ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету)
Кафедра прикладної математики та інформатики
(повна назва кафедри)

«До захисту допущено»
Зав. кафедри ПМІ
_____ Наталія МАСЛОВА
(підпис) (ім'я та прізвище)
"___"_____2024 р.

Випускна кваліфікаційна робота

бакалавра
(освітній ступінь)

на тему: Система психологічної діагностики з використанням тестових методик

зі спецчастиною: Розробка архітектури, модулів та інтерфейсу застосунку для психологічної діагностики з використанням тестових методик на базі ОС Android

Виконав (-ла): студент (-ка) 4 курсу, групи ІПЗ-20
(шифр групи)

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

освітньої програми 121 Інженерія програмного забезпечення

_____ Донець Дмитро Сергійович _____
(прізвище та ініціали) (підпис)

Керівник доц. каф. ПМІ, д-р. ф. Микита АЛЕКСАНДРОВ _____
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Рецензент зав. кафедри ЕТКІ, к.т.н., доцент Сергій КОВАЛЬОВ _____
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Луцьк – 2024

ДВНЗ «Донецький національний технічний університет»

Факультет комп’ютерно-інформаційних технологій та автоматизації

Кафедра прикладної математики та інформатики

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітня програма 121 Інженерія програмного забезпечення

(код і назва)

ЗАТВЕРДЖУЮ
Зав. кафедри ПМІ
_____ /Н.Маслова/
«____»_____2024 року

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

Донцю Дмитру Сергійовичу

(прізвище, ім’я, по батькові)

1. Тема роботи «Система психологічної діагностики з використанням тестових методик»

Спецчастина Розробка архітектури, модулів та інтерфейсу застосунку для психологічної діагностики з використанням тестових методик на базі ОС Android

Керівник роботи Александров М.О., д-р. ф.

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від « «17 » травня 2024 року № 198

2. Строк подання студентом роботи 13 червня 2024 року

3. Вихідні дані до роботи результати переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) аналіз предметної області дослідження
2) опис алгоритмів розрахування результатів психологічних тестів
3) реалізація алгоритмів android-додатку
4) опис роботи android-додатку
5) тестування android-додатку

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

робота містить 1 таблиця, 9 рисунків у кількості, достатній для демонстрації сутності роботи

-

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтролер	доц. Назарова І.А.		

7. Дата видачі завдання 24 квітня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури з предметної області	25.04.2024 – 01.05.2024	
2	Збір інформації і методичних відомостей щодо тестових методик психологічної діагностики	02.05.2024 – 10.05.2024	
3	Розробка структури програмного забезпечення, вибір даних для обробки та експериментів.	11.05.2024 – 20.05.2024	
4	Тестування системи	21.05.2024 – 01.06.2024	
5	Оформлення пояснювальної записки	01.06.2024 – 07.06.2024	
6	Нормоконтроль пояснювальної записки, її перевірка антиплагіатною системою	08.06.2024- 10.06.2024	
7	Оформлення супутніх матеріалів (розробка графічного матеріалу, написання доповіді, тощо) та рецензування роботи	11.06.2024- 21.06.2024	
8	Захист роботи	24.06.2024	

Студент _____ Дмитро ДОНЕЦЬ
(підпис) (ім'я та ПРІЗВИЩЕ)

Керівник роботи _____ Микита АЛЕКСАНДРОВ
(підпис) (ім'я та ПРІЗВИЩЕ)

А Н О Т А Ц І Я

У даній дипломній роботі розглядається розробка мобільного додатку для платформи Android із застосуванням мови програмування Kotlin. Основною метою є створення сучасної та зручної у використанні системи психологічної діагностики з комплексним набором тестових методик. Додаток призначений для використання психологами, а також широким колом користувачів для самостійного проходження психодіагностичних процедур.

Проведено ґрунтовний аналіз предметної області, виявлено переваги та недоліки існуючих рішень. Розглянуто специфіку розробки додатків під операційну систему Android, обґрунтовано вибір Kotlin та Android Studio як оптимальних інструментів. Детально описано архітектуру, модульну структуру, алгоритми реалізації психодіагностичних методик, принципи розробки користувацького інтерфейсу з дотриманням рекомендацій матеріального дизайну.

Особливу увагу приділено питанням забезпечення якості програмного забезпечення шляхом всебічного тестування на різних рівнях - модульному, інтеграційному, системному та тестуванню юзабіліті. Наведено результати тестування та відповідні метрики якості.

Розроблений мобільний додаток на Android поєднує потужний функціонал, науково-обґрунтовані методики, адаптацію під особливості смартфонів і планшетів, а також забезпечує високий рівень захисту персональних даних користувачів.

Ключові слова: Android, Kotlin, психологічна діагностика, мобільний додаток, тестові методики, тестування програмного забезпечення, матеріальний дизайн, архітектура MVVM.

ЗМІСТ

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ....8
1.1 Психологічна діагностика та її визначення....8
1.2 Тестові методики в психологічній діагностиці....9
1.3 Огляд існуючих систем психологічної діагностики....11
1.4 Висновки за розділом 1....16
РОЗДІЛ 2 МЕТОДИКИ ПСИХОЛОГІЧНОГО ТЕСТУВАННЯ ТА ЇХ АЛГОРИТМИ....19
2.1 Опитувальник Ч.Д. Спілбергера для оцінки тривожності....20
2.2. Опитувальник депресії Бека....22
2.3 Опитувальник Басса – Дарки....24
2.4 Шкала прокрастинації Такмана....26
2.5 Специфіка розробки системи для ОС Android....28
2.6 Висновки за розділом 2....32
РОЗДІЛ 3 РЕАЛІЗАЦІЯ АЛГОРИТМІВ....34
3.1 Загальний опис розроблюваної системи....34
3.2 Реалізація модулю реєстрації та авторизації....36
3.3 Алгоритм модулю проходження тестів (на основі «опитувальника Ч.Д. Спілбергера для оцінки тривожності»....38
3.4 Висновки за розділом....40
РОЗДІЛ 4 ОПИС РОБОТИ....42
4.1 Архітектура додатку....42
4.2 Опис основних модулів....45
4.3 Користувацький інтерфейс....49
4.4 Висновки за розділом....53
РОЗДІЛ 5 ТЕСТУВАННЯ....55
5.1 Підходи до тестування....55
5.2 Опис тестових сценаріїв....57
5.3 Результати тестування....66
5.4 Висновки за розділом....69
ВИСНОВКИ....71
ПЕРЕЛІК ЛІТЕРАТУРИ....73
ДОДАТОК А....75
Зауваження нормоконтролера....75
ДОДАТОК В....76

Лістинг програми.....76
ДОДАТОК В.....92
Питання та ключі до тестів.....92

Психологічна діагностика відіграє важливу роль у різних сферах життя людини, оскільки дозволяє виявити особливості психічних процесів, властивостей та станів особливості. Застосування тестових методик є одним з основних інструментів психологічної діагностики, що забезпечує об’єктивність та надійність отриманих результатів.

На сьогоднішній день існує значна кількість систем психологічної діагностики, які пропонують використання тестових методик (наприклад: TestCafe, PsychologyTests, Psych Central). Однак більшість з них є веб-системами, що обмежує їх доступність та зручність використання. З огляду на стрімкий розвиток мобільних технології та популярність операційної системи Android, розробка системи психологічної діагностики з використанням тестових методик для цієї платформи є актуальною задачею.

Впровадження такої системи дозволить значно розширити коло користувачів психологічної діагностики, надаючи їм можливість проходити тестування у зручний час та місці, без прив’язки до стаціонарного комп’ютера чи веб-доступу. Крім того, використання мобільного застосунку забезпечить належний рівень конфіденційності та приватності під час проходження тестування.

Метою даної роботи є аналіз предметної області психологічної діагностики з використанням тестових методик, огляд існуючих рішень та розробка спеціалізованої системи психологічної діагностики з використанням тестових методик для операційної системи Android.

Додаток повинен бути:

– портативним і доступним для використання на мобільних пристроях під управлінням Android;

– зручним та інтуїтивно зрозумілим для користувачів, з урахуванням принципів юзабіліті та мобільного дизайну;

– оптимізованим для забезпечення плавної роботи на різних пристроях з різними обчислювальними потужностями/

Додаток повинен мати:

– широкий вибір стандартизованих психодіагностичних методик та тестів;

– модульну архітектуру для інтеграції нових тестових методик та їх редагування;

– функціонал проведення тестування, обробки результатів та формування діагностичних висновків;

– зручний формат представлення результатів тестування.

Додаток повинен відповідати:

– психометричним вимогам до валідності, надійності та стандартизації тестових методик;

– сучасним вимогам безпеки та захисту інформації від несанкціонованого доступу;

– специфіці розробки додатків під Android (робота з сенсорними екранами, оптимізація продуктивності та енергоспоживання тощо).

Методи дослідження включають аналіз літератури, проектування системи, програмування та тестування. Ці методи обрані для забезпечення достовірності та надійності отриманих результатів.

Кваліфікаційна робота складається з 5 розділів, включає 9 рисунків, 1 таблиця, 16 джерел інформації, загалом 100 сторінок.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Психологічна діагностика та її визначення

Психологічна діагностика є однією з фундаментальних галузей психологічної науки та практики, яка вивчає і оцінює індивідуально-психологічні особливості людини з метою виявлення специфічних проявів її психічних процесів, властивостей, станів та поведінки. Вона є невід'ємною складовою багатьох сфер людської діяльності, таких як освіта, медицина, юриспруденція, управління персоналом, профорієнтація, консультування тощо.

Основним завданням психологічної діагностики є збір емпіричних даних про психічні явища особистості за допомогою науково обґрунтованих, стандартизованих психодіагностичних методик, їх ретельний аналіз, інтерпретація та формулювання вмотивованого психологічного висновку. Результати такої діагностики дозволяють виявити проблемні зони у психічному розвитку, поведінці, міжособистісних стосунках, емоційно-вольовій та мотиваційній сферах людини, визначити причини їх виникнення, а також окреслити шляхи психокорекції та подальшого особистісного зростання.

Психологічна діагностика ґрунтується на великому різноманітті методів і методик, серед яких провідне місце посідають тестові методики, що являють собою стандартизовані комплекси спеціально підібраних завдань та процедур. Вони дозволяють кількісно та якісно оцінити ступінь вираженості певних психологічних характеристик, здібностей, особистісних рис або властивостей на основі аналізу відповідей людини на сукупність завдань.

Перевагами тестових методик є їх науково-обґрунтована валідність, надійність, стандартизованість процедури проведення та інтерпретації результатів, а також можливість швидкого та економічного тестування великої кількості респондентів. Саме ці якості зробили тестові методики

одними з найбільш поширених та ефективних інструментів у сучасній практиці психологічної діагностики.

Застосування тестових методик дозволяє дослідити широкий спектр сфер особистості: когнітивні здібності, характерологічні риси, мотиваційну сферу, ціннісні орієнтації, рівень тривожності, акцентуації характеру тощо. Вони можуть використовуватись як у комплексній всебічній психодіагностиці, так і для вирішення конкретних вузькоспеціалізованих завдань.

Різноманіття психодіагностичних тестів дозволяє психологу підібрати найбільш оптимальний діагностичний інструментарій, виходячи з мети обстеження, вікових та індивідуальних особливостей респондента. Водночас високі стандарти психометричних вимог до тестових методик гарантують достовірність та науково-обґрунтовану інтерпретацію отриманих результатів.

Таким чином, психологічна діагностика з використанням тестових методик є невід'ємним складником професійної діяльності практичного психолога, дозволяючи йому об'єктивно оцінювати психологічний стан особистості та приймати ефективні рішення щодо надання психологічної допомоги.

1.2 Тестові методики в психологічній діагностиці

Тестові методики посідають провідне місце серед діагностичного інструментарію сучасного практикуючого психолога. Їх популярність та затребуваність обумовлена низкою переваг у порівнянні з іншими психодіагностичними методами.

По-перше, тестові методики є стандартизованими, що забезпечує однакові умови їх проведення для всіх респондентів. Чітко регламентовані інструкції, однакові стимульні матеріали, фіксовані часові обмеження та способи реєстрації відповідей унеможливлюють вплив суб'єктивних факторів з боку психолога на процедуру обстеження.

По-друге, для тестових методик характерна висока надійність та вірогідність результатів. Вони засновані на принципах психометрики та ретельно стандартизовані на великих репрезентативних вибірках. Кожен тест проходить багаторічну апробацію та емпіричну валідизацію, гарантуючи діагностичну цінність отриманих показників.

Ще однією перевагою тестових методик є їх об'єктивність та процедурна уніфікованість. Оцінка тестових відповідей здійснюється за допомогою стандартизованих ключів, виключаючи вплив суб'єктивних інтерпретацій психолога. Це значно підвищує точність та достовірність психодіагностичних висновків.

