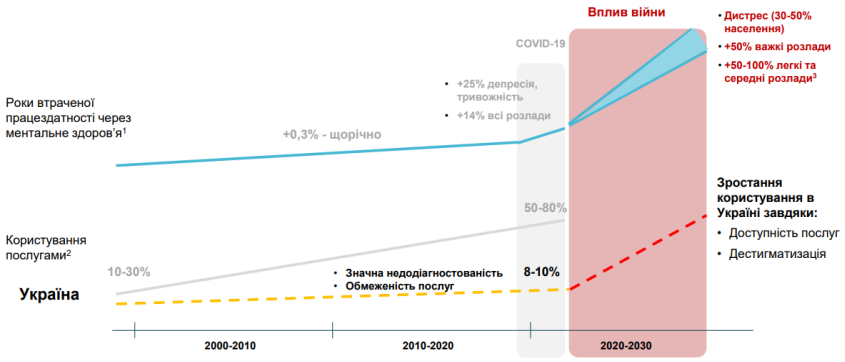

Рисунок 1.1 – Динаміка зростання глобальної поширеності психічних розладів серед дорослого населення

Попри наявність соціальних послуг в Україні, вони не здатні повністю охопити всі потреби громадян, особливо в умовах війни. Значна частина населення переживає психологічне навантаження, однак більшість не звертається по спеціалізовану допомогу через бар'єри доступу або брак інформації. Сьогодні система підтримки потребує суттєвого розширення, зокрема у напрямі надання первинної психологічної допомоги (рисунок 1.2).

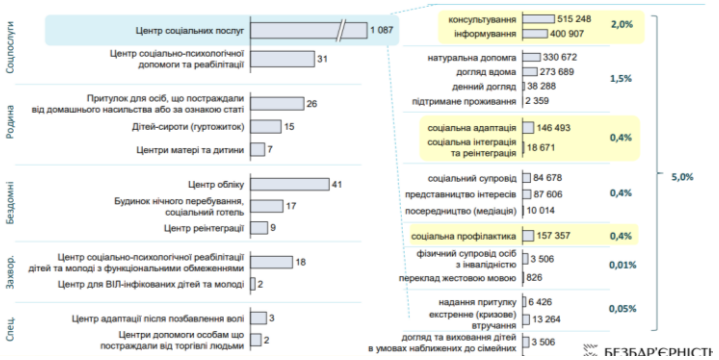

Рисунок 1.2 – Соціальні послуги не охоплюють всі потреби

За статистикою, лише 8% людей мають клінічні прояви, що потребують спеціалізованого лікування, серед яких 3% – це випадки, коли необхідна госпіталізація. Утім, це лише вершина айсберга. Переважна більшість українців – особливо тих, хто постраждав від війни, втрати, або внутрішнього переміщення – мають потребу в психологічній підтримці, яка

не завжди передбачає лікарське втручання, але є критично важливою для збереження ментального здоров’я нації (рисунок 1.3)[2]

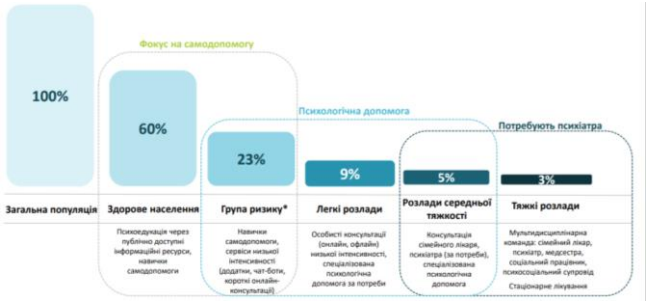

Рисунок 1.3 – Співвідношення клінічних випадків і загальної потреби у психологічній підтримці серед постраждалого населення

1.2 Аналіз існуючих мобільних застосунків для психологічної підтримки

В Україні активно розвивається сектор мобільних застосунків для підтримки психологічного здоров’я, зокрема в умовах війни та соціальних викликів. Наразі існує кілька таких застосунків, які орієнтовані на різні цільові аудиторії та потреби. Ось деякі з них:

а) База – застосунок розроблений для надання психологічної підтримки ветеранам та військовослужбовцям. Він містить едукаційні матеріали про психічне здоров’я, вправи для самодопомоги та кризову кнопку для екстрених ситуацій. Застосунок доступний на платформах Android та iOS [3].

Переваги:

1) адаптований для потреб військових;

2) містить інтерактивні вправи та екстрену допомогу;

3) безкоштовний, доступний на Android і iOS.

Недоліки:

1) низька відомість серед цивільного населення;

2) обмежена кількість інструментів для довготривалої терапії.

б) Melty – це безкоштовний застосунок, який допомагає користувачам визначати свій психоемоційний стан та дізнаватися про дієві методи самодопомоги. Він доступний для завантаження в Google Play та App Store [4].

Переваги:

1) дружній інтерфейс і простота у використанні;

2) можливість швидко оцінити свій стан;

3) підходить широкій аудиторії.

Недоліки:

1) відсутність живого зв'язку з психологом;

2) діагностика суб'єктивна й не замінює професійного підходу.

в) Wishnya – перший український додаток, який допомагає сповіщати друзів чи родину за заданим сценарієм у разі виникнення складних ситуацій. Цей застосунок може бути корисним у кризових моментах, коли потрібна швидка допомога близьких [5].

Переваги:

1) висока швидкість реагування в кризових моментах;

2) актуально для людей із тривожними розладами чи ПТСР.

Недоліки:

1) обмежена функціональність: більше підходить як SOS-інструмент, ніж як терапевтичний засіб.

2) потребує налаштувань і сценаріїв наперед.

г) Розкажи мені – безкоштовна інтернет-платформа психологічної підтримки, яка працює цілодобово. Вона надає можливість отримати психологічну допомогу онлайн, що особливо актуально в умовах війни [6].

Переваги:

1) працює 24/7;

2) допомога надається фахівцями;

3) не потребує інсталяції – доступ через веб.

Недоліки:

1) переважно текстовий/чат-формат, не завжди зручно;

2) потрібне стабільне інтернет-з’єднання.

ґ) Хаб стійкості – це платформа з доступними сервісами для психологічної підтримки українців та українок у кризовий час. Вона об'єднує різні ресурси та сервіси, що надають психологічну допомогу [7].

Переваги:

централізований доступ до великої кількості сервісів;

орієнтація як на спеціалістів, так і на користувачів.

Недоліки:

може бути перевантаженим для новачків;

не має власного мобільного додатку –доступ через сайт.

д) Spokiy – це онлайн-підтримка ментального здоров’я для дорослих і дітей. Вона пропонує різноманітні ресурси та інструменти для підтримки психічного здоров’я [8].

Переваги:

1) працює з різними віковими групами;

2) орієнтація на сімейний формат підтримки.

Недоліки:

1) не завжди має спеціалізований контент під конкретні запити (наприклад, ПТСР);

2) потрібно більше інтерактивності для залучення молоді.

Таким чином, в Україні існує кілька мобільних застосунків, спрямованих на підтримку психологічного здоров’я різних категорій населення. Ці сервіси надають доступ до психологічної допомоги онлайн, що є особливо важливим в умовах війни та соціальних викликів.

Нижче наведено порівняльну характеристику деяких з них, з акцентом на цільову аудиторію, функціональність та доступність (таблиця 1.1).

Таблиця 1.1 – Порівняльна таблиця застосунків

Назва застосунку	Цільова аудиторія	Основні функції	Доступність	Платність
База	Ветерани та військовослужбовці	Вправи самодопомоги, кризова кнопка	Android, iOS	Безкоштовний
Melty	Молодь (15-30 років)	Тестування психоемоційного стану, вправи	Android, iOS	Безкоштовний
Wishnya	Широке коло користувачів	Сповіщення близьких	Android, iOS	Безкоштовний
Розкажи мені	Дорослі українці	Онлайн-консультації з психологами	Веб-платформа	Безкоштовний
Хаб стійкості	Усі, хто потребує підтримки	Онлайн-консультації, самодопомога	Веб-платформа	Безкоштовний
Spokiy	Дорослі та діти	Діагностика, онлайн-консультації	Веб-платформа	Безкоштовний

Усі розглянуті застосунки є безкоштовними та доступними на різних платформах, що забезпечує широкий доступ до психологічної підтримки.

Застосунки пропонують різні функції – від самодіагностики та самодопомоги до онлайн-консультацій з фахівцями.

Кожен застсоунок орієнтований на певну аудиторію, що дозволяє користувачам обрати найбільш відповідний сервіс для своїх потреб, але згідно наведеної таблиці наглядно показано, що не у всіх є застосунок на

телефон, тому потрібно адаптуватися під умови сьогодення, та впроваджувати мобільні технології у життя людей, у подальшому це полегшить їх життя.

Таким чином, в Україні існує широкий спектр безкоштовних мобільних застосунків та онлайн-платформ, спрямованих на підтримку психологічного здоров’я різних категорій населення. Ці сервіси є важливими інструментами для надання допомоги в умовах війни та соціальних викликів.

1.3 Аналіз діяльності психологічного центру medART

Психологічний центр medART може мати низку переваг порівняно з іншими мобільними застосунками для психологічної підтримки, особливо якщо врахувати його потенційний розвиток у вигляді мобільного застосунку. Ось порівняльний аналіз і ключові переваги medART.

1.3.1 Основна місія та підхід

medART позиціонує себе як безпечний простір, де кожен може отримати підтримку, розуміння та рішення своїх проблем. Центр об'єднує висококваліфікованих психологів та психотерапевтів з багаторічним досвідом, які використовують сучасні методики, що довели свою ефективність [9].

Центр надає такі послуги:

- індивідуальні консультації: психологічна допомога для дорослих, підлітків та дітей;
- групова терапія: сеанси до 7 осіб, спрямовані на спільне опрацювання проблем;
- сімейна та парна терапія: робота з міжособистісними стосунками та конфліктами;

- арт-терапія: творчі методи для емоційного розвитку та самовираження;

- немедикаментозна психотерапія: робота з птср, депресією, окр, емоційним вигоранням та іншими розладами;

- майстер-класи та тренінги: освітні заходи для дорослих і дітей, включаючи тренінги для медичних працівників [9].