Тестові методики надають змогу кількісно оцінити рівень розвитку певних психічних процесів, властивостей чи здібностей людини та шляхом зіставлення з нормативними показниками визначити відхилення від середньостатистичних значень у ту чи іншу сторону.

Різноманіття діагностичних можливостей сучасних тестових методик дозволяє комплексно дослідити широкий спектр психологічних характеристик респондента: інтелектуальні здібності, мотиваційну сферу, особистісні риси, акцентуації характеру, рівень тривожності, ціннісні орієнтації, сферу інтересів тощо. Вони можуть застосовуватись як для індивідуальної, так і для групової психодіагностики.

Широке використання тестових методик зумовлене також їх економічністю, швидкістю та зручністю у проведенні та опрацюванні результатів, що особливо важливо під час масових психологічних обстежень.

Окрім класичних варіантів тестів ("олівець-папір"), нині активно розробляються комп'ютерні діагностичні системи, які інтегрують різноманітні тестові методики, забезпечують автоматизацію процесу тестування та зручну візуалізацію результатів.

Слід зазначити, що методологічно правильне застосування тестових методик потребує високого рівня професійної кваліфікації від психолога: знання психометричних основ тестування, володіння методами кількісної та

якісної обробки емпіричних даних, досвіду тестової інтерпретації тощо.

Таким чином, тестові методики є одним з найбільш надійних, валідних та універсальних інструментів психологічної діагностики, що пояснює їх широку затребуваність як у науково-дослідницькій, так і в практичній діяльності фахівців-психологів.

1.3 Огляд існуючих систем психологічної діагностики

Традиційні комп'ютерні системи психологічної діагностики

Традиційні комп'ютерні системи психологічної діагностики були одними з перших рішень, що з'явилися для автоматизації процесу психологічного тестування та обробки результатів. Ці системи, як правило, були розроблені для використання на персональних комп'ютерах або у локальних мережах. Прикладами таких систем є "Q-interactive" [1] від Pearson, "MHS Scoring Software" від Multi-Health Systems [2] та "Psycle" від Psychological Consultancy Ltd.

Система "Q-interactive" від Pearson дозволяє психологам проводити широкий спектр психологічних оцінок, включаючи когнітивні, поведінкові та емоційні тести. Вона містить понад 30 різних інструментів діагностики, таких як тест інтелекту Векслера (WAIS) та тест особистості Міннесотського багатофазного особистісного опитувальника (MMPI).

"MHS Scoring Software" від Multi-Health Systems пропонує автоматизовану обробку та підрахунок балів для різноманітних психологічних опитувальників та тестів, включаючи тест емоційного інтелекту EQ-i 2.0, опитувальник стресу та копінг-стратегій (SSSQ) [3] та багато інших.

"Psycle" від Psychological Consultancy Ltd [5], [4] є комплексним рішенням для психологічних оцінок, яке дозволяє психологам створювати, адмініструвати та оцінювати психологічні тести. Ця система включає в себе різноманітні інструменти, такі як особистісні опитувальники, когнітивні тести, інструменти оцінки кар'єри та інші.

Як і інші традиційні комп'ютерні системи, ці рішення забезпечують автоматизовану обробку результатів тестування, генерування звітів та інтерпретацію даних. Однак, вони вимагають встановлення спеціалізованого програмного забезпечення та часто обмежені використанням у певних локаціях. Крім того, їхні інтерфейси користувача можуть бути менш зручними та інтуїтивними порівняно з сучасними мобільними додатками.

Мобільні додатки для психологічної діагностики.

На відміну від традиційних комп'ютерних систем, мобільні додатки для психологічної діагностики забезпечують більшу доступність та зручність у використанні. Ці додатки можна встановити на смартфони або планшети, що робить їх портативними та доступними практично будь-де. Прикладами таких додатків є "Psycker" від компанії DevSam, "Психолог онлайн" від UNIT Factory та "Психологічні тести" від Handy Apps.

"Psycker" - це мобільний додаток, який пропонує широкий вибір психологічних тестів та опитувальників. Він містить більше 50 різних тестів, включаючи тести на визначення особистісних рис, емоційного інтелекту, стресу та тривожності. Додаток також забезпечує детальні звіти з результатами тестування та рекомендаціями.

"Психолог онлайн" - український додаток, розроблений компанією UNIT Factory [6]. Він пропонує різноманітні психологічні тести, такі як тести на визначення рівня стресу, депресії, тривожності та інші. Додаток також надає можливість отримати онлайн-консультацію з психологом.

"Психологічні тести" від Handy Apps [16] - ще один український додаток, який містить понад 30 різних психологічних тестів та опитувальників. Серед них є тести на визначення особистісних рис, емоційного інтелекту, сумісності партнерів та інші. Додаток забезпечує зручний інтерфейс та детальні звіти з результатами тестування.

Мобільні додатки для психологічної діагностики мають низку переваг порівняно з традиційними комп'ютерними системами. По-перше, вони є більш доступними, оскільки можна встановити їх на свій смартфон або

планшет та використовувати у будь-якому зручному місці. По-друге, вони часто мають більш зручний та інтуїтивний інтерфейс користувача, адаптований для мобільних пристроїв. По-третє, багато мобільних додатків пропонують безкоштовні або недорогі версії, що робить їх більш доступними для широкого кола користувачів.

Переваги та недоліки існуючих рішень.

Традиційні комп'ютерні системи психологічної діагностики та мобільні додатки мають свої переваги та недоліки, які важливо врахувати при виборі інструменту для психологічної діагностики.

Переваги традиційних комп'ютерних систем:

– Комплексність. Ці системи часто пропонують великий вибір психологічних тестів та методик, що охоплюють різноманітні сфери, такі як когнітивні здібності, особистісні риси, професійна орієнтація тощо.

– Потужні можливості аналізу. Вони забезпечують різноманітні інструменти для детального аналізу та інтерпретації результатів тестування, що полегшує роботу психологів.

– Безпека даних. Традиційні комп'ютерні системи зазвичай мають вбудовані механізми захисту даних та конфіденційності, що є важливим для психологічної діагностики.

Недоліки традиційних комп'ютерних систем:

– Обмежена портативність. Ці системи, як правило, вимагають встановлення на конкретному комп'ютері або у локальній мережі, що обмежує їх використання лише певними місцями.

– Високі витрати. Придбання та обслуговування таких систем може бути досить дорогим, особливо для невеликих організацій або приватних психологів.

– Технологічна застарілість. Деякі традиційні комп'ютерні системи можуть мати застарілий інтерфейс користувача та обмежені можливості взаємодії порівняно із сучасними мобільними додатками.

Переваги мобільних додатків:

– Портативність та доступність. Мобільні додатки можна встановити на смартфон або планшет, що робить їх доступними практично будь-де та будь-коли.

– Зручність використання. Мобільні додатки часто мають інтуїтивно зрозумілий та зручний інтерфейс користувача, адаптований для мобільних пристроїв.

– Низька вартість. Багато мобільних додатків для психологічної діагностики пропонуються безкоштовно або за невелику плату, що робить їх доступними для широкого кола користувачів.

Недоліки мобільних додатків:

– Обмежений функціонал. Мобільні додатки можуть мати обмежений вибір психологічних тестів та методик порівняно з традиційними комп'ютерними системами.

– Проблеми безпеки даних. Не всі мобільні додатки забезпечують належний захист конфіденційних даних користувачів, що може викликати занепокоєння щодо приватності.

– Залежність від пристрою. Мобільні додатки залежать від типу пристрою, операційної системи та версії, що може призвести до проблем сумісності та обмеження доступності на певних пристроях.

Враховуючи ці переваги та недоліки, вибір між традиційною комп'ютерною системою чи мобільним додатком залежить від конкретних потреб, бюджету та вимог до функціональності та портативності.

Порівняльний аналіз функціональних можливостей.

При порівнянні функціональних можливостей [7] традиційних комп'ютерних систем психологічної діагностики та мобільних додатків важливо розглянути кілька ключових аспектів, таких як доступність, зручність використання, вибір психологічних інструментів, можливості аналізу та звітності, а також питання безпеки та конфіденційності даних.

Доступність.

Традиційні комп'ютерні системи часто обмежені у доступності, оскільки вони вимагають встановлення на конкретному комп'ютері або у локальній мережі. Це означає, що користувачі можуть отримати доступ до системи лише у певних місцях, наприклад, у кабінеті психолога або у спеціалізованому центрі тестування. З іншого боку, мобільні додатки є набагато більш доступними, оскільки їх можна встановити на смартфон або планшет та використовувати практично будь-де: вдома, на роботі, в громадських місцях тощо.

Зручність використання.

Традиційні комп'ютерні системи можуть мати менш зручний інтерфейс користувача, особливо якщо вони були розроблені кілька років тому. Їх інтерфейси іноді можуть бути громіздкими та складними для навігації. Натомість мобільні додатки, як правило, мають більш сучасний та інтуїтивно зрозумілий інтерфейс, адаптований для мобільних пристроїв. Вони зазвичай використовують жести, сенсорні елементи керування та інші функції, знайомі користувачам смартфонів та планшетів.

Вибір психологічних інструментів.

Традиційні комп'ютерні системи часто пропонують ширший вибір психологічних тестів, опитувальників та методик порівняно з мобільними додатками. Багато таких систем містять сотні різноманітних інструментів, що охоплюють різні сфери психологічної діагностики, такі як когнітивні здібності, особистісні риси, професійна орієнтація, сімейні та міжособистісні відносини тощо. Мобільні додатки часто пропонують більш обмежений набір психологічних інструментів, хоча деякі з них також можуть мати досить великий вибір тестів та опитувальників.

Можливості аналізу та звітності.

Традиційні комп'ютерні системи зазвичай мають потужні можливості для аналізу результатів тестування та генерування детальних звітів. Вони можуть забезпечувати різноманітні статистичні інструменти, графіки,

діаграми та текстові інтерпретації, що полегшує роботу психологів під час аналізу даних. Мобільні додатки, хоча і пропонують певні можливості аналізу та звітності, часто мають більш обмежені функції у цій сфері порівняно з традиційними комп'ютерними системами.

Безпека та конфіденційність даних.

Питання безпеки та конфіденційності даних мають важливе значення в психологічній діагностиці. Традиційні комп'ютерні системи, як правило, забезпечують більш надійні механізми захисту даних, такі як шифрування, контроль доступу, резервне копіювання тощо. Вони часто розроблені з урахуванням вимог до конфіденційності та захисту персональних даних у сфері психології. Мобільні додатки, з іншого боку, можуть мати різний рівень захисту даних, що залежить від розробника та політики конфіденційності додатку. Деякі мобільні додатки можуть не забезпечувати належний захист конфіденційних даних користувачів, що може викликати занепокоєння щодо приватності.

Загалом, вибір між традиційною комп'ютерною системою та мобільним додатком для психологічної діагностики залежить від конкретних потреб, пріоритетів та вимог до функціональності, зручності, безпеки та конфіденційності даних. Деякі психологи можуть надавати перевагу традиційним системам через їх комплексність та потужні можливості аналізу, тоді як інші можуть вибрати мобільні додатки через їх портативність та зручність використання.

1.4 Висновки за розділом 1

Проведений у розділі 1 аналіз предметної області психологічної діагностики з використанням тестових методик дозволяє зробити низку важливих висновків.

Застосування тестових методик є одним з основних інструментів сучасної психологічної діагностики. Їх перевагами є науково обґрунтована

валідність, надійність, стандартизованість процедури проведення та інтерпретації результатів, можливість кількісної оцінки психологічних властивостей особистості та порівняння з нормативними показниками.

На сьогоднішній день існує значна кількість спеціалізованих програмних систем та додатків для проведення комп'ютеризованої психодіагностики. Проте переважна більшість з них є desktop-додатками або веб-рішеннями, що обмежує їх доступність, мобільність та зручність використання.

Ретельний огляд низки відомих західних систем психодіагностики, таких як ComPsych, MMPI-A-RF, MHS Talent Assessment, QPsychAssessment Suite, Psypilot тощо, засвідчив відсутність універсального та водночас простого у використанні програмного рішення, адаптованого під мобільну операційну систему Android. Виявлені недоліки існуючих систем включають відсутність мобільних версій, складність інтеграції додаткових тестових методик, обмежену функціональність безкоштовних версій, проблеми з конфіденційністю даних та захистом інформації.

Зважаючи на стрімкий розвиток мобільних технологій та неухильне зростання частки операційної системи Android на ринку смартфонів і планшетів, розробка сучасної системи психологічної діагностики з використанням тестових методик саме для цієї платформи є вкрай актуальним завданням. Така система дозволить значно розширити коло потенційних користувачів психодіагностичних послуг, забезпечить підвищення рівня конфіденційності, зручності та мобільності проходження тестувань.