1.3.2 Інноваційні методики

medART впроваджує інноваційні підходи, такі як VR-терапія, яка допомагає клієнтам опрацьовувати страхи, тривожність та інші емоційні стани через віртуальну реальність .

Центр орієнтується на:

- дорослих: осіб, які стикаються з тривожністю, депресією, стресом, емоційним вигоранням та іншими психологічними труднощами;

- пари та сім'ї: подружжя та родини, які потребують допомоги у вирішенні конфліктів та покращенні взаєморозуміння;

- підлітків та дітей: молодь, яка потребує підтримки в період адаптації, розвитку та емоційного становлення;

- осіб, які пережили травматичні події: людей, які зазнали психологічних травм, включаючи військові дії та інші стресові ситуації;

- медичних працівників: фахівців, які потребують професійної підтримки та розвитку навичок через спеціалізовані тренінги [9].

Таким чином, психологічний центр medART є багатопрофільною установою, яка надає широкий спектр психологічних послуг, орієнтованих на різні групи населення. Використання сучасних та інноваційних методик дозволяє ефективно вирішувати психологічні проблеми клієнтів та сприяти їх особистісному розвитку (таблиця 1.2).

Таблиця 1.2 – Порівняння: medART та інші застосунки

Критерій	medART (у концепції мобільного застосунку)	Інші застосунки
Підхід до клієнта	Індивідуальний, з можливістю підбору психотерапевта з урахуванням запиту, віку, статі, типу терапії	Переважно автоматизовані сервіси або загальні консультації
Живі консультації	Прямий доступ до фахівців medART – психотерапевтів, психіатрів, коучів	Переважно чат або короткі онлайн-сесії
Офлайн + онлайн формат	Поєднання мобільного застосунку з реально існуючим центром, де можна пройти офлайн-сесії	Лише онлайн або чат
Додаткові сервіси	Арт-терапія, психодіагностика, психокорекційні програми, групова терапія, курси	Переважно лише консультації або вправи самодопомоги
Якість фахівців	Сертифіковані практикуючі спеціалісти з клінічним досвідом	У деяких застосунках – волонтери або студенти психології
Персоналізація	Історія клієнта, план терапії, нагадування, трекінг прогресу	Мінімальна персоналізація або відсутня
Арт-терапевтична складова	Унікальна особливість medART – залучення творчості в терапію	У більшості не передбачено
Український контекст	Врахування менталітету, воєнних травм, роботи з внутрішньо переміщеними особами	Не всі адаптовані до українських реалій

Переваги medART як мобільного застосунку:

- комплексний сервіс – не просто «застосунок із вправами», а частина повноцінного психологічного центру з командою спеціалістів;
- інтеграція з офлайн-сервісами – користувачі можуть переходити з онлайн у реальні консультації, що особливо цінно для глибшої терапії;
- можливість обрати терапевта – з профілями спеціалістів, відгуками, рейтингами;
- арт-компонент – в medART є акцент на творчість як інструмент самопізнання й емоційного відновлення (малювання, колажі, мандали тощо);
- мова та культура – україномовний інтерфейс, фахівці з розумінням актуальних проблем, пов’язаних з війною, вимушеною міграцією, втратою тощо;
- інноваційний UX – застосунок може включати гейміфікацію, щоденники емоцій, трекінг стану та особистісного зростання.

У той час як більшість існуючих застосунків фокусуються або на кризовій підтримці, або на самодопомозі, medART має потенціал стати персоналізованим психологічним сервісом повного циклу. Це дозволяє йому зайняти нішу між простими застосунками та дорогими приватними консультаціями, пропонуючи якісну, доступну, глибоку допомогу.

2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ MEDART

2.1 Аналіз вимог мобільного застосунку

Для того щоб нам чітко усвідомлювати в якому напрямку рухатися надалі, нам необхідно скласти список вимог, яким має відповідати наша програма.

Технічні вимоги:

- відповідати розробленому дизайну;
- працювати без серйозних помилок;
- працювати з Android 8.0;
- працювати без візуальних затримок.

Функціональні вимоги.

Користувач заходить у програму, без будь-якої авторизації, як тільки програма завантажитися має відобразитися розділ «Новини».

Для переходу в інший розділ повинні бути кнопка відповідна дизайну вгорі зліва та можливість відкриття меню жестом по екрану зліва направо. У відкритому меню має бути логотип компанії у шапці меню, список розділів нижче, з інтуїтивно зрозумілими картинками та назвами розділів.

У розділі «Новин» має бути можливість прокручування, як тільки завантажені новини закінчуються у списку, повинні завантажитись ще обмежена кількість новин, якщо новин більше немає, то більше завантаження відбуватися не повинно.

У розділі «Напрямку» при його відкритті має з'являтися список напрямків, при натисканні на елемент списку має відкритися нове вікно або список підкатегорій з таким же функціоналом. У новому вікні має відображатися вміст, який надійшов із сервера.

У розділі «Контакти» повинні відображатися контактні дані

підприємства.

У розділі «Про нас» відображається таблиця з працівниками організації з фото співробітників, при натисканні на картинку відкривається докладна інформація про співробітника в новому вікні.

Згідно з вище перерахованими вимогами, які ми визначаємо для нашого мобільного застосунку, можна провести аналіз, який допоможе нам ефективно, швидко і без зайвих фінансових та людських витрат здійснити комплексну, самодостатню розробку програми. Застосунок можна буде легко розширювати, підтримувати, містити програмний код, що легко читається, буде зрозумілий людям, які будуть його розширювати і підтримувати надалі.

Для того щоб глибше зрозуміти принципи розробки, необхідно визначити, для якої саме програмної платформи буде створюватися застосунок. У даному випадку мова йде про операційну систему Android, яка вже понад десять років утримує лідерство на ринку мобільних операційних систем.

За останніми даними, Android займає понад 70% глобального ринку мобільних пристроїв. Платформа активно розвивається з моменту свого запуску у 2008 році, і станом на 2025 рік актуальною є версія Android 14, представлена у жовтні 2023 року. Водночас Android 15 перебуває на стадії бета-тестування і очікується до офіційного релізу восени 2025 року. Завдяки відкритому вихідному коду Android (через проєкт AOSP – Android Open Source Project), розробники з усього світу можуть створювати власні застосунки, модифікації системи та навіть цілі прошивки [4].

Успіх Android можна прослідкувати ще з початку 2010-х років, коли Google активно інвестувала у розвиток спільноти розробників. Наприклад, у 2010 році компанія почала ініціативи з безкоштовного поширення власних пристроїв, зокрема Nexus One. Спочатку багато хто сприйняв це як містифікацію, проте згодом такі кроки зміцнили довіру до бренду. Раніше для розробників створювалися спеціальні пристрої під назвою Android Dev

Phone (ADP) – першими такими апаратами стали модифіковані версії T-Mobile G1 (HTC Dream) та Nexus S (на базі Android 2.3 Gingerbread), які дозволяли отримувати рут-доступ та експериментувати з системою без обмежень [5].

З роками Android перетворився на повноцінну екосистему, яка охоплює не лише смартфони, а й планшети, телевізори, автомобільні системи (Android Auto), смарт-годинники (Wear OS), а також пристрої IoT. Завдяки цьому він є однією з найбільш універсальних платформ для розробки застосунків, що дає широкі можливості для впровадження інновацій у різних сферах – від освіти до охорони здоров'я.

Android – операційна система для смартфонів, інтернет-планшетів, електронних книг, цифрових програвачів, наручних годинників, ігрових приставок, нетбуків, смартбуків, окулярів Google, телевізорів та інших пристроїв. потім купила Google. Згодом Google ініціювала створення альянсу Open Handset Alliance (OHA), який зараз займається підтримкою та подальшим розвитком платформи [6].

Android залишається однією з найпотужніших і найгнучкіших мобільних операційних систем, що постійно розвивається. Серед її ключових переваг – розгалужений фреймворк, який надає розробникам широкий набір API для створення різноманітних типів додатків. Ці API підтримують можливості повторного використання та заміни компонентів, як від самої системи, так і від сторонніх додатків, що забезпечує модульність та високу інтеграцію між елементами системи.

Замість застарілої Dalvik, сучасні версії Android (починаючи з 5.0) використовують ART (Android Runtime), що забезпечує більш ефективне виконання коду, кращу продуктивність та оптимізацію використання ресурсів пристрою. Операційна система підтримує багатий набір графічних бібліотек, зокрема OpenGL ES та Vulkan, для розробки 2D- і 3D-графіки. Також доступна розширена підтримка мультимедійних форматів, включаючи

Ogg Vorbis, MP3, AAC, H.264, H.265 (HEVC), WebP та інші. Крім того, Android пропонує API для роботи з камерою, GPS-навігацією, компасом, акселерометром, гіроскопом, сенсорним екраном, контролерами та ін. Для обробки мультимедіа та аудіо у фоновому режимі використовуються сучасні бібліотеки, як-от Jetpack Media3 або ExoPlayer, що особливо важливо при створенні ігор та мультимедійних застосунків [7]. Не всі Android-пристрої володіють усіма цими можливостями – апаратний поділ. Звичайно, список можливостей Android не вичерпується згаданими мною. Архітектура Android формується із набору компонентів. Кожен компонент побудований з урахуванням елементів нижчого рівня. На рисунок 2.1 представлений короткий огляд основних компонентів Android [14].

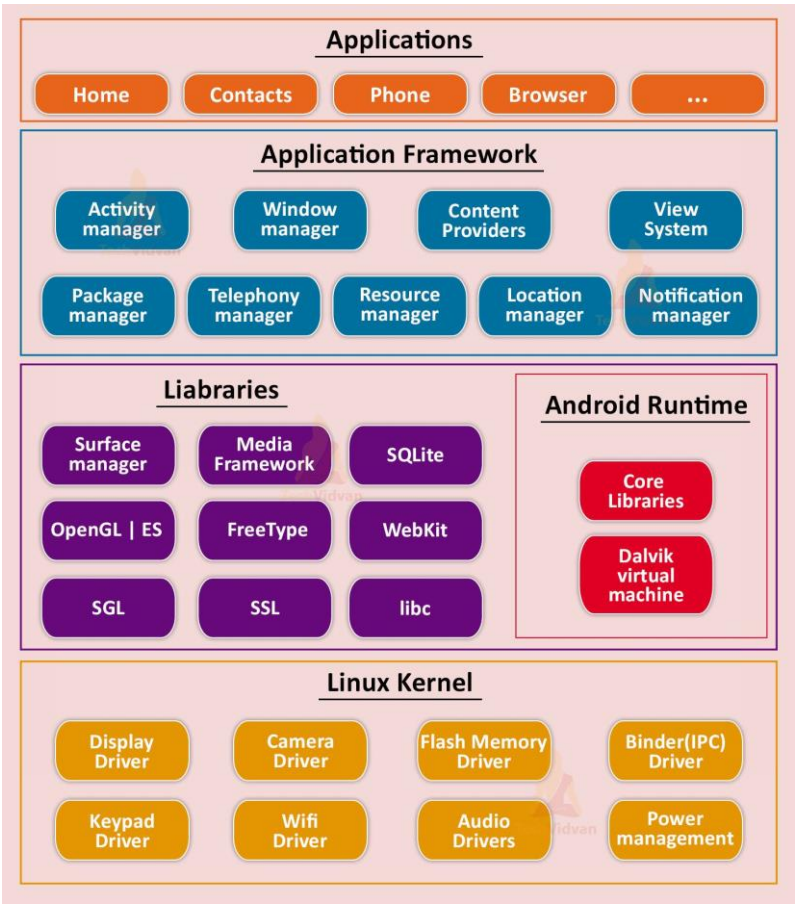

Рисунок 2.1 – Архітектура Android

Фреймворк програми пов'язує разом системні бібліотеки та середовище виконання, створюючи таким чином користувальницьку сторону Android. Фреймворк управляє додатками та пропонує продумане середовище, в якому вони працюють. Розробники створюють додатки для цього фреймворку за допомогою набору програмних інтерфейсів на Java, що охоплюють такі області, як розробка інтерфейсу користувача, фонові служби, оповіщення, управління ресурсами, доступ до периферії і т.д. буд. Усі ключові програми, які постачаються разом з ОС Android (наприклад, поштовий клієнт), написані за допомогою цих API. Програми, будь-які з інтерфейсом або фоновими службами, можуть зв'язуватися з іншими програмами. Цей зв'язок дозволяє одному додатку використовувати компоненти інших.

Подібний алгоритм знімає значну частину ноші з програміста, а також дозволяє налаштувати різноманітність аспектів поведінки Android.

В програмуванні для Android використовуються об'єктно-орієнтовані технології, тому ми наведемо огляд об'єктних технологій. Іменник може бути адекватно представлений програмним об'єктом у поняттях атрибутів (наприклад, ім'я, колір і розмір) і поведінок (наприклад, обчислення, переміщення та передача даних) Розробники програм бачать, що використання модульної структури та об'єктно-орієнтованого проектування при розробці додатків підвищує продуктивність роботи. простіше зрозуміти та змінити [15].

2.2 Архітектура застосунку

Дизайн інтерфейсу користувача є фактором, що впливає на три основні показники якості програмного продукту: його функціональність, естетику і продуктивність.

Функціональність є чинником, який розробники додатків часто звертають основну увагу. Вони намагаються створювати програми так, щоб

користувачі могли виконувати свої завдання, і їм було зручно це робити. Функціональність важлива, проте це не єдиний показник, який повинен враховуватися в ході розробки.

Естетичний зовнішній вигляд самого додатка та способу його подання дозволяє сформувати у споживача позитивну думку про програму. Проте естетичні характеристики дуже суб'єктивні і описати їх кількісно набагато складніше, ніж функціональні вимоги чи показники продуктивності. Вся естетика програми часто зводиться до простого вибору: чи співвідносяться між собою використовувані кольори, чи передають елементи інтерфейсу їх призначення і сенс операцій, що представляються, що відчуває людина при використанні тих чи інших елементів управління і наскільки успішно він їх використовує.

Продуктивність, а також надійність, також впливають на перспективу застосування програми. Якщо програма добре виглядає, має просте і зручне управління, але, наприклад, повільно промальовує екрани, регулярно «підвисає» на десяток-другий секунд або, того гірше, падає з критичною помилкою при некоректних діях користувача, у нього, ймовірно, буде мало шансів на тривалу експлуатацію. У свою чергу, швидка та стабільна робота програми можуть частково компенсувати його не найстильніший дизайн або відсутність якихось вторинних функцій.

Для забезпечення успішної роботи користувача від дизайнера інтерфейсу потрібно дотримуватися балансу між вище переліченими факторами протягом усього життєвого циклу розробки програми. Це досягається послідовним і ретельним опрацюванням деталей інтерактивної взаємодії на кожному з етапів розробки інтерфейсу користувача включають:

а) проектування:

1) функціональні вимоги: визначення мети розробки та вихідних вимог;

2) аналіз користувачів: визначення потреб користувачів, розробка

сценаріїв, оцінка відповідності сценаріїв очікуванням користувачів;

3) 4концептуальне проектування: моделювання процесу, для якого розробляється програма;

4) логічне проектування: визначення інформаційних потоків у додатку;

5) фізичне проектування: вибір платформи, на якій буде реалізовано проект та засоби розробки.

б) реалізація:

1) прототипування: розробка паперових та/або інтерактивних макетів екранних форм;

2) конструювання: створення програми з урахуванням можливості зміни її дизайну.

2.3 Моделювання користувацького інтерфейсу

Прототипи стали незамінним етапом професійної розробки програмного забезпечення. Прототипи полегшують життя всім людям, залученим до створення того чи іншого проекту: клієнту, менеджерам, дизайнерам і розробникам. Прототип візуалізує кінцевий продукт і допомагає виявити серйозні проблеми на ранніх етапах, допомагаючи уникнути проблем на пізніх стадіях розробки.

Починати прототипування ми почнемо з розробки лівого меню, оскільки цей функціонал є основним у додатку. Перше, що буде додано до макету це так звані StatusBar і Toolbar (рисунок 2.2).

Рисунок 2.2 – Зверху Statusbar, знизу Toolbar

Statusbar – це візуальний графічний елемент Android для взаємодії користувача з системою.

Toolbar – також відомий як панелі дій, є одним з найважливіших елементів дизайну в діяльності нашої програми, так як він забезпечує візуальну структуру та інтерактивні елементи, які знайомі користувачам.

Для створення меню ми звернемося до Google Material Pattern [8], Material Design в якому описується найкращі рішення для побудови дизайну програми.

У рекомендаціях матеріального дизайну Google йдеться про те, що об'єкти інтерфейсу користувача повинні відкидати тіні, як і об'єкти реального світу. Згідно з рекомендаціями дизайнерів Google, меню повинно містити елементи у вигляді списку з іконками та написами, так само мати заголовок з необхідною функціональністю (рисунок 2.3).

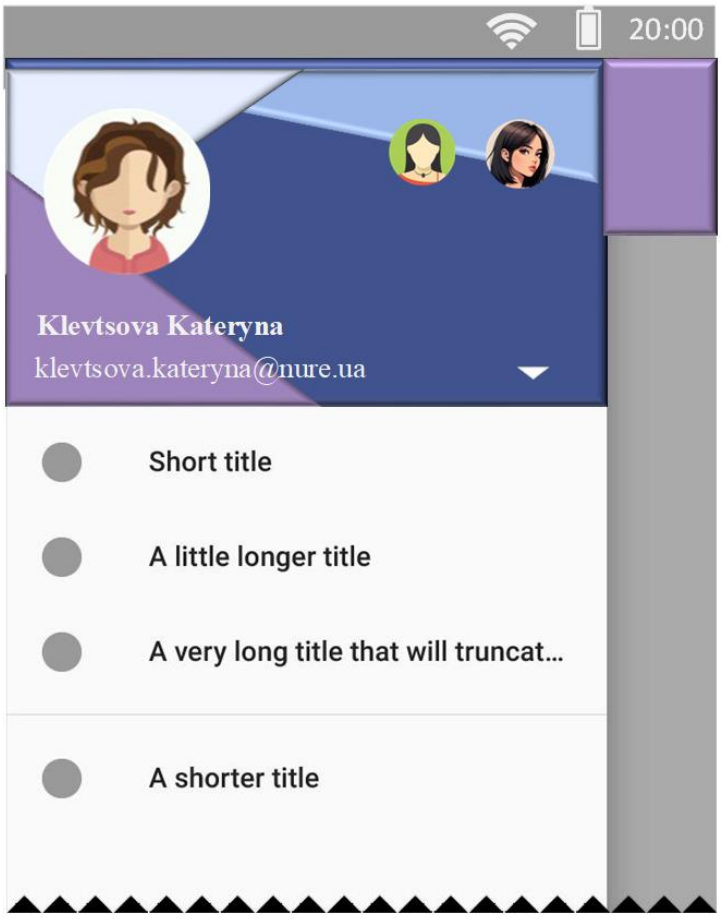

Рисунок 2.3 – Типовий вид меню програми

Наш Navigation Drawer містить заголовок з логотипом компанії в стилі Material Design [9], нижче йдуть список розділів програми, якими здійснюється навігація розділів (рисунок 2.4).