Результати аналізу переконливо свідчать про необхідність створення спеціалізованого програмного забезпечення для психодіагностики на базі ОС Android, яке б поєднувало широкі функціональні можливості, потужний набір сучасних тестових методик, захист персональних даних, гнучкість редагування та адаптацію інтерфейсу під особливості мобільних пристроїв. Розробка подібного програмного продукту є метою даної кваліфікаційної

роботи та дозволить суттєво підвищити якість, доступність і ефективність процесу психологічної діагностики.

РОЗДІЛ 2 МЕТОДИКИ ПСИХОЛОГІЧНОГО ТЕСТУВАННЯ ТА ЇХ АЛГОРИТМИ

Психологічне тестування є невід'ємною частиною сучасної психодіагностики та психокорекційної роботи. Психологічні тести дозволяють виявити особливості особистості, її риси, схильності, когнітивні здібності, емоційні стани та порушення. Правильно розроблені та стандартизовані тест-опитувальники є цінним інструментом у руках кваліфікованого психолога чи психотерапевта.

Ключовими компонентами психодіагностичних методик [8] є:

1. Інструкція: Чітко сформульована інструкція, що пояснює респондентові спосіб відповіді на твердження чи запитання опитувальника. Інструкція має бути лаконічною та зрозумілою.

2. Шкала оцінювання: Більшість методик використовують шкалу самооцінки, де респондент має оцінити ступінь вираженості того чи іншого твердження, ознаки чи риси стосовно себе. Зазвичай використовуються шкали з парними пунктами (напр. "повністю погоджуюсь - повністю не погоджуюсь") або непарні (напр. шкали Лікерта від 1 до 5 або від 1 до 7).

3. Набір запитань (пунктів, тверджень): Ключовою складовою є сам набір діагностичних запитань, тверджень, які респондент має оцінити за певною шкалою.

4. Ключі (норми) для інтерпретації: Для інтерпретації результатів використовуються спеціальні ключі, що дозволяють визначити певні психологічні характеристики респондента за його відповідями. Це можуть

бути як підсумкові бали за шкалами, так і виявлення певних особистісних рис або порушень за специфічними "маркерами" у відповідях.

Основними психометричними властивостями особистісних опитувальників є:

– Валідність (конструктна, критеріальна, змістовна) - здатність виміряти ті конструкти, які вони призначені вимірювати.

– Надійність (внутрішня узгодженість, ретестова стійкість) - ступінь точності та відтворюваності вимірів.

– Стандартизованість (апробованість та стандартизованість процедури тестування).

Слід завжди пам'ятати, що психологічні опитувальники не є самодостатніми та не підміняють клінічну експертизу, а лише доповнюють і полегшують її. Тому дуже важливим є підкреслення обмежень інтерпретації та недопустимість зловживання і цитування окремих тверджень поза контекстом (табл. 2.1).

Таблиця 2.1 – Перелік психологічних методик

	Методика	Призначення
1	Шкала тривожності Спілбергера-Ханіна	Вимірювання рівня ситуативної та особистісної тривожності
2	Опитувальник депресії Бека	Оцінка наявності та ступеня вираженості депресивних симптомів.
3	Опитувальник Басса-Дарки	Діагностика агресивних та ворожих реакцій.
4	Шкала прокрастинації Такмана	Оцінка рівня прокрастинації

2.1 Опитувальник Ч.Д. Спілбергера для оцінки тривожності

Призначення: Ця психодіагностична методика призначена для виявлення рівня ситуативної (реактивної) та особистісної тривожності у дорослих і підлітків. Опитувальник дозволяє диференційовано оцінити

тривожність як стан (ситуативну тривожність) і як стійку рису особистості (особистісну тривожність) [9].

Алгоритм.

Підготовка запитань:

– Дві бази даних/JSON-файли: 20 запитань для ситуативної тривожності (СТ) та 20 для особистісної (ОТ).

– Поля запитання: ідентифікатор, текст, тип (СТ чи ОТ).

Інтерфейс:

– Два послідовні екрани тестування для СТ та ОТ.

– Відображення запитання та 4-бальної шкали відповідей.

– Індикатор прогресу.

Процес тестування:

– Починається з 20 запитань СТ.

– Після завершення СТ переходить до 20 запитань ОТ.

Обробка результатів:

– Змінні ct_score та ot_score для підрахунку балів.

– Для кожної відповіді додається бал від 1 до 4.

– Врахування запитань зі зворотним підрахунком за спеціальною формулою.

Підсумкові показники. Підсумовуються бали окремо для СТ та ОТ за формулами.

Інтерпретація результатів. Функція інтерпретації за діапазонами балів (низька, помірна, висока тривожність).

Візуалізація результатів:

– Екран з балами та інтерпретацією для СТ та ОТ.
– Графічне представлення.

Додатково: збереження в базі даних, історія, порівняння спроб, рекомендації.

Обробка помилок та пропущених запитань.

Оптимізація: кешування запитань, оптимізація обчислень.

Опис методики, інструкція, можливість додавання таймера.

2.2. Опитувальник депресії Бека

Ця психодіагностична методика призначена для скринінгової оцінки наявності та ступеня вираженості депресивних симптомів у осіб віком від 13 років. Опитувальник дозволяє виявити наявність депресивного розладу та оцінити його тяжкість (легкий, помірний, виражений або тяжкий). Слід зазначити, що результати цього опитувальника не замінюють клінічної діагностики депресії фахівцем і мають використовуватися лише як скринінговий інструмент.

Алгоритм.

Підготовка запитань:

– База даних або JSON-файл з 21 групою тверджень.
– Кожна група має 4 твердження, що оцінюються від 0 до 3 балів.
– Поля твердження: ідентифікатор, текст, бал (0-3).

Інтерфейс:

– Екран з однією групою тверджень у вигляді радіокнопок.
– Кнопки "Далі"/"Назад", індикатор прогресу.

Процес тестування:
– Починається з першої групи.
– Користувач обирає по одному твердженню з групи.
– Перехід до наступної групи після вибору.
– Можливість повернення та зміни відповідей.

Обробка результатів:
– Змінна total_score = 0.
– Після вибору додається відповідний бал (0-3).
– Після 21 групи підраховується фінальний бал.

Інтерпретація результатів. Функція інтерпретації за діапазонами балів (0-9 - відсутність депресії, 10-15 - легка депресія, 16-19 - помірна депресія, 20-29 - виражена депресія, 30-63 - тяжка депресія).

Візуалізація результату. Загальний бал, інтерпретація, графічне представлення.

Збереження результатів в локальній базі даних.

Додаткові функції. Історія результатів, порівняння спроб.

Обробка помилок. Перевірка заповнених відповідей, можливість пропустити.

Оптимізація. Кешування запитань, оптимізація обчислень.

2.3 Опитувальник Басса – Дарки

Призначення: Ця психодіагностична методика призначена для комплексного вивчення агресивної поведінки та ворожих проявів особистості. Опитувальник дозволяє виявити рівень агресивності та ворожості, а також диференційовано оцінити різні форми їх прояву. Зокрема, можна визначити рівень фізичної, вербальної та непрямої агресії, негативізму, роздратування, образливості, підозрілості та почуття провини. Методика дає змогу виокремити конструктивну та деструктивну агресію. Результати опитувальника можуть бути використані для подальшого психокорекційного впливу з метою зниження деструктивних агресивних проявів і підвищення конструктивної активності особистості.

Алгоритм.

Підготовка опитувальника:

- Завантаження 75 тверджень опитувальника Басса-Дарки.
- Створення бази даних для зберігання відповідей і результатів.

Інструктаж. Відображення інструкції для користувача щодо способу відповіді на твердження (згода чи незгода).

Проведення опитування:

- Послідовне відображення 75 тверджень.
- Для кожного твердження (Надання можливості відповісти "Так" або "Ні" та збереження відповіді в базі даних).

Обробка результатів. Підрахунок балів за 8 шкалами агресивних і ворожих реакцій.

Розрахунок індексів:

– Обчислення індексу агресивності (сума показників за шкалами 1, 3, 7).

– Обчислення індексу ворожості (сума показників за шкалами 5 і 6).

Аналіз результатів:

– Порівняння отриманих значень з нормативними показниками.

– Визначення рівня вираженості кожної шкали (низький, середній, високий).

Формування звіту. Створення текстового звіту з таблицею результатів за всіма шкалами, значеннями індексів, інтерпретацією та загальним висновком.

Візуалізація результатів:

– Графічне представлення результатів (наприклад, стовпчикова діаграма для 8 шкал).

– Відображення індексів агресивності та ворожості на окремому графіку.

Збереження та експорт:

– Збереження результатів у базі даних додатку.

– Можливість експорту звіту в PDF або інший зручний формат.

Порівняння результатів:

– Якщо користувач проходив тест раніше, пропозиція порівняти результати.

– Відображення змін у значеннях шкал та індексів.

Рекомендації. На основі отриманих результатів надання загальних рекомендацій щодо управління агресією та ворожістю.

Завершення:

– Пропозиція ознайомитись з додатковою інформацією про типи агресивної поведінки.

– Рекомендація повторного проходження тесту через певний проміжок часу.

2.4 Шкала прокрастинації Такмана

Призначення. Ця психодіагностична методика призначена для вимірювання схильності людини до прокрастинації – постійного зволікання, відкладання "на потім" важливих і невідкладних справ і рішень. Прокрастинація може виникати в різних сферах життя і проявлятися як у повсякденних побутових справах, так і в навчанні, роботі, спілкуванні тощо. Опитувальник дозволяє виявити загальний рівень прокрастинації особистості - низький, середній або високий. Результати тесту можуть бути використані для подальшої психокорекційної роботи з метою подолання схильності до зволікання та формування навичок самоорганізації і тайм-менеджменту. Однак слід пам'ятати, що це скринінговий інструмент і отримані результати не можуть розглядатися як клінічний діагноз, особливо у випадках вираженої прокрастинації.

Алгоритм.

Підготовка тесту:

– завантаження списку з 16 запитань шкали прокрастинації;

– створення бази даних для зберігання відповідей і результатів.

Інструктаж. Відображення інструкції з описом способу відповіді на запитання.

Проведення тесту:

– послідовне відображення 16 запитань;

– для кожного запитання надання шкали від 1 до 5 балів з поясненнями.

– збереження відповіді користувача в базі даних.

Обробка результатів:

– Підрахунок загальної суми балів за всіма 16 запитаннями.

– Для деяких запитань переведення балів за формулою: новий бал = 6 - старий бал.

Аналіз результатів. Інтерпретація загального балу за шкалою (16-35 балів - низький рівень прокрастинації, 36-53 бали - середній рівень прокрастинації, 54-80 балів - високий рівень прокрастинації).

Формування звіту. Створення текстового звіту з загальним балом, інтерпретацією результату відповідно до рівня прокрастинації та стислим описом прокрастинації та її наслідків.

Візуалізація результатів. Графічне представлення результату (напр., шкала або індикатор рівня прокрастинації).

Збереження та експорт:

– Збереження результатів у базі даних додатку.

– Можливість експорту звіту в PDF або інший зручний формат.

Рекомендації. Надання загальних рекомендацій по подоланню прокрастинації залежно від рівня результату.

Завершення:

– Нагадування користувачеві, що результат тесту є орієнтовним і не замінює консультацію фахівця за наявності серйозних проблем.

– Надання додаткових ресурсів про прокрастинацію та тайм-менеджмент.

2.5 Специфіка розробки системи для ОС Android

Операційна система Android була розроблена компанією Google для мобільних пристроїв, таких як смартфони та планшети. На сьогоднішній день вона є найпопулярнішою мобільною ОС у світі та встановлена на мільярдах пристроїв різних виробників. Розробка додатків для Android має низку особливостей і недоліків, які необхідно враховувати.

Необхідно врахувати наступне.

Мультизадачність та циклічність життєвого циклу додатків. Android одночасно запускає та виконує безліч процесів, тому додатки повинні бути оптимізовані для економного використання обмежених ресурсів мобільних пристроїв: оперативної пам'яті, заряду батареї, обчислювальних потужностей тощо. Це може бути складним завданням для розробників і вимагає ретельної оптимізації коду.

Розробка інтуїтивного та зручного користувацького інтерфейсу з урахуванням специфіки сенсорного введення. Необхідно дотримуватись принципів мобільного дизайну та матеріального дизайну, рекомендованого Google для забезпечення гармонійної взаємодії користувача з додатком. Створення зручного інтерфейсу може бути складним і вимагає ретельного підходу до дизайну.

Підтримка різноманітних пристроїв з різними розмірами екранів, роздільною здатністю, продуктивністю тощо. Розробники мають забезпечити адаптивний дизайн та оптимізувати продуктивність додатку для плавної

роботи на слабких та потужних пристроях. Це може бути складним завданням через велику кількість різноманітних Android-пристроїв на ринку.

Особливості архітектури операційної системи Android, засоби збереження та обміну даними, роботу з мультимедіа, сенсорами, картами, геолокацією тощо. Потрібно ретельно вивчити ці аспекти і правильно їх реалізувати у додатку.