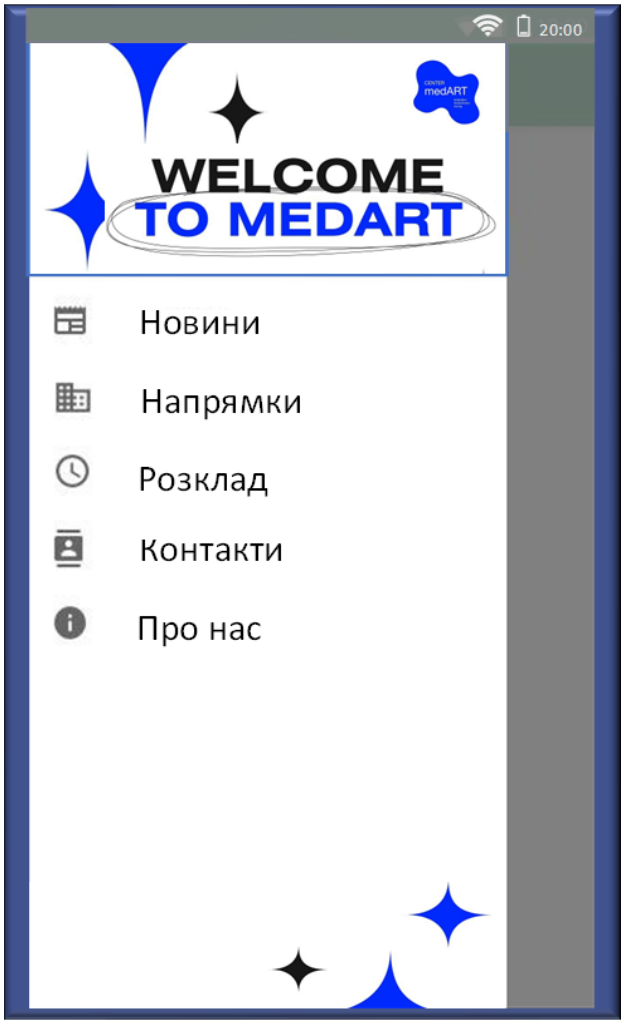

Рисунок 2.4 – Navigation Drawer

Спрототипований Navigation Drawer відповідає вимогам Material Design для Android і надає користувачеві програми зручні можливості для навігації по розділах.

Розробляючи прототип розділу новин, було застосувано дизайн карток, який використовують багато програм, в Android вони називаються CardView (рисунок 2.5).

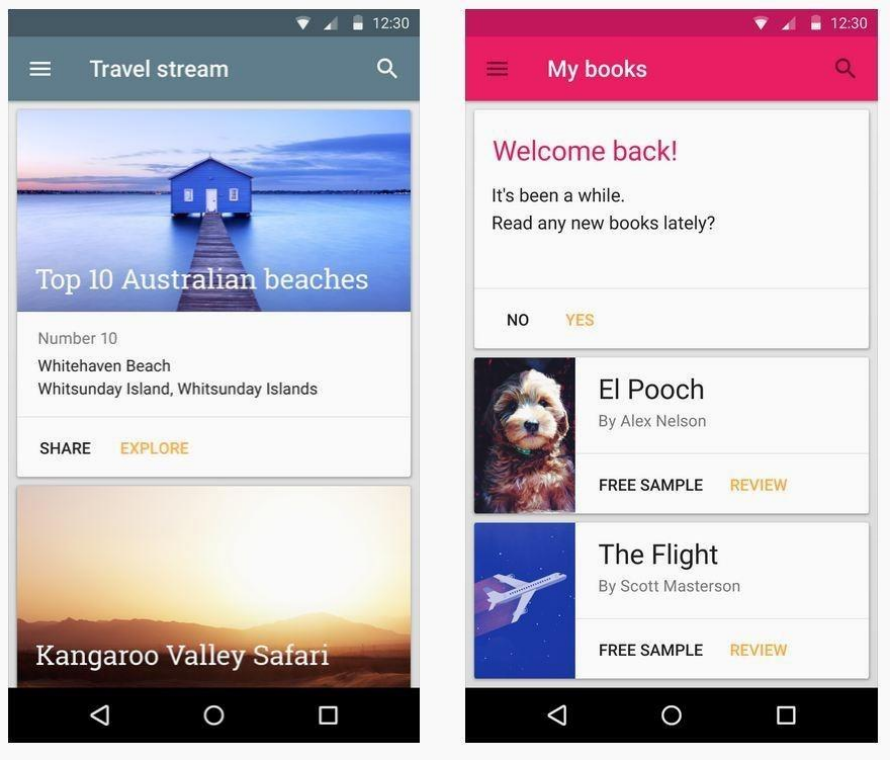

Рисунок 2.5 – Приклад використання CardView

Новий компонент CardView з'явився в Android Lollipop але завдяки бібліотеці сумісності доступний і для старих пристроїв. CardView.

У цій роботі CardView підходить для створення списку розділу новин, оскільки кожна новина має картинку, заголовок, дату події, короткий опис. Для підкреслення заголовка використовувалася смуга чорного кольору з прозорістю 56 відсотків, даний відсоток прозорості є найбільш прийнятним варіантом і стандартним для даних заголовків (рисунок 2.6).

Рисунок 2.6 – Стрічка оновлених статей

На дисплеї елементів у списках дуже поширена картина в мобільних застосунках. Користувач бачить список елементів, і їх можна прокручувати. Якщо він вибирає один із елементів списку, це може оновити ActionBar або тригери докладну екран для вибору.

Android надає ListView класу, який здатний відображати прокручувати список елементів. Ці елементи можуть бути будь-якого типу.

Розділ напряму у зв'язку з його структурою матиме стандартний обліковий вигляд (рисунок 2.7).

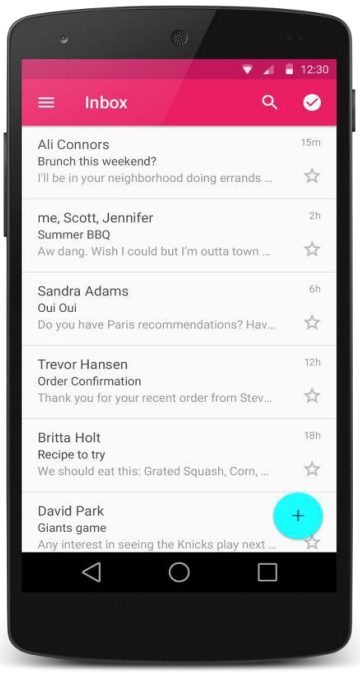

Рисунок 2.7 – Приклад використання ListView

Після вивчення структури розділу розкладів на сайті установи було створено прототип розділу у застосунку (рисунок 2.8).

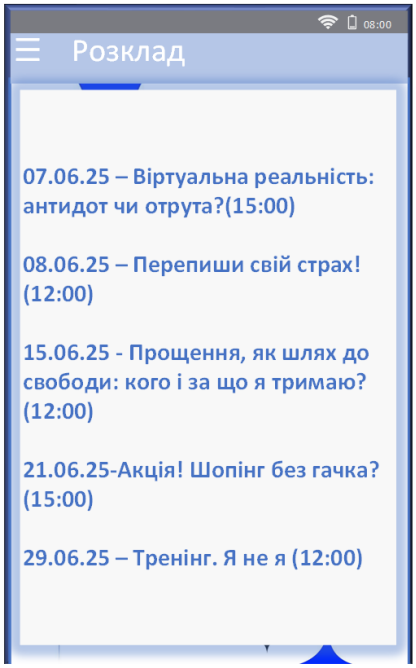

Рисунок 2.8 – Розділ розкладу тренінгів

Подальше прототипування розділу напряму було спрямоване на вікно з

інформацією про напрям, був обраний варіант з появою нового вікна. У цьому вікні буде відображатися інформація, доступна користувачеві. Спочатку має відображатися картинка із зображенням заданим на віддаленому сервері, знизу буде знаходитись докладна інформація у певному, зручному форматуванні. Для зручної навігації згори-зліва знаходиться кнопка назад для закриття вікна з детальною інформацією. Користувач також повинен розуміти, що знаходитись у новому вікні для цього змінюється заголовок вікна на назву відкритого напрямку (рисунок 2.9).

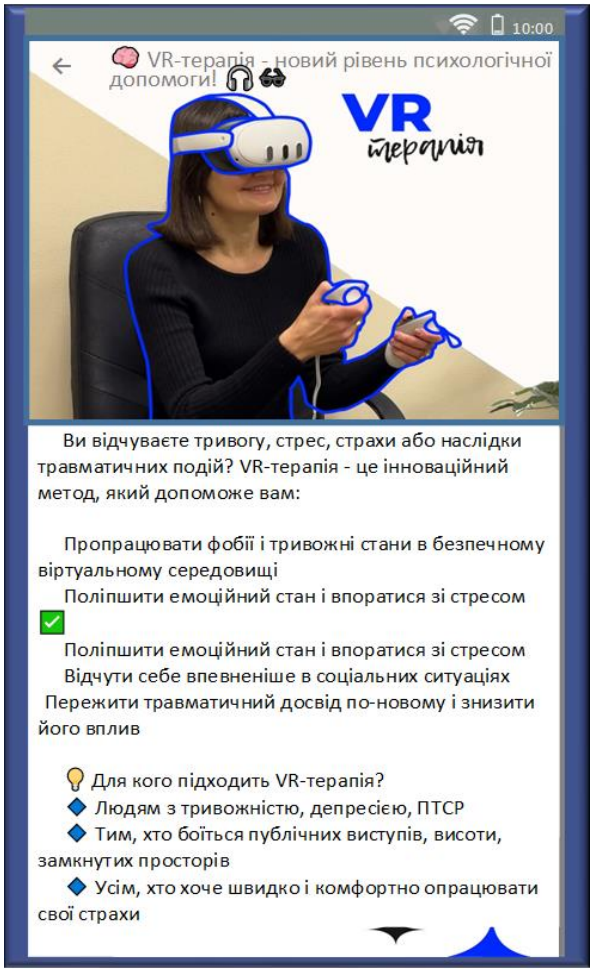

Рисунок 2.9 – Детальна інформація про напрямок

Розділ із контактною інформацією має просту форму відображення текстової інформації (рисунок 2.10).

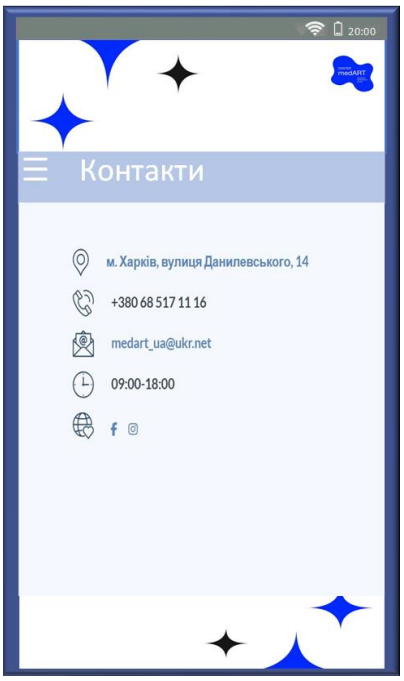

Рисунок 2.10 – Розділ контакти

Останній розділ «Про нас», у розділі йдуть співробітники підприємства у вигляді іконок (рисунок 2.11).

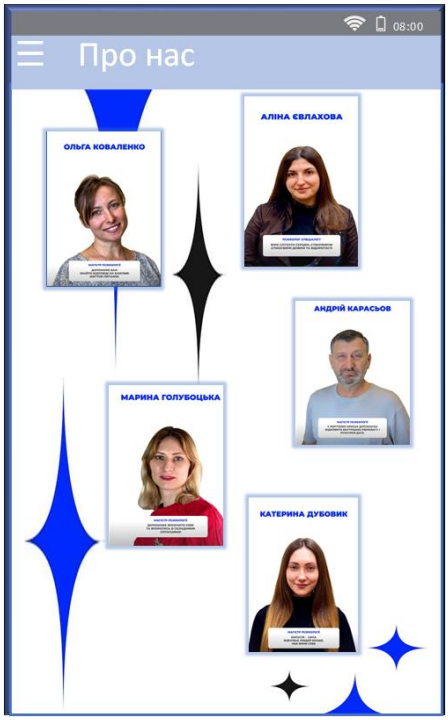

Рисунок 2.11 – Розділ «Про нас» – Список співробітників

3 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Використані технології та інструменти розробки

Стрімке зростання продажів пристроїв на базі Android відкриває визначні можливості перед розробниками програм Android.

На платформі Android зараз працюють смартфони, планшети, електронні книги, роботи, реактивні двигуни, супутники NASA, ігрові приставки, холодильники, телевізори, камери, медичні пристрої, «розумний годинник», автомобільні інформаційні системи (для керування радіо, GPS, телефонами, термостатами тощо) і багато інших пристроїв.

За останніми прогнозами, доходи від мобільних додатків (за всіма мобільними платформами) в 2025 році досягнуть 270 мільярдів доларів. [3].

Одна з головних переваг платформи Android – її відкритість. Операційна система Android побудована на основі відкритого вихідного коду і знаходиться у вільному розповсюдженні. Це дозволяє розробникам отримати доступ до вихідного коду Android та зрозуміти, яким чином реалізовані методи, властивості та функції програм. Будь-який користувач може взяти участь, де можна отримати вихідний код Android, дізнатися про ідеологію, закладену в основу операційної системи з відкритим кодом, та отримати ліцензійну інформацію.

Відкритість платформи сприяє швидкому оновленню. На відміну від закритої системи iOS компанії Apple, доступної лише на пристроях Apple, система Android доступна на пристроях десятків виробників обладнання (OEM, Original Equipment Manufacturer) та телекомунікаційних компаній у всьому світі. Усі вони конкурують між собою, що йде на користь кінцевому споживачеві.

Для розробки програм для ОС Android потрібно встановити Android

Studio.

Встановлення Android SDK. Інструменти для розробки Android SDK можна завантажити на веб-сайті для розробників. Під час встановлення можна вибрати потрібні для розробки платформи та елементи SDK.

При розробці програм Android використовується Java – одна з найпоширеніших мов програмування. Використання Java стало логічним вибором для платформи Android, тому що це потужна, вільна і відкрита мова, відома мільйонам розробників. Досвідчені програмісти Java можуть швидко освоїти Android програмування, використовуючи інтерфейси Google Android API (Application Programming Interface) та інші розробки незалежних фірм.

Мова Java є об'єктно-орієнтованою, надає розробникам доступ до потужних бібліотек класів, що прискорюють розробку додатків.

Програмування графічного інтерфейсу користувача керується подіями, які реагують на події, що ініціюються користувачами, такі як торкання екрана. Крім безпосереднього написання коду програм можна скористатися середами розробки Eclipse та Android Studio, що дозволяють збирати графічний інтерфейс з готових об'єктів, таких як кнопки та текстові поля, перетягуючи їх у певні місця екрана, додаючи підписи та змінюючи їх розміри.

Ці середовища розробки дозволяють швидко та зручно створювати, тестувати та налагоджувати програми Android.

Компоненти графічного інтерфейсу Android називаються уявленнями (views). Вертикальне уявлення LinearLayout використовується для розміщення тексту та графіки так, щоб кожне уявлення займало половину вертикального простору LinearLayout. Компонент LinearLayout також дозволяє розміщувати вистави по горизонталі. Для виведення тексту в програмі буде використовуватися компонент TextView, а графіка відображатиметься у компоненті ImageView. Графічний інтерфейс, створений для стандартної програми, містить компонент TextView. Різні

параметри цього компонента - текст, розмір шрифту, колір тексту, розмір щодо компонента ImageView в LinearLayout і т. д. – налаштовуються у вікні властивостей середовища розробки. Потім ми перетягнемо компонент ImageView з палітри в макет графічного інтерфейсу та налаштуємо його властивості, включаючи джерело графічних даних та позицію в LinearLayout.

Мова XML (eXtensible Markup Language, тобто розширювана мова розмітки) є природним способом опису графічних інтерфейсів. Розмітка XML добре читається як людиною, так і комп'ютером; відобразити макет у макетному редакторі та згенерувати код Java, який формує графічний інтерфейс на стадії виконання.

У кожного застосунку існує тема, що визначає оформлення стандартних компонентів, які використовують. Тема програми вказується у файлі AndroidManifest.xml програми. Можна налаштувати різні аспекти теми (наприклад, складові колірної схеми), визначаючи ресурси у файлі styles.xml, що знаходиться в папці res/values програми.

Ресурсний файл style.xml містить стиль з ім'ям AppTheme, посилання на який включається до файлу AndroidManifest.xml програми для призначення теми. Цей стиль також визначає батьківську тему, яка може розглядатися як аналог суперкласу Java – новий стиль успадковує атрибути батьківської теми та їх значення за замовчуванням. Стиль може перевизначити атрибути батьківської теми значеннями, адаптованими для конкретних програм (наприклад, для використання у програмі фірмової колірної гами компанії). Було використано цю концепцію для налаштування трьох кольорів, які використовуються в програмі. Як згадувалося раніше, шаблони додатків Android Studio тепер включають підтримку бібліотек AppCompat, що дозволяють використовувати нові можливості Android у старих версіях платформи. За замовчуванням Android Studio обирає батьківську тему Theme.AppCompat.Light.DarkActionBar, одну з декількох стандартних тем бібліотеці AppCompat – програми, що використовують цю

тему, відображаються на світлому фоні, а у верхній частині програми розташовується темна панель програми. Всі теми AppCompat використовують рекомендації для матеріального дизайну Google для оформлення графічних інтерфейсів.

Встановлення JDK та JRE. Для розробки потрібне середовище виконання Java Runtime Environment (JRE), комплект розробника Java Development Kit (JDK), які можна завантажити з офіційного сайту Oracle.

Створення віртуального пристрою Android. Емулятор «Android Virtual Device» (AVD) дозволяє тестувати програми на віртуальному мобільному пристрої з ОС Android

Проект – це група пов'язаних файлів (наприклад, файли коду, ресурси та графічні файли), що утворюють програму. Робота над програмою починається зі створення проекту. Щоб створити програму в середовищі Android Studio для операційної системи Android. Відкриємо Android Studio. Для створення нового проекту треба перейти до пункту меню File -> New-> New Project (рисунок 3.1). Після цього у нас з'явиться діалогове вікно створення нового проекту.

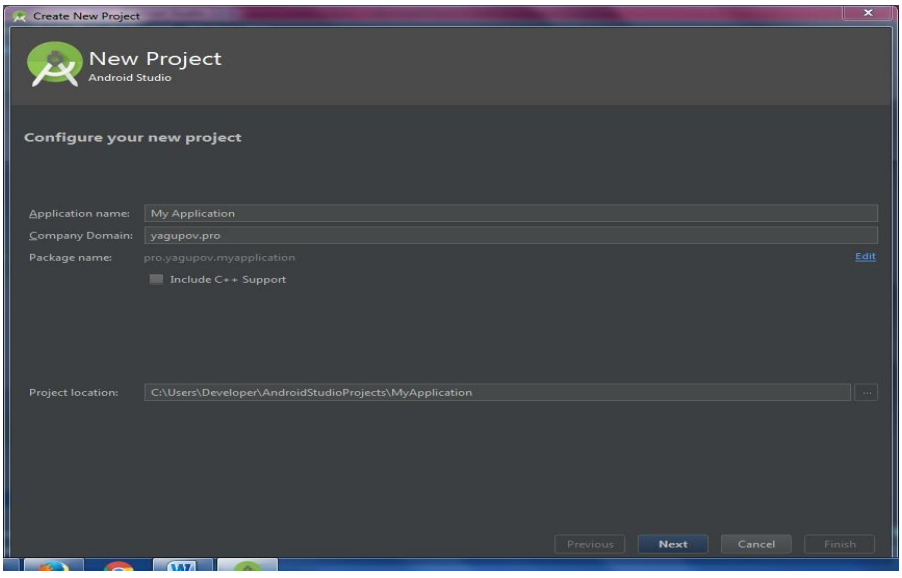

Рисунок 3.1 – Вікно створення нового проекту

На кроці Configure your new project майстра Create New Project (рисунок 3.1) введемо таку інформацію:

- Application Name: – ім'я програми;

- Company Domain: – доменне ім'я веб-сайту компанії. створення можна використати ім'я example.com;

- Package Name: – ім'я пакета Java для вихідного коду програми. Android та магазин Google Play використовують це ім'я як унікальний ідентифікатор програми, який повинен залишатися постійним у всіх версіях програми, яка надсилається до магазину Google Store. Ім'я пакета зазвичай починається з доменного імені компанії або установи, записаної у зворотному порядку. Наприклад, було використано доменне ім'я deitel.com, тому імена пакетів починаються із префіксу example.com. Далі зазвичай слідує ім'я програми. За загальноприйнятими угодами в імені пакета використовуються лише літери нижнього регістру без пропусків. За промовчанням IDE вибирає ім'я пакета на основі тексту, який вводиться в полях Application Name та Company Domain. Щоб змінити ім'я Package, клацніть посилання Edit праворуч від згенерованого імені пакета;

- Project Location: – шлях до папки на вашому комп'ютері, в якій буде зберігатися проект. праворуч від поля та виберіть папку для збереження проекту. Після того, як папка буде вибрана, клацніть на кнопці OK, а потім перейдіть до наступного кроку кнопкою Next (рисунок 3.2).