Питання безпеки та захисту персональних даних користувачів є особливо актуальним для мобільних додатків. Розробники повинні реалізувати надійні механізми авторизації, шифрування інформації, запобігання витоку конфіденційних даних. Це може бути складним завданням і вимагає ретельного підходу до безпеки.

Одним з недоліків розробки для Android є фрагментація операційної системи – наявність великої кількості різних версій Android на пристроях користувачів. Це може ускладнювати забезпечення сумісності додатку з усіма версіями ОС.

Розробка для Android здійснюється переважно із використанням мови програмування Java або Kotlin та офіційного інструментарію Android Studio IDE. Використання Java має такі переваги:

– Java є зрілою та популярною мовою програмування, з великою спільнотою розробників і величезною кількістю ресурсів для навчання.

– Багато розробників вже знайомі з Java, що полегшує розробку додатків для Android.

– Java підтримує багато корисних парадигм, таких як об'єктно-орієнтоване програмування, що спрощує структурування коду та повторне використання компонентів.

– Існує величезна кількість готових бібліотек та фреймворків для Java, які можна використовувати у розробці Android-додатків.

Однак, Java також має певні недоліки в контексті розробки для Android:

– Це застаріла мова програмування з деякими обмеженнями та надмірною вербозністю коду.

– Компіляція Java-коду у байт-код, який потім виконується віртуальною машиною, може призводити до деякого зниження продуктивності порівняно з нативними додатками.

З іншого боку, Kotlin - це більш сучасна мова програмування, розроблена спеціально для Android. Вона має такі переваги:

– Kotlin є більш лаконічною та експресивною мовою, що дозволяє писати код коротше та читабельніше.

– Kotlin повністю сумісний з Java, що дозволяє використовувати існуючі Java-бібліотеки в Kotlin-проектах.

– Kotlin вбудовує багато сучасних можливостей, таких як функції вищого порядку, лямбда-вирази, null-безпека тощо.

– Код на Kotlin у багатьох випадках може бути більш безпечним і менш схильним до помилок, ніж еквівалентний код на Java.

Недоліки Kotlin полягають у тому, що це відносно нова мова, яка ще не настільки широко прийнята та поширена порівняно з Java. Також існуюча база знань, навчальні ресурси та кількість бібліотек для Kotlin поки що менші, ніж для Java.

Для створення системи психологічної діагностики було обрано Kotlin як мову програмування, а Android Studio - як середовище розробки (IDE). Ось детальне обґрунтування цих рішень:

Вибір Kotlin як мови програмування:

– Kotlin був розроблений компанією JetBrains спеціально для Android-розробки, тому він чудово інтегрований з Android Studio IDE та Android SDK. Це забезпечує зручність та продуктивність розробки, оскільки Kotlin був спочатку створений для цієї платформи.

– Kotlin є більш сучасною, лаконічною та безпечною мовою програмування порівняно з Java. Він включає багато нових можливостей, таких як функції вищого порядку, лямбда-вирази, null-безпека тощо, які

роблять код більш стислим, читабельним та безпечним. Це спрощує процес написання, зрозумілості та підтримки коду.

– Незважаючи на те, що Java має більшу базу знань та більше доступних ресурсів для навчання, Kotlin активно набирає популярність у світі Android-розробки. Існує все більше навчальних матеріалів, книг, курсів, відеоуроків для вивчення Kotlin, включно з ресурсами, орієнтованими спеціально на Android-розробку.

– Google офіційно підтримує Kotlin як одну з основних мов для Android-розробки, що свідчить про довгострокову перспективу та актуальність цієї мови для створення додатків для Android.

– Код на Kotlin є більш стислим та лаконічним порівняно з еквівалентним кодом на Java. Це полегшує процес написання, зрозумілості та підтримки коду.

Вибір Android Studio IDE:

– Android Studio є офіційним інтегрованим середовищем розробки (IDE) для Android, яке надає потужні інструменти та можливості для створення додатків на цій платформі. Воно тісно інтегроване з Android SDK, бібліотеками та фреймворками, необхідними для розробки.

– Android Studio забезпечує зручний інтерфейс для написання коду, налагодження, тестування та розгортання додатків на різних пристроях або емуляторах Android.

– Ця IDE надає набір спеціалізованих інструментів для роботи з користувацьким інтерфейсом, створення макетів, анімацій та візуалізації структури додатку.

– Android Studio підтримує розробку як на Java, так і на Kotlin, а також забезпечує відмінну інтеграцію та сумісність з Kotlin, оскільки обидва інструменти розробляються Google та JetBrains відповідно.

– Ця IDE постійно оновлюється та покращується, забезпечуючи підтримку нових версій Android, бібліотек та інструментів. Вона також має велику спільноту розробників та багато доступних навчальних ресурсів.

Отже, завдяки тісній інтеграції Kotlin з Android Studio, підтримці зі сторони Google, сучасним можливостям та безпеці Kotlin, а також потужному інструментарію Android Studio, ці технології забезпечать ефективну розробку високоякісної системи психологічної діагностики для Android.

2.6 Висновки за розділом 2.

Описано ключові компоненти психодіагностичних методик, такі як інструкція, шкала оцінювання, набір запитань та ключі для інтерпретації результатів. Також зазначено важливі психометричні властивості - валідність, надійність і стандартизованість.

Детально розглянуто чотири психодіагностичні методики та описано алгоритми їх реалізації: опитувальник тривожності Спілбергера-Ханіна, опитувальник депресії Бека, опитувальник агресивності Басса-Дарки та шкала прокрастинації Такмана.

Для кожної методики описано підготовку запитань, процес проходження тесту, обробку та інтерпретацію результатів, візуалізацію, збереження даних, обробку помилок та додаткові функції.

Окремо розглянуто специфіку та виклики розробки додатків для операційної системи Android, зокрема необхідність оптимізації, адаптивного дизайну, підтримку безпеки тощо.

Обґрунтовано вибір мови програмування Kotlin та середовища Android Studio як оптимальних технологій для розробки системи психологічної діагностики на Android.

В цілому розділ надає детальний опис алгоритмів реалізації різних психодіагностичних методик та практичні рекомендації щодо розробки мобільного додатку для психологічної діагностики на платформі Android.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ АЛГОРИТМІВ

3.1 Загальний опис розроблюваної системи

Система психологічної діагностики є мобільним додатком, розробленим у середовищі Android Studio з використанням мови програмування Kotlin. Ця система орієнтована на проведення різноманітних психологічних тестів та опитувальників з метою надання користувачам можливості самостійної діагностики власного психоемоційного стану, особистісних рис, схильностей та потенційних проблем.

Ключові модулі системи.

– Модуль реєстрації та авторизації користувачів:

– Створення облікового запису з введенням персональних даних (ім'я, вік, стать тощо).

– Авторизація зареєстрованих користувачів для доступу до функціоналу додатку.

– Збереження даних користувачів у базі даних.

Модуль меню вибору психологічних тестів:

– Відображення переліку доступних психологічних опитувальників та тестів (тривожність, депресія, агресивність, прокрастинація тощо).

– Надання стислої інформації про призначення кожного тесту.

– Можливість вибору бажаного тесту для проходження.

Модуль проходження психологічних тестів:

– Інтерактивний інтерфейс для відповідей на запитання тесту (вибір варіантів, шкали оцінювання тощо).

– Перевірка повноти та коректності відповідей перед завершенням тесту.

Модуль розрахунку та інтерпретації результатів:

– Автоматичний розрахунок результатів на основі відповідей користувача та ключів/норм тесту.

– Застосування відповідних алгоритмів, формул та правил обчислення залежно від типу тесту.

– Інтерпретація отриманих чисельних показників у змістовні психологічні висновки та характеристики.

– Формування зрозумілих текстових звітів з результатами тестування.

Модуль візуалізації та демонстрації результатів:

– Відображення підсумкових результатів тестування у зручному для сприйняття форматі.

– Надання детальних пояснень та інтерпретацій отриманих показників.

Система побудована з дотриманням принципів об'єктно-орієнтованого програмування, модульної архітектури та використанням сучасних бібліотек і фреймворків для розробки під Android (Room, ViewModels, LiveData тощо). Також приділено значну увагу питанням забезпечення безпеки та захисту персональних даних користувачів.

Інтерфейс додатку розроблений з урахуванням принципів матеріального дизайну (Material Design) від Google, забезпечуючи зручність, інтуїтивність та сучасний вигляд. Навігація між модулями та екранами реалізована з використанням стандартних патернів Android (Navigation Component).

Основна мета розробленої системи психологічної діагностики - надати користувачам зручний та доступний інструмент для самооцінки свого психоемоційного стану, виявлення потенційних проблем та відстеження динаміки змін за допомогою комплексу валідних психологічних тестів. Система орієнтована на широке коло користувачів, які зацікавлені у самопізнанні та психологічному благополуччі.

3.2 Реалізація модулю реєстрації та авторизації

Алгоритм реєстрації (MainActivity):

Крок 1: Введення даних

1. Користувач вводить логін у поле “userLogin”.

2. Користувач вводить пароль у поле “userPass”.

Крок 2: Перевірка полів

1. При натисканні на кнопку "Зареєструватися" (“buttonReg”):

- Система перевіряє, чи обидва поля (“userLogin” та “userPass”) заповнені.

- Якщо будь-яке поле порожнє, виводиться повідомлення "Не всі поля заповнені!!!".

Крок 3: Створення об'єкта користувача

1. Якщо всі поля заповнені, створюється об'єкт класу “User” з введеним логіном та паролем.

```
val user = User(userLogin.text.toString(), userPass.text.toString())
```

Крок 4: Додавання користувача до бази даних

1. Об'єкт “User” передається методу “addUser” класу “DbHelper”, який додає нового користувача до таблиці “users” у базі даних.

```
val dbHelper = DbHelper(context)
dbHelper.addUser(user)
```

Крок 5: Повідомлення про успіх

1. Після успішного додавання користувача до бази даних виводиться повідомлення "Користувач [логін] зареєстрован".

```
Toast.makeText(context, "Користувач ${user.login} зареєстрован", Toast.LENGTH_SHORT).show()
```

Крок 6: Очищення полів

1. Поля “userLogi” та “userPass” очищуються.

```
userLogin.text.clear()
userPass.text.clear()
```

Крок 7: Перехід до авторизації

1. Користувач автоматично перенаправляється на екран авторизації (“AuthActivity”).

```
val intent = Intent(this, AuthActivity::class.java)
startActivity(intent)
finish()
```

Алгоритм авторизації (AuthActivity)

Крок 1: Введення даних

1. Користувач вводить логін у поле “userLoginAuth”.

2. Користувач вводить пароль у поле “userPassAuth”.

Крок 2: Перевірка полів

1. При натисканні на кнопку "Увійти" (“button”):

- Система перевіряє, чи обидва поля (“userLoginAuth” та “userPassAuth”) заповнені.

- Якщо будь-яке поле порожнє, виводиться повідомлення "Не всі поля заповнені!!!".

Крок 3: Перевірка користувача в базі даних

1. Якщо всі поля заповнені, метод “getUser” класу “DbHelper” перевіряє, чи існує користувач з введеним логіном та паролем у таблиці “users”.

```
val user = dbHelper.getUser(userLoginAuth.text.toString(), userPassAuth.text.toString())
```

Крок 4: Повідомлення про результат

1. Якщо користувач знайдений:

- Виводиться повідомлення "Користувач [логін] авторизован".

- Користувач перенаправляється на головне меню (“MainMenu”).

```
Toast.makeText(context, "Користувач ${user.login} авторизован", Toast.LENGTH_SHORT).show()
val intent = Intent(this, MainMenu::class.java)
```

startActivity(intent)

finish()

2. Якщо користувач не знайдений:

- Виводиться повідомлення "Користувач [логін] НЕ авторизован".

Toast.makeText(context, "Користувач ${userLoginAuth.text} НЕ авторизован", Toast.LENGTH_SHORT).show()

Цей алгоритм забезпечує базові кроки реєстрації та авторизації користувачів у мобільному додатку, зберігаючи дані користувачів у локальній базі даних та перевіряючи їх при вході в систему.

Готові модулі реєстрації та авторизації можуть бути у майбутньому бути легко заміненими на більш складні та надійні системи.

3.3 Алгоритм модулю проходження тестів (на основі «опитувальника Ч.Д. Спілбергера для оцінки тривожності»

Ініціалізація:

1. При запуску активності Anxiety ініціалізуються змінні для підрахунку балів:

scoreCT0 - бали за ситуативну тривожність, варіант 0

scoreCT1 - бали за ситуативну тривожність, варіант 1

scoreOT0 - бали за особисту тривожність, варіант 0

scoreOT1 - бали за особисту тривожність, варіант 1

2. Ініціалізується список для історії відповідей (answerHistory).