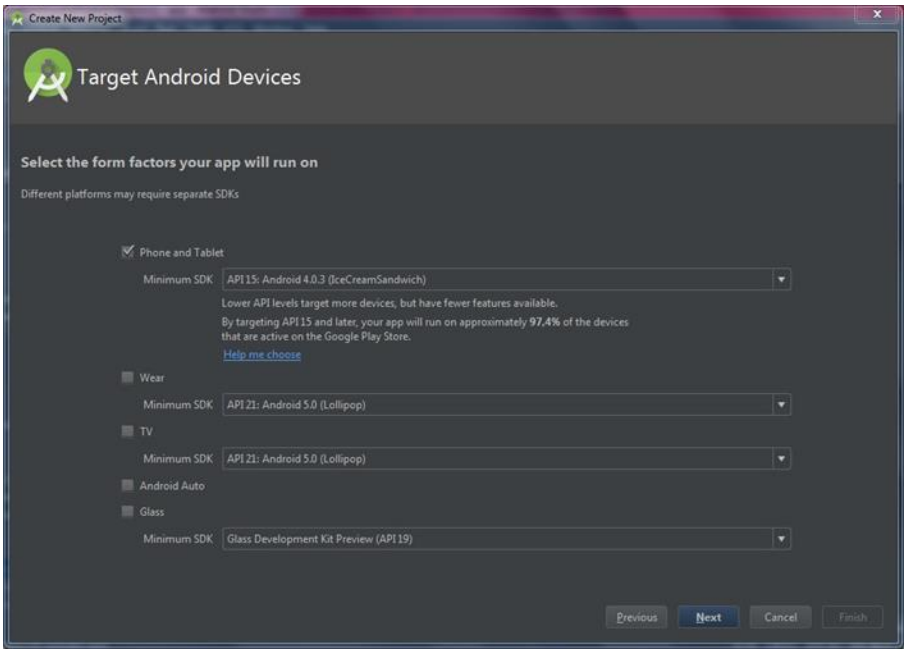

Рисунок 3.2 – Вибір версії Android

На цьому кроці буде запропоновано встановити мінімальну версію проекту, що підтримується. За промовчанням встановлюється версія Android 8.0, що покриває майже 94% пристроїв Android. Залишимо за замовчуванням і натиснемо кнопку Next.

На наступному кроці потрібно вибрати Activity (рисунок 3.3), яка буде використовуватися за замовчуванням для головного екрана програми.

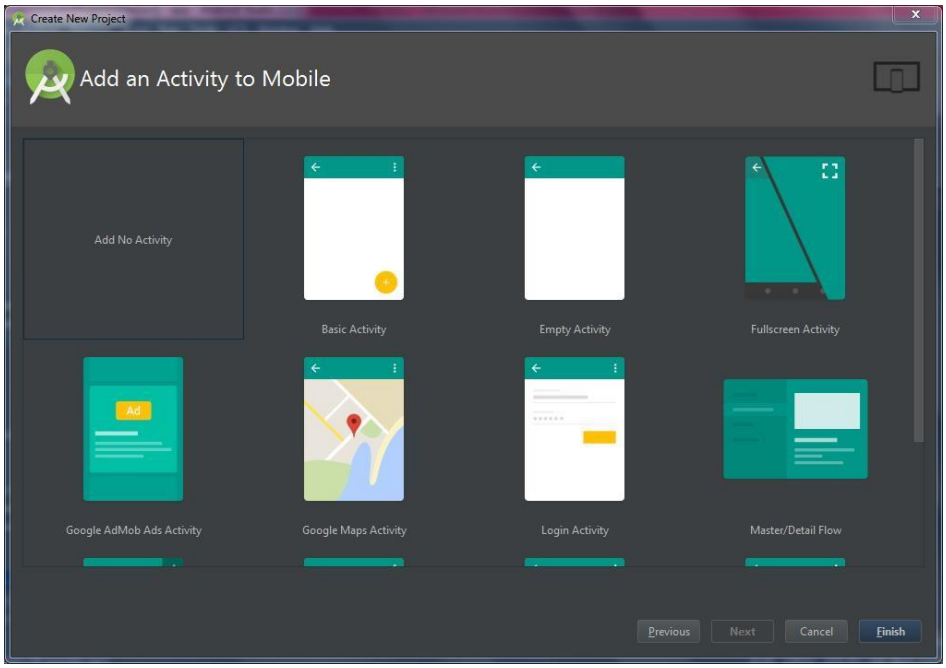

Рисунок 3.3 – Вибір головної Activity

У цьому застосунку не додано Activity звичайним чином, тому що відбувається неефективна генерація програмного коду. Пізніше буде створено базову реалізацію батьківської Activity інших вікон докладання, дотримання принципу DRY [18].

3.2 Реалізація основного функціоналу

Розглянемо структуру проекту програми під ОС Android, що створюється за умовчанням.

Проект може включати різні модулі та всі модулі описуються файлом setting.gradle.

І якщо ми подивимося на структуру проекту, весь значний код – файли інтерфейсу, класи java і т.д. за замовчуванням знаходяться у папці (модулі) app

Файл build.gradle містить інформацію, яка використовується при побудові проекту.

Кожен модуль має свій файл build.gradle, який визначає конфгурацію

побудови проекту, специфічну для цього модуля. Так, якщо ми подивимося на вміст app папки, то, якраз знайдемо в ній такий файл. На початковому етапі дані файли менш важливі, досить розуміти, навіщо вони потрібні.

За замовчуванням кожен проект включає один модуль – app. Власне, весь код, з яким будемо працювати, розташовується всередині цього модуля.

У цьому модулі можемо побачити кілька папок і файлів, з яких найважливішими для нас є:

- каталог libs – призначений для зберігання бібліотек, що використовуються програмою;
- каталог src – призначений для зберігання вихідного коду. Він містить низку підкаталогів. Вихідні коди розміщуються в папці main.

Папка main має складну структуру (рисунок 3.4).

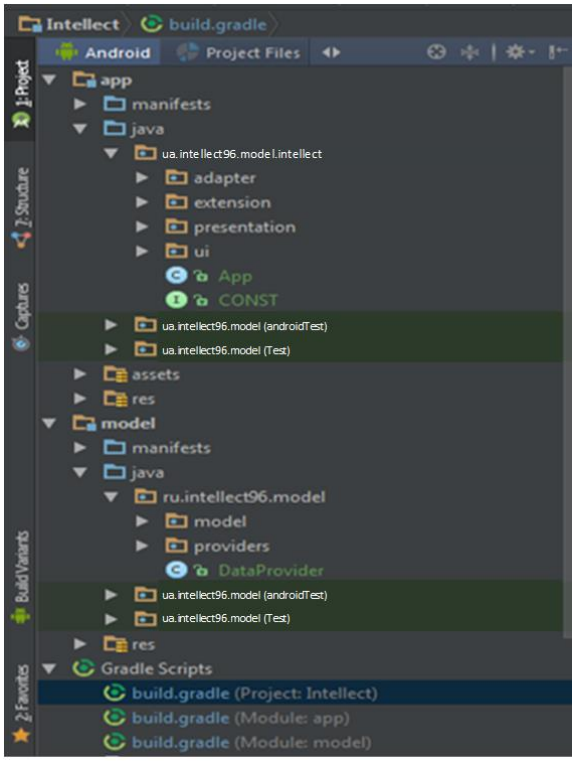

Рисунок 3.4 –Структура Android проекту

Після розгляду загальних питань побудови проекту можемо розпочати самої реалізації докладання. Почнемо зі створення зі службового класу App,

який відповідає за ініціалізацію допоміжних компонентів (лістинг 3.1).

Лістинг 3.1 – Клас App

```
public classAppextendsApplication{

@Override
publicvoidonCreate() {
super.onCreate();

DataProvider.onCreate(getApplicationContext());
}
}
```

Клас App успадковується від службового системного класу Application який реалізує фундаментальну концепцію додатка, у якого перевизначено метод onCreate, який викликається при запуску програми та відбувається ініціалізація службового класу нашого додатка.

Activity є основою для побудови візуального інтерфейсу Android застосунків, і в цьому випадку розроблено каркас для подальшого використання, який полегшуватиме модифікацію програми надалі, для цього створимо клас BaseActivity, який буде оголошений абстрактним, що означає, що створити на пряму без наслідування буде неможливо (лістинг 3.2).

Лістинг 3.2 – Реалізація базового класу Activity

```
public abstract class BaseActivity extends AppCompatActivity{

@Nullable
@BindView(R.id.toolbar)
Tool bar mToolbar;

@Override
Public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(getLayoutId());
if(mToolbar!=null)
{mToolbar.setNavigationIcon(R.drawable.ic_arrow_back);
setSupportActionBar(mToolbar);
getSupportActionBar().setHomeButtonEnabled(true);
}
```

```
onAction();
}

@Override
Public booleanon  OptionsItemSelected(MenuItem item) {
switch(item.getItemId()) {
case android.R.id.home: {
finish();
return true;
}
}

return super.onOptionsItemSelected(item);
}

protected abstract void onAction();

protected abstract intget LayoutId();
}
```

У відмінність від багатьох додатків Java, програми Android не містять метод main. Натомість у них використовуються чотири типи компонентів, що виконуються – активності (activities), служби (services), провайдери контенту і широкомовні приймачі (broadcast receivers). Програма може мати багато активностей, одна з яких – перше, що ми бачимо під час запуску програми. Користувачі взаємодіють з активностями через уявлення (views) – компоненти GUI, що успадковують від класу View (пакет android.view).