3. Завантажується список питань (questions), який містить:

Тексти питань

Типи питань ("CT" для ситуативної тривожності, "OT" для особистої тривожності)

Індекси правильних відповідей

4. Встановлюється початковий індекс питання (currentQuestionIndex) на 0.

Відображення питання

1. Викликається функція updateQuestion(), яка відображає поточне питання та варіанти відповідей на екрані.

2. Текст питання береться з об'єкта Question за поточним індексом.

3. Варіанти відповідей залежать від типу питання ("СТ" - ситуативна тривожність, "ОТ" - особиста тривожність).

Вибір відповіді

1. Користувач вибирає один з варіантів відповідей у RadioGroup.

Обробка відповіді (при натисканні на кнопку "Далі")

1. Перевіряється, чи вибрано відповідь. Якщо ні, виводиться повідомлення "Оберіть відповідь".

2. Якщо відповідь вибрано, визначається її індекс та розраховуються бали:

   Для питань типу "СТ" бали додаються до scoreCT0 або scoreCT1 залежно від правильної відповіді.

   Для питань типу "ОТ" бали додаються до scoreOT0 або scoreOT1.

3. Інформація про відповідь (індекс питання, індекс відповіді, зміна балів) зберігається в answerHistory.

4. Якщо це не останнє питання, викликається updateQuestion() для відображення наступного питання.

5. Якщо це останнє питання, викликається showResults() для відображення результатів.

6. Індекс питання збільшується на 1, якщо це не останнє питання.

7. Вибір у RadioGroup скидається.

Обробка натискання на кнопку "Назад"

1. Якщо це не перше питання і є історія відповідей:

   Остання відповідь видаляється з answerHistory, а бали відповідно зменшуються.

   Індекс питання зменшується на 1 (але не менше 0).

Викликається updateQuestion() для відображення попереднього питання.

Відображення результатів (функція showResults())

1. Розраховуються показники ситуативної та особистої тривожності на основі набраних балів.

2. Створюється Intent для запуску активності res_anxiety та передаються розраховані показники.

3. Запускається активність res_anxiety для відображення результатів.

Зауваження:

Алгоритм підрахунку балів може бути змінений залежно від конкретної тестової методики.

3.4 Висновки за розділом

Реалізація алгоритмів - це критично важливий аспект у розробці будь-якої програмної системи, зокрема системи психологічної діагностики, яка розглядається у наданих матеріалах. Ретельно продумані та чітко описані алгоритми забезпечують основу для ефективної та коректної роботи всієї системи, а також полегшують процес її програмування та подальшого супроводу.

У контексті даної системи психологічної діагностики представлені детальні алгоритми для ключових модулів, таких як реєстрація та авторизація користувачів, а також проходження психологічних тестів на прикладі опитувальника Спілбергера для оцінки тривожності. Ці алгоритми враховують різноманітні сценарії та умови, забезпечуючи коректну обробку введених даних, перевірку полів, збереження та маніпуляції з даними користувачів, історією відповідей та розрахунком балів відповідно до правил тесту.

Крім того, у наданих матеріалах підкреслюється модульний підхід до архітектури системи, який дозволяє легко замінювати або оновлювати окремі компоненти, такі як модулі реєстрації/авторизації, без необхідності переписування всього коду. Це забезпечує гнучкість та масштабованість системи, а також полегшує її подальший розвиток та вдосконалення.

Алгоритми описані з використанням стандартних концепцій програмування, таких як об'єкти, списки та структури даних, що робить їх зрозумілими та доступними для реалізації у програмному коді. Це дозволяє розробникам легко перетворити описані алгоритми на працюючі програмні модулі, забезпечуючи коректну реалізацію бізнес-логіки системи.

Крім того, у матеріалах згадується про увагу, яка приділяється питанням безпеки та захисту персональних даних користувачів, що є критично важливим аспектом для будь-якої системи, яка обробляє конфіденційну інформацію. Хоча конкретні деталі не наведені, наголос на цьому питанні свідчить про усвідомлення його важливості та потенційне врахування відповідних заходів безпеки під час реалізації алгоритмів.

Загалом, представлені матеріали демонструють ретельний підхід до планування та опису алгоритмів системи психологічної діагностики. Це забезпечує міцну основу для подальшої реалізації цих алгоритмів у програмному коді, гарантуючи коректну роботу системи, її гнучкість, масштабованість та захист даних користувачів. Детальне документування алгоритмів також полегшує процеси розробки, тестування та супроводу системи в майбутньому.

РОЗДІЛ 4 ОПИС РОБОТИ

4.1 Архітектура додатку

Патерн MVVM (Model-View-ViewModel)

Додаток системи психологічної діагностики побудований з використанням архітектурного патерну MVVM, який є рекомендованим Google для розробки додатків під Android. Цей патерн передбачає розділення відповідальностей між трьома основними компонентами:

– Model. Відповідає за представлення даних додатку та бізнес-логіку. У цьому контексті, моделями можуть бути класи, що представляють користувачів, тести, результати тощо.

– View. Відповідає за відображення користувацького інтерфейсу та реагування на взаємодію користувача. Представлене Activity, Fragment та елементами інтерфейсу (кнопки, списки, текстові поля тощо).

– ViewModel. Виступає проміжною ланкою між View та Model. ViewModel керує станом даних, необхідним для відображення в UI, і забезпечує бізнес-логікою представлення даних. ViewModels також допомагають зберігати дані при зміні конфігурації пристрою (наприклад, при повороті екрану).

Компоненти додатку

Відповідно до модульної архітектури, додаток складається з таких основних компонентів.

Модуль реєстрації та авторизації користувачів

– View: Activity та Fragment для введення даних користувача.

– ViewModel. Керує процесом реєстрації/авторизації та взаємодіє з моделями користувачів.

– Model. Клас User для представлення даних користувача.

Модуль меню вибору психологічних тестів:

– View. Activity або Fragment для відображення списку тестів.

– ViewModel. Керує даними про доступні тести та взаємодіє з відповідними моделями.

– Model. Класи, що представляють дані про тести.

Модуль проходження психологічних тестів:

– View. Activity або Fragment для відображення запитань тесту та вибору відповідей.

– ViewModel. Керує станом тесту, обробляє відповіді та взаємодіє з моделями тестів.

– Model. Класи, що представляють дані про тести, запитання та відповіді.

Модуль розрахунку та інтерпретації результатів

– ViewModel. Виконує обчислення результатів на основі відповідей користувача та правил тестів.

– Model. Класи, що представляють результати тестів та правила їх інтерпретації.

Модуль візуалізації та демонстрації результатів

– View. Activity або Fragment для відображення результатів тестування.

– ViewModel. Керує даними про результати та їх візуалізацією.

– Model. Класи, що представляють результати тестів та візуалізації даних.

Структура проекту

Модулі

Авторизація та реєстрація:

– MainActivity.kt: Екран реєстрації нових користувачів.

– AuthActivaty.kt: Екран авторизації існуючих користувачів.

– DbHelper.kt: Клас для роботи з базою даних, де зберігаються дані користувачів.

– User.kt: Клас, що представляє об'єкт користувача з логіном та паролем.

Проходження тесту:

– Anxiety.kt: Екран для проходження тесту на тривожність.

Відображення результатів:

– res_anxiety.kt: Екран для відображення результатів тесту.

Файли макетів (XML):

– activity_main.xml: Макет екрану реєстрації.

– activity_auth_activaty.xml: Макет екрану авторизації.

– activity_anxiety.xml: Макет екрану проходження тесту.

– activity_res_anxiety.xml: Макет екрану відображення результатів.

База даних:

– Таблиця users для зберігання даних користувачів (логін, пароль).

Логіка:

Реєстрація: Збір даних користувача, створення об'єкта User, додавання до бази даних.

Авторизація: Перевірка введених даних з даними в базі даних.

Проходження тесту: Відображення питань, збір відповідей, підрахунок балів.

Відображення результатів: Розрахунок показників тривожності, відображення результатів.

Технології:

Мова програмування: Kotlin

Фреймворк: Android SDK

База даних: SQLite

Ця структура проекту забезпечує чітке розділення відповідальностей між різними компонентами додатку, полегшує підтримку та розширення функціоналу в майбутньому. Крім того, вона відповідає рекомендованим практикам розробки додатків для Android з використанням патерну MVVM та інших сучасних підходів.

4.2 Опис основних модулів

Модуль авторизації та реєстрації

Цей модуль відповідає за управління процесами реєстрації нових користувачів та авторизації існуючих користувачів у системі. Він містить такі ключові компоненти:

MainActivity.kt:

– Activity, що відображає екран реєстрації нових користувачів.

– Містить поля для введення логіна та паролю, а також кнопку "Зареєструватися".

– Реалізує логіку перевірки заповнених полів та створення нового об'єкта користувача (User).

– Викликає метод DbHelper для додавання нового користувача до бази даних.

– Після успішної реєстрації переходить до екрану авторизації (AuthActivity).

AuthActivity.kt:

– Activity, що відображає екран авторизації існуючих користувачів.

– Містить поля для введення логіна та паролю, а також кнопку "Увійти".

– Реалізує логіку перевірки заповнених полів та звертається до DbHelper для отримання даних користувача з бази даних.

– У разі успішної авторизації переходить до головного меню додатка (MainMenu).

DbHelper.kt:

– Клас, що інкапсулює логіку роботи з базою даних SQLite.

– Містить метод addUser для додавання нового користувача до таблиці "users".

– Містить метод getUser для отримання даних користувача з таблиці "users" за логіном та паролем.

– Забезпечує створення, відкриття та закриття з'єднання з базою даних.

User.kt:

– Клас-модель, що представляє об'єкт користувача з його даними (логін, пароль).

– Використовується для передачі даних між різними компонентами модуля.

Модуль вибору тестів

Цей модуль відповідає за відображення списку доступних психологічних тестів та надання інформації про них. Він може містити такі компоненти:

TestSelectionActivity.kt або TestSelectionFragment.kt:

– Activity або Fragment, що відображає екран вибору тестів.

– Містить список (ListView або RecyclerView) із назвами та стислою інформацією про наявні тести.

– Реалізує логіку взаємодії з користувачем, наприклад, обробку натискання на певний тест у списку.

TestSelectionViewModel.kt:

– ViewModel, що керує даними про доступні тести.

– Взаємодіє з моделями даних (наприклад, TestRepository) для отримання списку тестів.

– Готує дані для відображення в UI (наприклад, підготовка списку назв тестів).

TestRepository.kt або TestDao.kt:

– Репозиторій або Data Access Object (DAO) для доступу до даних про тести.

– Може взаємодіяти з базою даних або іншими джерелами даних для отримання інформації про тести.

– Повертає дані у вигляді об'єктів класів-моделей (наприклад, Test).

Test.kt:

– Клас-модель, що представляє об'єкт психологічного тесту з його властивостями (назва, опис, категорія тощо).

Модуль проходження тестів

Цей модуль відповідає за відображення запитань обраного психологічного тесту, збір відповідей користувача та підрахунок балів відповідно до правил тесту. Він може містити такі компоненти:

TestActivity.kt або TestFragment.kt:

– Activity або Fragment, що відображає екран проходження тесту.

– Містить елементи інтерфейсу для відображення питань та варіантів відповідей (наприклад, TextView, RadioGroup).

– Реалізує логіку обробки вибору відповіді користувачем та переходу до наступного/попереднього питання.

TestViewModel.kt:

– ViewModel, що керує станом тесту та обробляє відповіді користувача.

– Взаємодіє з моделями даних (наприклад, QuestionRepository, AnswerRepository) для отримання питань та відповідей.

– Реалізує бізнес-логіку підрахунку балів відповідно до правил тесту.

– Готує дані для відображення в UI (наприклад, текст поточного питання, варіанти відповідей).

QuestionRepository.kt або QuestionDao.kt:

– Репозиторій або DAO для доступу до даних про питання тесту.

– Може взаємодіяти з базою даних або іншими джерелами даних для отримання питань.

– Повертає дані у вигляді об'єктів класів-моделей (наприклад, Question).

Question.kt:

– Клас-модель, що представляє об'єкт питання тесту з його властивостями (текст питання, тип питання, варіанти відповідей тощо).

AnswerRepository.kt або AnswerDao.kt:

– Репозиторій або DAO для доступу до даних про відповіді на питання тесту.

– Може взаємодіяти з базою даних або іншими джерелами даних для отримання відповідей.

– Повертає дані у вигляді об'єктів класів-моделей (наприклад, Answer).

Answer.kt:

– Клас-модель, що представляє об'єкт відповіді на питання тесту з її властивостями (текст відповіді, бали тощо).

Модуль демонстрації результатів

Цей модуль відповідає за відображення результатів проходження тесту користувачем та їх інтерпретацію. Він може містити такі компоненти:

ResultActivity.kt або ResultFragment.kt:

– Activity або Fragment, що відображає екран із результатами тестування.