Протягом свого існування активність може бути в одному з кількох станів – активному (тобто виконуваному), зупиненому або зупиненому. Переходи активностей між цими станами відбуваються у відповідь різні події. «Активна активність» відображається на екрані і «володіє фокусом» – тобто взаємодіє з користувачем. Припинена активність помітна на екрані, але не має фокусу (наприклад, на час відображення діалогового вікна з повідомленням). Користувач не може взаємодіяти з призупиненою активністю, доки вона знову не стане активною, наприклад, після того, як користувач закриє діалогове вікно. Зупинена активність не відображається на екрані і, ймовірно, буде знищена системою, коли потрібно звільнити пам'ять,

яку вона займає. Активність зупиняється, коли інша активність перетворюється на активний стан. Наприклад, коли ми відповідаємо на телефонний дзвінок, програма, яка керує дзвінками, стає активною, а попередня програма зупиняється. При переходах активності між цими станами виконавче середовище Android викликає різні методи життєвого циклу (всі ці методи визначаються в класі Activity з android.app). У додатках кожної активності буде перевизначатися метод onCreate. Цей метод викликається виконавчою системою Android при запуску активності, тобто коли її графічний інтерфейс готовий до відображення, щоб користувач міг взаємодіяти з активністю. Також у активностей існують інші методи життєвого циклу: onStart, onPause, onRestart, onResume, onStop і onDestroy.

Кожен метод життєвого циклу активності, що перевизначається вами, повинен викликати версію методу з суперкласу; в іншому випадку відбувається виняток. Виклик версії суперкласу необхідний, тому що кожен метод життєвого циклу в суперкласі Activity містить код, який повинен виконуватися крім коду, який ви визначаєте в перевизначених методах життєвого циклу.

При використанні нових можливостей на попередніх платформах Android розробник стикається з проблемою забезпечення сумісності. Тепер Google відкриває доступ до багатьох нових можливостей Android через Android Support Library – набір бібліотек, завдяки яким розробник може використовувати ці можливості на сучасних та старих платформах Android. Одна з таких бібліотек – AppCompat – забезпечує підтримку панелі програми. Оновлені шаблони програм Android Studio також використовують бібліотеку AppCompat, тому нові програми, які ми створимо, працюватимуть майже на всіх пристроях Android. Шаблон Android Studio Empty Activity визначає клас MainActivity програми як субклас AppCompatActivity (пакет package android.support.v7.app) – непрямого субкласу Activity, що забезпечує використання нових засобів Android на сучасних і старих платформах

Android.

Android також підтримує неявні (implicit) інтенти – розробник не вказує компонент для обробки інтенту. Наприклад, можна створити інтент для відображення вмісту URL-адреси і доручити Android запустити найбільш підходящу активність (браузер) в залежності від типу даних вибирає активність Якщо система не може знайти активність для обробки дії, метод startActivity ініціює виключення ActivityNotFoundException. У загальному випадку рекомендується передбачити обробку цього виключення в програмах.

Для будь-якої Activity викликається метод onCreate, він відповідає за створення вікна. Для того щоб можна було відобразити вид вікна викликається метод setContentView, в який передається числове значення ідентифікатора ресурсу відображення, що отримується від класу нащадка. Тут також встановлюється ActionBar, який є у всіх Activity програми.

Наступний метод onOptionsItemSelected відповідає за обробку натискання меню, яке знаходиться в ActionBar у вигляді стандартної кнопки «Назад». Якщо кнопка була натиснута користувачем, тоді відбувається завершення роботи Activity, що знаходиться на передньому плані. Нижче йдуть два абстрактні методи onAction і getLayoutId, які зобов'язані реалізувати нащадки даного класу, перший метод викликається у нащадка після ініціалізації попередніх елементів методу onCreate, другий при запиті ідентифікатора ресурсу виду.

Наступним ключовим елементом програми є клас ViewFactory, який відповідає за створення класів уявлень програми, які мають інтерфейс BaseView, який уніфікує роботу з уявленнями в додатку.

Фабрика уявлень є реалізацією патерну фабричний метод [19], який дозволяє отримати готовий екземпляр виду без явного інстанцування об'єкта, що дозволяє приховувати спосіб його створення. Так як класи виду реалізують один інтерфейс BaseView ми можемо створювати і повертати їх

без будь-яких проблем, тому що програміст не буде знати з яким саме типом об'єкта він працює, що робить програму більш гнучкою, легко модифікує і розширюється.

Приступимо до розгляду інтерфейсу BaseView, який представляє можливість працювати універсально з класами уявлень (лістинг 3.3).

Лістинг 3.3 –Інтерфейс BaseView

```
public interfaceBaseView{

voidshow(@NonNullfinalContext pContext,@NullablefinalBundle
pArguments);

voidshow(@NonNullfinalContext pContext,@NullablefinalBundle
pArguments,intpRequestCode);

voidshow();voidhide();
voidclose();
}
```

Цей інтерфейс є важливою частиною програми, завдяки якій програму можна буде легко розширювати, додаючи нові уявлення без великих зусиль, і тимчасових витрат, що є важливим для розробки мобільних застосунків.

Клас уявлення SplashActivity успадковує від BaseActivity, але з реалізує інтерфейс BaseView, оскільки у цій ситуації це потрібно.

У методі onAction створено обробник із затримкою на одну секунду для відображення вікна програми. Після того як мине час, буде виконано фрагмент коду в блоці який виконаний із застосуванням замикань[10] із синтаксису Java 11. Відбудеться запуск головного вікна та закриття поточного. Не менш важливим класом уявлення є BaseViewActivity, який є батьківським для інших уявлень, що реалізують інтерфейс BaseView (лістинг 3.4).

Лістинг 3.4 – Фрагмент базового уявлення

```
public abstract class BaseViewActivity<P extends BasePresenter>
```

```
extends
BaseActivity implements BaseView{

privateP mPr;

@Override
Public void show(@NonNull final Context pContext,@Nullable final
Bundle pArguments) {
Intent intent=new Intent(pContext,getClass());
if(pArguments!=null)intent.putExtras(pArguments);
pContext.startActivity(intent);
}
```

Реалізований метод show даного класу реалізує стандартну логіку відображення Activity в екосистемі Android, метод hide ховає вікно. Додаток, що закриває вікно, але залишається в стеку додатків, onAction викликає абстрактний метод onSetup і далі створює Presenter і викликає його метод onCreate для його ініціалізації, метод pr повертає Presenter.

З цього класу починає працювати користувальницький патерн MVP [11].

MVP – шаблон проектування інтерфейсу користувача, який був розроблений для полегшення автоматичного модульного тестування та покращення поділу відповідальності в презентаційній логіці (відділення логіки від відображення).

Модель (англ. Model) – надає дані для інтерфейсу користувача.

Подання (англ. View) – реалізує відображення даних (Моделі) і маршрутизацію команд або подій Presenter'у.

Презентер управляє Моделлю та Поданням. Наприклад, витягує дані з Моделі та форматує їх для відображення у Поданні.

MainActivity є відправною точкою програми, звідси можна відкрити всі уявлення програми. Спочатку створюється ліве меню програми, далі встановлюється кнопка відкриття меню.

Користувач може відкрити меню двома способами:

- натиснути кнопку меню;

- свайпом праворуч.

Відкриється меню і можна буде вибрати пункт меню, після чого у тому вікні відкриється інший розділ.

Настав час розпочати створення розділів програми, які базуються на новому для Android компоненті під назвою Fragment. Фрагмент (клас Fragment) представляє поведінку чи частину інтерфейсу користувача в Activity. Розробник може об'єднати кілька фрагментів в одну Activity для побудови багатопанельного інтерфейсу користувача і повторного використання фрагмента в декількох Activity. Фрагмент можна як модульну частину Activity. Така частина має свій життєвий цикл та самостійно обробляє події введення. Крім того, її можна додати або видалити безпосередньо під час виконання Activity. Це щось подібне до вкладеної Activity, яку можна багаторазово використовувати в різних Activity.

Кожен фрагмент, як і активність, має свій життєвий цикл та надає методи, які можна перевизначати для обробки подій життєвого циклу. У цьому додатку буде перевизначено метод onCreateView. Цей метод викликається після onCreate; він повинен побудувати та повернути об'єкт View з графічним інтерфейсом фрагмента. Як ми невдовзі побачимо, він отримує об'єкт LayoutInflater, який використовується для програмного заповнення графічного інтерфейсу фрагмента за компонентами, заданими в заздалегідь граничному макеті у форматі XML.

Фрагмент завжди має бути вбудований в Activity, і на його життєвий цикл впливає життєвий цикл Activity. Наприклад, коли операцію призупинено, в тому ж стані знаходяться всі фрагменти всередині неї, а коли Activity знищується, знищуються і всі фрагменти. Однак, поки Activity виконується (це відповідає стану відновлення життєвого циклу), можна маніпулювати кожним фрагментом незалежно, наприклад, додавати або видаляти їх. Коли розробник виконує такі транзакції з фрагментами, він може додати їх у стек переходів назад, яким керує операція. Кожен елемент стеку

переходів назад в операції є записом виконаної транзакції з фрагментом.

Батьківська активність використовує для керування своїми фрагментами об'єкт FragmentManager (пакет android.app), який повертається методом getFragmentManager класу Activity. Якщо активності потрібно взаємодіяти з фрагментом, який оголошений у макеті активності та має ідентифікатор id, то для отримання посилання на заданий фрагмент активність може викликати метод findFragmentById класу FragmentManager. FragmentManager може використовувати об'єкти FragmentTransaction для динамічного додавання, видалення та перемикання між фрагментами.

В інтернет провайдері створюється сервіс для доступу до даних в інтернеті за допомогою бібліотеки Retrofit із застосуванням Gson та RxJava, також проводиться логування операцій, та застосування нового Http клієнта OkHttp, який підтримує безпечну передачу даних у мережі.

Створимо графічний інтерфейс користувача для програми. Макетний редактор (Layout Editor) дозволяє створити графічний інтерфейс користувача шляхом перетягування у вікно програми компонентів GUI, таких як Button, TextView, ImageView та ін. За замовчуванням опис макета для шаблону Empty App зберігається у XML-файлі з ім'ям activity_main. xml у папці res/layout. Було використано макетні редактори та вікна Component Tree для побудови програми. Розмітка XML у файлі activity_main.xml буде відредагована лише для того, щоб змінити спосіб розміщення компонентів TextView та ImageView, які використовуються у програмі.