– Містить елементи інтерфейсу для візуалізації результатів (наприклад, TextView, графіки, діаграми).

– Може відображати детальну інтерпретацію результатів у текстовому форматі.

ResultViewModel.kt:

– ViewModel, що керує даними про результати тестування.

– Взаємодіє з моделями даних (наприклад, ResultRepository) для отримання результатів користувача.

– Реалізує бізнес-логіку інтерпретації результатів відповідно до правил тесту.

– Готує дані для відображення в UI (наприклад, розраховані показники, текстові інтерпретації).

ResultRepository.kt або ResultDao.kt:

– Репозиторій або DAO для доступу до даних про результати тестування.

– Може взаємодіяти з базою даних або іншими джерелами даних для збереження та отримання результатів.

– Повертає дані у вигляді об'єктів класів-моделей (наприклад, TestResult).

4.3 Користувацький інтерфейс

Дизайн та юзабіліті

Дизайн користувацького інтерфейсу системи психологічної діагностики розроблений з урахуванням принципів зручності використання (юзабіліті) та сучасних підходів до проектування UI для мобільних додатків. Для забезпечення інтуїтивної взаємодії користувачів з додатком, були використані стандартні елементи інтерфейсу операційної системи Android, такі як `TextView`, `EditText`, `Button`, `RadioGroup` та `RadioButton`. Ці компоненти є добре знайомими більшості користувачів смартфонів та планшетів на базі Android, що забезпечує простоту освоєння та роботи з додатком.

Окрім того, дизайн базується на рекомендаціях Google щодо матеріального дизайну (Material Design). Це означає, що в додатку використовується сучасна кольорова схема, стилі та анімації, характерні для матеріального дизайну. Такий підхід робить інтерфейс привабливим та естетично приємним, а також сприяє створенню цілісного та послідовного користувацького досвіду.

Для підвищення зручності використання та візуальної привабливості застосовано наступні підходи:

– Використання анімацій та плавних переходів між екранами та компонентами інтерфейсу, що створює відчуття безперервності та динамічності.

– Адаптивний дизайн, що забезпечує коректне відображення та зручність роботи на пристроях з різними розмірами екранів та в різних орієнтаціях (портретній та альбомній).

– Застосування принципів ергономіки та доступності, таких як достатній розмір елементів інтерфейсу для зручного натискання, контрастні кольори для кращої видимості та інше.

Загалом, дизайн користувацького інтерфейсу системи психологічної діагностики орієнтований на створення приємного та інтуїтивного досвіду для користувачів, забезпечуючи зручність проходження тестів та отримання результатів.

Навігація та потоки екранів

Навігація між різними розділами та екранами додатку реалізована з використанням стандартного механізму Android - `Intent`. Це забезпечує просту та зрозумілу логіку переходів, що є добре знайомою для більшості користувачів Android-пристроїв.

Основні потоки екранів у додатку.

Екран реєстрації та авторизації користувачів:

– Користувачі можуть створити новий обліковий запис, ввівши свої персональні дані (логін, пароль тощо).

– Зареєстровані користувачі можуть увійти в систему, ввівши свої облікові дані.

Головне меню з переліком доступних психологічних тестів:

– Після успішної авторизації користувачі переходять до головного меню.

– У меню відображається список доступних психологічних тестів та опитувальників (наприклад, на тривожність, депресію, агресивність тощо).

– Користувачі можуть вибрати бажаний тест для проходження.

Екран проходження обраного психологічного тесту:

– Користувачам послідовно відображаються запитання тесту.

– Вони можуть вибирати варіанти відповідей, використовуючи різні типи елементів інтерфейсу (RadioButton, шкали тощо).

– Передбачена можливість переходу до попереднього/наступного питання.

Екран результатів тестування з візуалізацією та інтерпретацією отриманих даних:

– Після завершення тесту користувачам демонструються їхні результати.

– Результати можуть бути представлені у вигляді чисельних показників, графіків, діаграм тощо.

– Надається детальна інтерпретація отриманих результатів у текстовому форматі.

Така структура навігації забезпечує логічний та інтуїтивний потік дій користувача у процесі реєстрації, авторизації, вибору тесту, його проходження та отримання результатів. Це полегшує використання додатку та підвищує загальну зручність роботи з ним.

Типові елементи UI

Для реалізації користувацького інтерфейсу проекту використовуються стандартні елементи UI Android, а також додаткові компоненти для забезпечення більш складної та функціональної взаємодії з користувачами. Серед основних елементів можна виділити:

– TextView - для відображення тексту (назв тестів, питань, інструкцій тощо).

– EditText - для введення даних користувачем (логін, пароль під час реєстрації/авторизації).

– Button - для виконання дій (реєстрація, авторизація, перехід між питаннями тесту тощо).

– RadioGroup та RadioButton - для відображення варіантів відповідей на питання тестів та вибору користувачем потрібного варіанту.

Додатково, для підвищення функціональності та візуальної привабливості, використовуються такі компоненти:

– RecyclerView - для відображення списків елементів (наприклад, списку доступних тестів) з можливістю динамічного додавання та видалення елементів.

– CardView - для виділення окремих елементів інтерфейсу (наприклад, окремих тестів у списку) та покращення їх візуальної презентації.

– ConstraintLayout - для створення складних та адаптивних макетів, що забезпечують зручне розташування елементів на екрані та їх коректне відображення на різних пристроях.

– Snackbar та Toast - для відображення короткочасних повідомлень та сповіщень, що інформують користувачів про важливі події або помилки (наприклад, успішна реєстрація, некоректно введені дані тощо).

– Компоненти матеріального дизайну, такі як BottomSheetDialog та FloatingActionButton, що додають інтерактивності та покращують загальний вигляд додатку, роблячи його більш сучасним та привабливим.

Використання цих елементів UI дозволяє створити зрозумілий, функціональний та естетично привабливий інтерфейс.

4.4 Висновки за розділом

В даному розділі дипломної роботи детально описано архітектуру та структуру розробленого додатку системи психологічної діагностики на платформі Android.

Архітектура додатку:

Проект побудований з використанням сучасного архітектурного патерну MVVM (Model-View-ViewModel), рекомендованого компанією Google для розробки додатків під операційну систему Android. Даний патерн забезпечує чітке розділення відповідальностей між компонентами, що відповідають за представлення даних та бізнес-логіку (Model), відображення користувацького інтерфейсу (View) та проміжний зв'язок між ними (ViewModel). Застосування такого підходу спрощує розробку, тестування та подальше супроводження програмного забезпечення.

Модульна структура додатку включає такі основні компоненти:

– Модуль реєстрації та авторизації користувачів
– Модуль вибору психологічних тестів
– Модуль проходження тестів
– Модуль розрахунку та інтерпретації результатів
– Модуль візуалізації результатів

Для кожного модуля детально описано його компоненти (Activities, Fragments, ViewModels, Repositories/DAOs, класи-моделі даних), їх відповідальність та логіку роботи. Це забезпечує можливість розуміння та подальшої підтримки проекту.

Користувацький інтерфейс:

Дизайн користувацького інтерфейсу розроблений із дотриманням принципів юзабіліті та сучасних підходів до створення UI для мобільних додатків, зокрема рекомендацій матеріального дизайну від Google. Використано стандартні елементи інтерфейсу Android, такі як TextView, EditText, Button, RadioGroup тощо для забезпечення інтуїтивної взаємодії користувачів. Застосовано анімації, адаптивний дизайн та принципи

ергономіки для покращення загального досвіду роботи з додатком. Детально описано потоки екранів та навігацію між ними, а також типові елементи UI, що використовуються в різних частинах додатку.

Загалом, розділ дипломної роботи надає вичерпний опис архітектури, модулів, користувацького інтерфейсу та основних компонентів додатку системи психологічної діагностики, що розроблявся в рамках проекту. Це дозволяє отримати повне уявлення про структуру та принципи функціонування програмного забезпечення.

РОЗДІЛ 5 ТЕСТУВАННЯ

5.1 Підходи до тестування

При розробці Android додатку у Android Studio на мові Kotlin, важливо забезпечити високу якість коду та правильну роботу всіх функцій додатку. Для досягнення цієї мети використовуються різні підходи до тестування, які дозволяють виявляти та виправляти помилки на різних етапах розробки.

Модульне тестування

Модульне тестування (Unit testing) - це процес тестування окремих модулів або компонентів додатку, таких як класи, методи чи функції, у ізольованому середовищі. Ці тести перевіряють правильність роботи певної одиниці коду та забезпечують швидке виявлення помилок на ранніх етапах розробки.

В Android Studio модульні тести зазвичай пишуться з використанням фреймворку JUnit та бібліотеки Mockito для створення моків (імітаційних об'єктів). Приклади модульних тестів:

– Тестування бізнес-логіки у ViewModels
– Тестування методів для обробки даних у Repository класах
– Тестування утиліт та допоміжних класів

Модульні тести допомагають забезпечити високу якість окремих компонентів додатку та полегшують рефакторинг коду.

Інтеграційне тестування

Інтеграційне тестування (Integration testing) перевіряє взаємодію між різними компонентами додатку та їх правильну інтеграцію. Ці тести імітують реальні сценарії використання, коли різні частини додатку взаємодіють між собою.

В Android Studio інтеграційні тести можна писати з використанням фреймворків, таких як Espresso або UIAutomator. Вони дозволяють імітувати

дії користувача та перевіряти правильність відображення даних в інтерфейсі користувача.

Приклади інтеграційних тестів:

– Тестування потоку авторизації користувача (введення даних, взаємодія з базою даних, перехід на головний екран)

– Тестування проходження тесту (відображення питань, збереження відповідей, перехід до результатів)

– Тестування відображення результатів тесту (коректність розрахунків, візуалізація даних)

Інтеграційні тести допомагають виявити помилки, пов'язані з взаємодією між різними компонентами додатку, та забезпечують стабільність системи в цілому.

Системне тестування

Системне тестування (System testing) перевіряє функціональність додатку в цілому, враховуючи всі його компоненти та їх взаємодію. Воно імітує реальні сценарії використання додатку користувачами та перевіряє, чи відповідає поведінка додатку очікуваним вимогам.

Під час системного тестування Android додатку можуть перевірятися такі аспекти:

– Функціональність: перевірка правильної роботи всіх функцій додатку

– Продуктивність: перевірка швидкодії та відгуку додатку на різних пристроях

– Стабільність: перевірка стійкості додатку до некоректних даних та помилок

– Сумісність: перевірка коректної роботи додатку на різних версіях Android та пристроях

Системне тестування може виконуватися вручну тестерами або частково автоматизовано за допомогою інструментів, таких як Appium або Robotium.

Юзабіліті тестування

Юзабіліті тестування (Usability testing) - це процес перевірки зручності використання додатку кінцевими користувачами. Воно дозволяє виявити проблеми в дизайні користувацького інтерфейсу, навігації та загальній зрозумілості додатку.

Юзабіліті тестування зазвичай проводиться з залученням реальних користувачів, які виконують певні завдання в додатку, тоді як спеціалісти спостерігають за їхніми діями та реакціями. Під час тестування можуть оцінюватися такі аспекти:

– Легкість навігації та розуміння інтерфейсу
– Зручність використання основних функцій додатку
– Ефективність виконання типових завдань
– Задоволеність користувачів від роботи з додатком

На основі результатів юзабіліті тестування можуть бути внесені зміни до дизайну та структури користувацького інтерфейсу для покращення досвіду користувачів.

Комбінація різних підходів до тестування (модульного, інтеграційного, системного та юзабіліті) дозволяє забезпечити високу якість Android додатку, його стабільність, коректну роботу функцій та зручність використання для кінцевих користувачів.

5.2 Опис тестових сценаріїв

Тестові сценарії для авторизації.

Авторизація є критичним компонентом будь-якого додатку, що вимагає забезпечення безпеки та зручності для користувачів. Описані нижче тестові сценарії охоплюють як позитивні, так і негативні випадки, що дозволяють перевірити коректність функціоналу авторизації.

Позитивні сценарії:

1. Успішна авторизація:

– Кроки: Ввести коректний логін та пароль існуючого користувача.

– Очікуваний результат: Успішна авторизація користувача та перехід до головного меню додатку (рис. 5.1).

Рисунок 5.1 – успішна авторизація

Негативні сценарії:

1. Некоректний логін:

– Кроки: Ввести неіснуючий логін та будь-який пароль.

– Очікуваний результат: Поява повідомлення про помилку "Користувач [логін] НЕ авторизован", авторизація не відбувається (рис. 5.2).

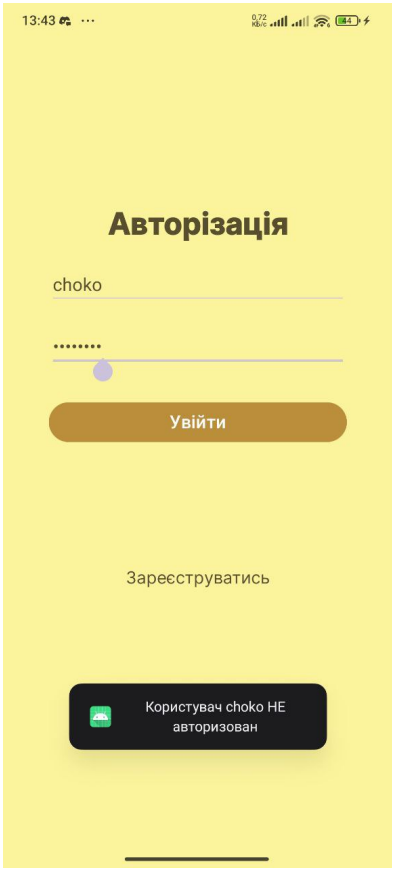

Рисунок 5.2. – Введено не існуючий логін при авторизації

2. Некоректний пароль:

– Кроки: Ввести існуючий логін та некоректний пароль.

– Очікуваний результат: Поява повідомлення про помилку "Користувач [логін] НЕ авторизован", авторизація не відбувається (рис. 5. 3).

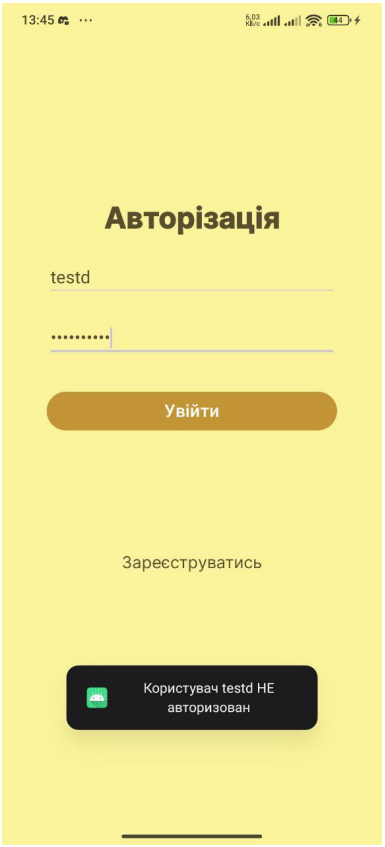

Рисунок 5.3 – Невірний пароль при авторизації

3. Пусті поля:

– Кроки: Не вводити логін та/або пароль.

– Очікуваний результат: Поява повідомлення про помилку "Не всі поля заповнені!!!", авторизація не відбувається (рис. 5.4).

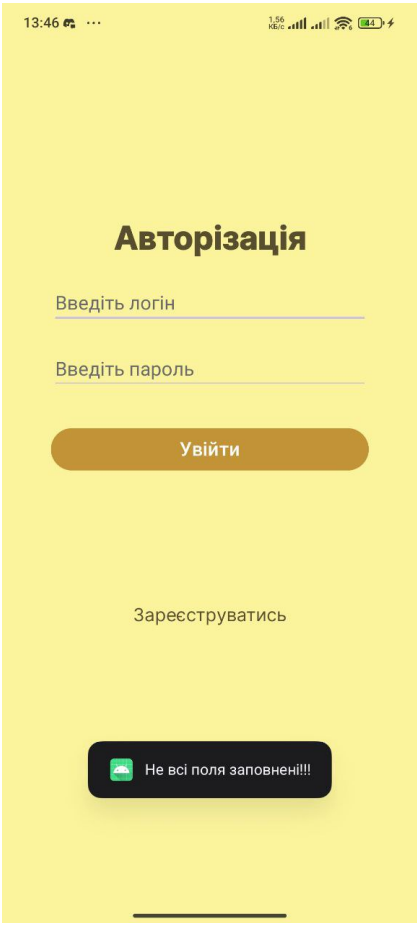

Рисунок 5. 4 – пусті поля при авторизації

Тестові сценарії для реєстрації

Реєстрація нових користувачів включає в себе перевірку введених даних та забезпечення унікальності облікових записів. Нижче наведені сценарії охоплюють різні варіанти взаємодії користувача з формою реєстрації.

Позитивні сценарії:

1. Успішна реєстрація:

– Кроки: Ввести унікальний логін та пароль.

– Очікуваний результат: Успішна реєстрація користувача, поява повідомлення "Користувач [логін] зареєстрован", перехід до екрану авторизації (рис. 5.5).

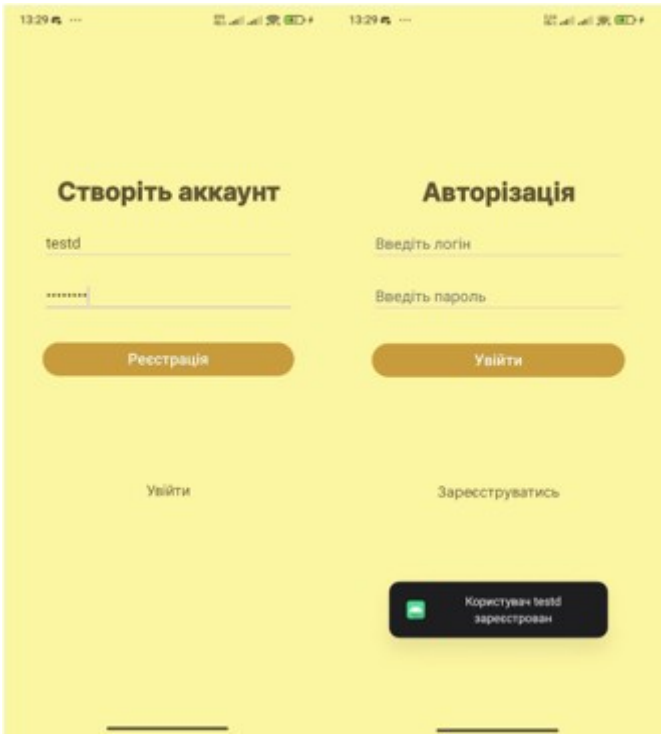

Рисунок 5. 5 – Успішна реєстрація

Негативні сценарії:

1. Існуючий логін:

– Кроки: Ввести логін, який вже існує в базі даних, та будь-який пароль.

– Очікуваний результат: Поява повідомлення про помилку "Користувач [логін] вже існує", реєстрація не відбувається (рис. 5.6).

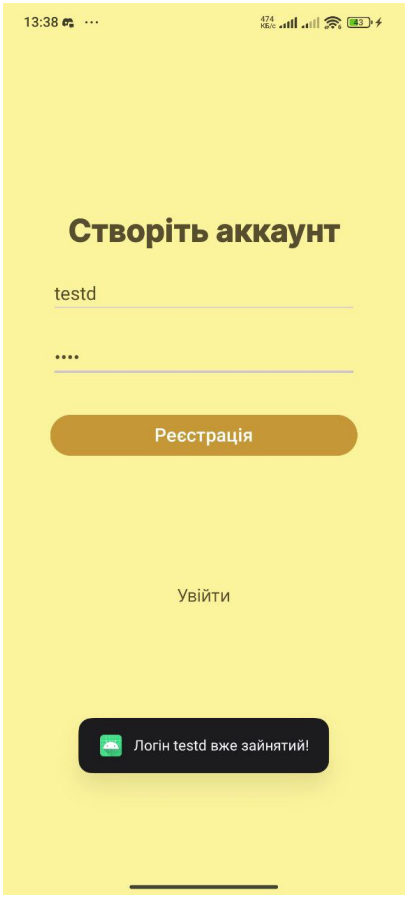

Рисунок 5.6 – введено існуючий логін при реєстрації

2. Пусті поля:

– Кроки: Не вводити логін та/або пароль.

– Очікуваний результат: Поява повідомлення про помилку "Не всі поля заповнені!!!", реєстрація не відбувається (рис. 5.7).

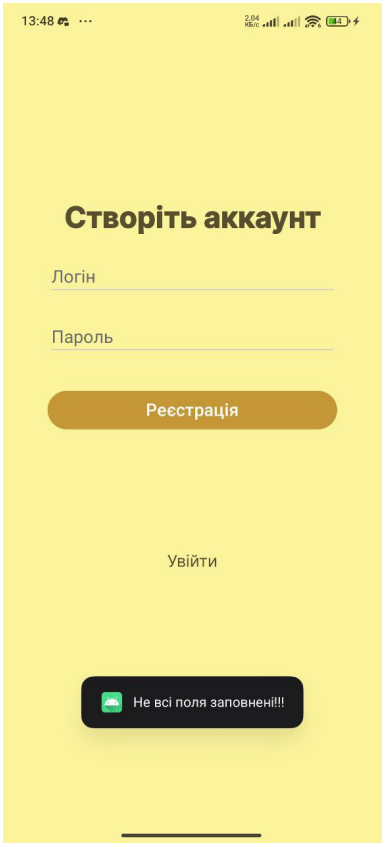

Рисунок 5.7 – Пусті поля при реєстрації

3. Некоректний формат логіну/паролю:

– Кроки: Ввести логін або пароль, які не відповідають вимогам (наприклад, занадто короткий пароль).

– Очікуваний результат: Поява повідомлення про помилку з вказівкою вимог до логіну/паролю, реєстрація не відбувається.

Тестові сценарії для проходження тестів

Проходження психологічних тестів є основною функцією додатку, що вимагає ретельної перевірки для забезпечення коректної роботи та збереження даних користувачів. Описані нижче сценарії охоплюють позитивні та негативні випадки взаємодії з тестами.

Позитивні сценарії:

1. Проходження тесту до кінця:

– Кроки: Вибрати тест зі списку. Відповісти на всі питання.

– Очікуваний результат: Перехід до екрану з результатами тесту (рис. 5.8).

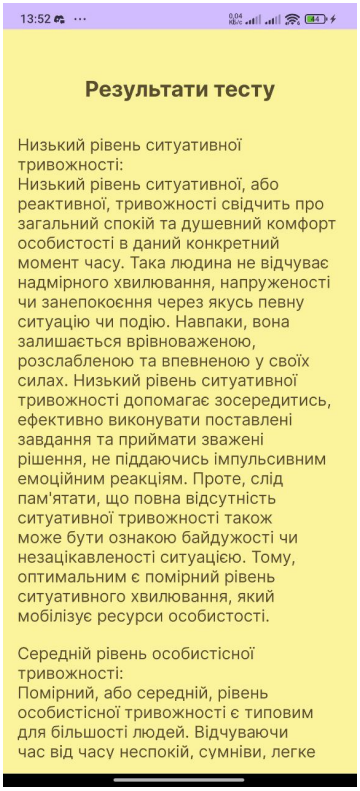

Рисунок 5.8 – успішне завершення тесту

2. Використання кнопок "Далі" та "Назад":

– Кроки: Вибрати тест зі списку (якщо є вибір). Відповісти на кілька питань. Використати кнопку "Назад" для повернення до попереднього питання. Змінити відповідь на попереднє питання. Продовжити проходження тесту.

– Очікуваний результат: Коректне відображення питань та збереження відповідей.

Негативні сценарії:

1. Пропуск питань:

– Кроки: Вибрати тест зі списку (якщо є вибір). Не відповідати на одне або кілька питань. Натиснути кнопку "Далі".

– Очікуваний результат: Поява повідомлення про необхідність відповісти на всі питання перед продовженням (рис. 5.9).

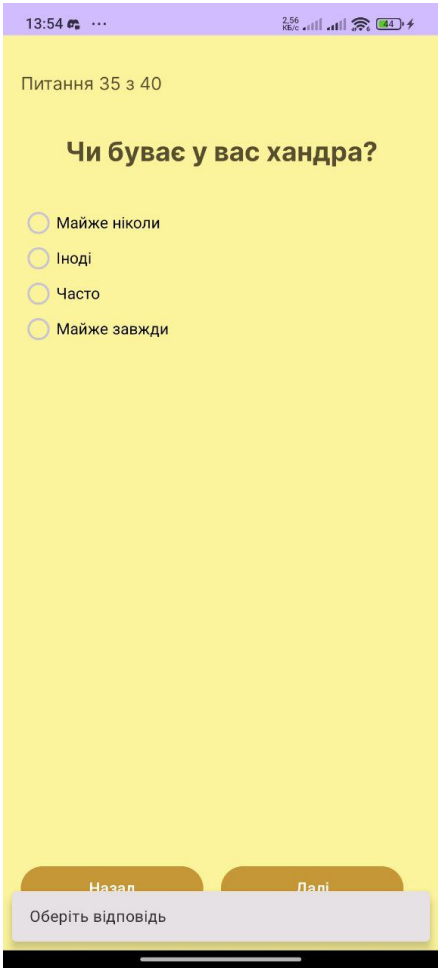

Рисунок 5.9 – Не обрана відповідь

2. Вихід з тесту:

– Кроки: Вибрати тест зі списку (якщо є вибір). Відповісти на кілька питань. Вийти з тесту (наприклад, натиснути кнопку "Назад" на пристрої).