Так як екрани пристроїв Android мають різні розміри, роздільну здатність і щільність пікселів (DPI, Dot Per Inch), розробник зазвичай надає зображення з різними дозволами, а операційна система вибирає графіку на підставі щільності пікселів пристрою. З цієї причини папка res проекту містить кілька вкладених папок, імена яких починаються з префікса drawable. Наприклад, зображення для пристроїв із щільністю пікселів, близьких до щільності екрана телефону Google Nexus 6 (560 dpi) для нашого AVD,

зберігатимуться в папці drawable-xxxhdpi.

Завдання розмірів у пікселах, незалежних від щільності (dp або dip), дозволяє платформі Android автоматично масштабувати графічний інтерфейс користувача в залежності від щільності пікселів екрана фізичного пристрою. 240 dpi розмір пікселя, незалежного від щільності, масштабуватиметься з коефіцієнтом 240/160 (тобто 1,5). Таким чином, компонент, розмір якого становить 100 пікселів, незалежних від щільності, буде масштабований до розміру 150 фізичних пікселів на такому екрані. На екрані з роздільною здатністю 120 пікселів на дюйм кожен незалежний від щільності піксел масштабується з коефіцієнтом 120/160 (тобто 0,75). Отже, 100 незалежних від щільностей пікселів перетворяться на такому екрані на 75 фізичних пікселів. Піксели, незалежні від масштабування, масштабуються так само, як і пікселі, незалежні від щільності, але їх масштаб залежить також і від розміру шрифту, що вибирається користувачем (в налаштуваннях пристрою), (лістинг 3.5).

Лістинг 3.5 – Верстка MainActivity

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayoutandroid:id="@+id/drawer_l
ayout"xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-
auto"android:layout_width="match_parent"
android:layout_height="match_parent">

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

<includelayout="@layout/view_toolbar"/>

<includelayout="@layout/view_container"/>

</LinearLayout>

<android.support.design.widget.NavigationViewandroid:id="@+id/na
v_view" android:layout_width="wrap_content"
android:layout_height="match_parent"
```

```
android:layout_gravity="start"
app:headerLayout="@layout/view_drawer_header" app:menu="@/>

</android.support.v4.widget.DrawerLayout>
```

У коді верстки використовуються вкладені елементи, завжди має бути кореневий елемент. У даному коді тулбар і контейнер з контентом вкладені в лінійний контейнер, нижче якого знаходиться меню, але насправді він знаходиться ліворуч, це відбувається, тому що все це всередині спеціального контейнера, який візуально переміщає його в собі (рисунок 3.5).

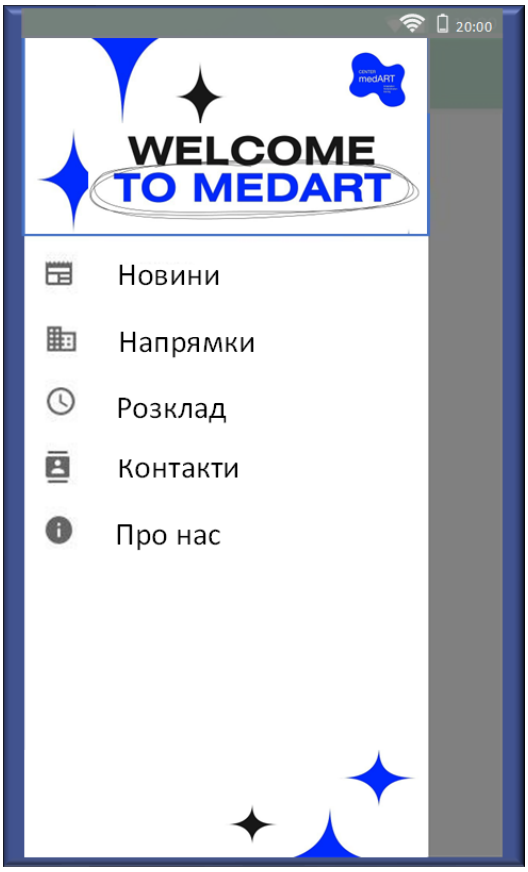

Рисунок 3.5 – Верстка MainActivity

Було створено простий графічний інтерфейс у макетному редакторі та налаштували властивості компонентів у вікні властивостей. У XML-файлі розмітки компонент RelativeLayout, що використовується за умовчанням, був замінений компонентом LinearLayout, який потім був налаштований для

вертикального розташування уявлень. Програма виводить текст у компоненті TextView, а зображення – у компоненті ImageView. Змінено компонент TextView графічного інтерфейсу за промовчанням, щоб текст вирівнювався по центру, збільшеним шрифтом та в одному з кольорів стандартної теми. Компоненти ImageView перетягувалися мишею з палітри компонентів макетного редактора.

У класі MainActivity задіяно багато засобів об'єктно-орієнтованого програмування мовою Java, включаючи класи, об'єкти, інтерфейси, анонімні внутрішні класи та успадкування. Також було представлено концепцію заповнення графічного інтерфейсу, тобто перетворення вмісту файлу XML на його екранне представлення. Було перевизначено метод onCreate для ініціалізації програми під час запуску. У методі onCreate метод findViewById класу Activity використовувався для отримання посилань на візуальні компоненти, з якими програма взаємодіє на програмному рівні. Відредагувано файл AndroidManifest.xml, щоб вказати, що MainActivity підтримує лише портретну орієнтацію, а клас MainActivity завжди має відображати віртуальну клавіатуру. Також були представлені інші елементи, включені Android Studio в маніфест при створенні проекту. Показано, як за допомогою об'єкта Configuration визначити, чи виконується програма на планшеті в альбомній орієнтації. Також у цьому розділі було показано, як керувати великою кількістю графічних ресурсів з використанням вкладених папок у папці assets програми, та як зверталися до цих ресурсів через AssetManager. Розглянуто інші папки з папки res програми – menu для зберігання файлів ресурсів меню, xml для зберігання файлів з розміткою XML. Показано, як використовувати ресурс списку кольорів станів, щоб гарантувати зручність читання тексту на кнопках як для доступного, так і для заблокованого стану.

ВИСНОВКИ

Метою кваліфікаційної роботи є створення мобільного застосунку для розвиваючого медичного центру, для більш ефективного залучення нових клієнтів (пацієнтів), підвищення поінформованості існуючих клієнтів про роботу установи, а також закріплення практичних навичок програмування для ОС Android у середовищі розробки Android Studio.

Для досягнення поставленої мети було сформульовано такі завдання:

- здійснити пошук та аналіз способів підвищення потоку клієнтів за допомогою мобільного застосунку;
- здійснити аналіз вимог до мобільного застосунку;
- спроектувати інтерфейс користувача;
- створити мобільний застосунок.

У першому розділі кваліфікаційної роботи проаналізовано стан сучасної літератури та ефективності створення мобільного застосунку.

У другому розділі був проведений аналіз і пред'явлені вимоги до мобільного застосунку, проаналізовано функціональні вимоги до інтерфейсу.

У третьому розділі за вимогами було створено прототип інтерфейсу користувача. За допомогою, якого надалі проводилася розробка мінімальної версії мобільного застосунку для ОС Android із застосуванням новітніх технологій у галузі мобільних технологій. При вирішенні задачі були вивчені теоретично методи пред'явлення та аналізу вимог до програмного забезпечення, інтерфейсу користувача. Отримано практичну навичку роботи з графічним редактором Adobe Photoshop, в якому було розроблено інтерфейс користувача, що відповідає вимогам стандартів. Було закріплено та отримано нові навички роботи в середовищі розробки Android Studio.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Steel Z., Marnane C., Iranpour C., Chey T., Jackson J. W., Patel V., Silove D. Is the global prevalence rate of adult mental illness increasing? Systematic review and meta-analysis // Psychological Medicine. – 2019. – Т. 49, № 11. – С. 1952–1966. – DOI: 10.1017/S0033291719000027.

2. National Mental Health Commission. The Fifth National Mental Health and Suicide Prevention Plan. – Australian Government, 2017. – [Електронний ресурс]. – Режим доступу: https://www.mentalhealthcommission.gov.au/sites/default/files/2024-03/the-fifth-national-mental-health-and-suicide-prevention-plan-2017.pdf.

3. Психологічна підтримка військових: Міноборони створює реєстр кваліфікованих фахівців // Міністерство оборони України. – [Електронний ресурс]. – Режим доступу: https://mod.gov.ua/news/psihologichna-pidtrimka-vijskovih-minoboroni-stvoryuye-reyestr-kvalifikovanih-fahivcziv.

4. Офіційний сайт компанії Melty. – [Електронний ресурс]. – Режим доступу: http://melty.com.ua/.

5. Wishnya.tech – технологічна платформа. – [Електронний ресурс]. – Режим доступу: https://wishnya.tech/.

6. «Розкажи мені»: в Україні запустили платформу психологічної підтримки // UNN. – [Електронний ресурс]. – Режим доступу: https://unn.ua/news/rozkazhi-meni-v-ukrayini-zapustili-platformu-psikhologichnoyi-pidtrimki.

7. Resilience Hub – платформа психологічної підтримки. – [Електронний ресурс]. – Режим доступу: https://resiliencehub.com.ua/.

8. Spokiy – платформа психологічної підтримки (Україна). – [Електронний ресурс]. – Режим доступу: https://www.spokiy.ee/.

9. ElectroIQ. Android statistics [Електронний ресурс]. – Режим

доступу: https://electroiq.com/stats/android-statistics/ (дата звернення: 01.06.2025).

10. AndroidAuthority. Google Phones: From Nexus to Pixel – A complete history [Електронний ресурс]. – Режим доступу: https://www.androidauthority.com/google-phones-nexus-pixel-history-3260339/ (дата звернення: 01.06.2025).

11. MedArt. Простір сучасної психології [Електронний ресурс]. – Режим доступу: https://medart.space/ (дата звернення: 01.06.2025).