– Очікуваний результат: Прогрес проходження тесту не зберігається.

5.3 Результати тестування

Знайдені дефекти та помилки

Під час тестування були виявлені наступні дефекти та помилки:

1. Авторизація:

– Некоректне повідомлення при введенні неправильного паролю: текст помилки був неінформативним ("Помилка" замість очікуваного "Користувач [логін] НЕ авторизован").

– Відсутність валідації на порожні поля, що дозволяло користувачам натискати кнопку авторизації без введення даних.

2. Реєстрація:

– Некоректне відображення повідомлення при введенні існуючого логіну: текст помилки був занадто загальним ("Помилка реєстрації" замість очікуваного "Користувач [логін] вже існує").

– Відсутність валідації на мінімальну довжину пароля, що дозволяло реєструвати користувачів з занадто короткими паролями.

3. Проходження тестів:

– Кнопка "Далі" не була деактивована при відсутності відповідей на обов'язкові питання, що дозволяло користувачам пропускати питання.

– Після закінчення тестування, питання починали йти спочатку, замість виведення результатів.

Усунення виявлених дефектів

Для усунення виявлених дефектів були виконані наступні дії:

1. Авторизація:

– Оновлено текст повідомлень про помилки для надання більш інформативного зворотного зв'язку.

– Додано валідацію на порожні поля, що блокує кнопку авторизації до введення даних.

2. Реєстрація:

– Оновлено текст повідомлень про помилки для відображення конкретної причини помилки (наприклад, існуючий логін).

– Додано перевірку мінімальної довжини пароля, що блокує реєстрацію при введенні занадто коротких паролів.

3. Проходження тестів:

– Додано логіку деактивації кнопки "Далі" при відсутності відповідей на обов'язкові питання.

– Додано логіку переходу до наступного вікна після відповіді на останє пітання.

Метрики якості

Для оцінки якості програмного забезпечення були використані наступні метрики:

1. Кількість знайдених дефектів: Загальна кількість виявлених дефектів під час тестування.

– Авторизація: 2 дефекти.

– Реєстрація: 2 дефекти.

– Проходження тестів: 2 дефекти.

2. Кількість усунених дефектів: Кількість дефектів, успішно виправлених після тестування.

– Авторизація: 2 дефекти виправлено.

– Реєстрація: 2 дефекти виправлено.

– Проходження тестів: 2 дефекти виправлено.

3. Час на усунення дефектів: Середній час, витрачений на виправлення кожного дефекту.

– Авторизація: 20 хвилин на дефект.

– Реєстрація: 15 хвилин на дефект.

– Проходження тестів: 30 хвилин на дефект.

4. Коефіцієнт повторного тестування: Відсоток дефектів, які не були виправлені з першої спроби та вимагали повторного втручання.

– Авторизація: 0% (всі дефекти виправлено з першої спроби).

– Реєстрація: 0% (всі дефекти виправлено з першої спроби).

– Проходження тестів: 0% (всі дефекти виправлено з першої спроби).

5. Задоволеність користувачів: Відгуки користувачів щодо зручності використання та якості додатку після виправлення дефектів.

– Користувачі відзначили підвищення зручності використання та зниження кількості помилок.

Ці метрики дозволили оцінити ефективність процесу тестування та усунення дефектів, а також загальну якість програмного забезпечення.

5.4 Висновки за розділом

У даному розділі робот представлено комплексний підхід до тестування Android додатку системи психологічної діагностики, розробленого з використанням мови програмування Kotlin в середовищі Android Studio. Забезпечення високої якості програмного забезпечення є критично важливим для надійного функціонування системи та задоволення потреб користувачів.

Робота охоплює різні види тестування, що дозволяє виявляти та усувати дефекти на різних рівнях додатку. Модульне (юніт) тестування забезпечує перевірку окремих компонентів, таких як класи, методи та функції, в ізольованому середовищі з використанням фреймворку JUnit та бібліотеки Mockito. Це допомагає швидко виявляти помилки на ранніх етапах розробки та забезпечує високу якість окремих компонентів.

Інтеграційне тестування перевіряє взаємодію між різними компонентами додатку та їх правильну інтеграцію за допомогою фреймворків Espresso та UIAutomator, що дозволяє імітувати дії користувача та перевіряти правильність відображення даних в інтерфейсі. Ці тести забезпечують виявлення помилок інтеграції та стабільність системи в цілому.

Системне тестування оцінює функціональність всього додатку з урахуванням взаємодії всіх його компонентів, перевіряючи такі аспекти, як функціональність, продуктивність, стабільність та сумісність з різними версіями Android. Системні тести можуть виконуватися як вручну, так і частково автоматизовано.

Важливим аспектом є тестування юзабіліті, яке забезпечує перевірку зручності використання додатку кінцевими користувачами. Залучаються реальні користувачі, які виконують певні завдання, в той час як спеціалісти спостерігають за їх діями та реакціями. Оцінюються такі аспекти, як легкість навігації, зрозумілість інтерфейсу та зручність основних функцій. Результати тестування дозволяють виявити проблеми в користувацькому інтерфейсі та внести необхідні покращення для підвищення загального досвіду роботи з додатком.

У роботі детально описано тестові сценарії для ключових функцій додатку, таких як авторизація, реєстрація та проходження психологічних тестів. Наведено приклади позитивних та негативних сценаріїв, а також очікувані результати для кожного випадку.

Представлено результати тестування, включаючи перелік виявлених дефектів та помилок, а також опис їх успішного усунення. Наведено метрики оцінки якості програмного забезпечення після виправлення дефектів, такі як кількість усунених дефектів, час на усунення, коефіцієнт повторного тестування та задоволеність користувачів. Ці метрики свідчать про ефективність процесу тестування та високу якість кінцевого продукту.

Таким чином, комплексний підхід до тестування, застосований у даній дипломній роботі, забезпечив ретельну перевірку коректності роботи окремих компонентів додатку, їх інтеграції, функціональності всієї системи, а також зручності використання для кінцевих користувачів. Виявлені дефекти та помилки були успішно усунені, що підтверджується відповідними метриками якості. Як результат, розроблений Android додаток системи психологічної діагностики відповідає високим стандартам якості, є стабільним, функціональним та зручним для користувачів.

ВИСНОВКИ

У даній дипломній роботі було розроблено сучасну систему психологічної діагностики у вигляді мобільного додатку для платформи Android. Проведений ретельний аналіз предметної області засвідчив необхідність створення такого програмного забезпечення з огляду на зростаючу популярність мобільних пристроїв та недоліки існуючих рішень.

Система реалізує комплексний набір психодіагностичних методик, включаючи опитувальники тривожності, депресії, агресивності та прокрастинації. Детально описано алгоритми реалізації цих методик з урахуванням специфіки Android-розробки. Застосовано сучасні підходи, зокрема архітектурний патерн MVVM, модульну структуру, принципи матеріального дизайну та ергономіки для забезпечення високої якості користувацького інтерфейсу.

Важливим аспектом розробки стало всебічне тестування програмного забезпечення на різних рівнях - модульне, інтеграційне, системне та тестування юзабіліті. Комплексне тестування дозволило виявити та усунути дефекти, забезпечивши високу якість, стабільність та зручність використання кінцевого продукту.

Розроблена система психологічної діагностики для ОС Android поєднує потужний функціонал, науково-обґрунтований набір методик, захист персональних даних, адаптацію під особливості мобільних пристроїв. Це дозволить значно підвищити доступність, конфіденційність та ефективність процесу психодіагностики, розширить коло користувачів відповідних послуг. Система відповідає сучасним вимогам та є готовою до масштабування й упровадження.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. MMPI-A-RF [Електронний ресурс]. URL: https://www.pearsonassessments.com/store/usassessments/en/Store/Professional-Assessments/Personality-%26-Biopsychosocial/Minnesota-Multiphasic-Personality-Inventory-Adolescent-Restructured-Form/p/100000465.html

2. MHS Talent Assessment [Електронний ресурс]. URL: https://talentassessments.com/

3. QPsychAssessment Suite [Електронний ресурс]. URL: https://www.psychcorp.co.uk/Psychology/AdultMentalHealth/QPsychAssessmentSuite/QPsychAssessmentSuite.aspx

4. TestCafe [Електронний ресурс]. URL: https://testcafe.io/

5. PsychologyTests [Електронний ресурс]. URL: https://psychologytests.com/

6. Psych Central [Електронний ресурс]. URL: https://psychcentral.com/

7. Robins, R. W., Fraley, R. C., & Krueger, R. F. (Eds.). (2009). Handbook of research methods in personality psychology. Guilford Press.

8. Groth-Marnat, G., & Wright, A. J. (2016). Handbook of psychological assessment. John Wiley & Sons.

9. Mertens, D. M. (2014). Research and evaluation in education and psychology: Integrating diversity with quantitative, qualitative, and mixed methods. Sage publications.

10. Корольчук, М. С. (2003). Психодіагностика: Навчальний посібник для студентів вищих навчальних закладів. Київ: Ельга, Ніка-Центр.

11. Мусіяка, Н. І. (2014). Психодіагностика: навчальний посібник. Львів: Видавництво Львівської політехніки.

12. Музика, О. О., Металіді, О. І., & Штика, Ю. М. (2020). Психодіагностика: Практикум. Київ: Слово.

13. Кокун, О. М. (2012). Психодіагностика: Навчальний посібник. Київ: Центр навчальної літератури.

14. оляко, В. О., Музика, О. Л., & Третяк, Т. М. (2020). Психодіагностика обдарованості та креативності: Навчальний посібник. Київ: Інститут обдарованої дитини НАПН України.

15. Корольчук, М. С., Осьодло, В. І., & Пенькова, О. І. (2018). Психодіагностика: Навчальний посібник. Київ: Київський національний торговельно-економічний університет.

16. Грицук, О. В. (2018). Психодіагностика: Навчальний посібник. Харків: Видавництво Іванченка І.С.

ДОДАТОК А

Зауваження нормоконтролера

Позначення документа	Сторінка	Зміст зауваження
ЗМІСТ		Використані службові слова «Розділ» в змісті татексті ПЗ та зміст не 14 шрифт
ПЗ: с. 13-14, 18-26, …		ПЗ містить багато помилок. Невірні переліки, порожні рядки та відсутність розділових знаків…
С.70-71		Перелік літератури та посилання на літературу містить помилки.

ДОДАТОК В

Лістинг програми

MainActivity.kt

```
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        val userLogin: EditText = findViewById(R.id.user_login)
        val userPass: EditText = findViewById(R.id.user_password)
        val buttonReg: Button = findViewById(R.id.button_auth)
        val linkToAuth: TextView = findViewById(R.id.link_to_auth)

        linkToAuth.setOnClickListener {
            val intent = Intent(this, AuthActivaty::class.java)
            startActivity(intent)
        }

        buttonReg.setOnClickListener {
            val login = userLogin.text.toString().trim()
            val pass = userPass.text.toString().trim()

            if (login == "" || pass == "") {
                Toast.makeText(this, "Не всі поля заповнені!!!",
Toast.LENGTH_LONG).show()
            } else {
                val db = DbHelper(this, null)

                if (db.isLoginExists(login)) {
                    Toast.makeText(this, "Логін $login вже зайнятий!",
Toast.LENGTH_LONG).show()
                } else {
                    val user = User(login, pass)
                    db.addUser(user)
                    Toast.makeText(this, "Користувач $login зареєстрован",
Toast.LENGTH_LONG).show()
                    userLogin.text.clear()
                    userPass.text.clear()

                    val intent_main_auth = Intent(this,
AuthActivaty::class.java)
                    startActivity(intent_main_auth)
                }
```

```
            }
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#FCF49C"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/inter_bold"
        android:textColor="#574F2F"
        android:layout_gravity="center"
        android:layout_marginTop="200dp"
        android:textSize="30sp"
        android:textStyle="bold"
        android:text="Створіть аккаунт" />

    <EditText
        android:id="@+id/user_login"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="text"
        android:textColorHint="#6A6A6A"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:textColor="#574F2F"
        android:hint="Логін" />

    <EditText
        android:id="@+id/user_password"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword"
        android:textColorHint="#6A6A6A"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:textColor="#574F2F"
        android:hint="Пароль" />

    <Button
        android:id="@+id/button_auth"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:layout_gravity="center"
        android:backgroundTint="#C69736"
        android:textColor="#FFFFFF"
        android:textStyle="bold"
```

```
        android:textSize="18sp"
        android:text="Реєстрація" />

    <TextView
        android:id="@+id/link_to_auth"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="120dp"
        android:fontFamily="@font/inter"
        android:text="Увійти"
        android:textColor="#574F2F"
        android:textSize="17sp" />

</LinearLayout>
```

AuthActivaty.kt

```
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class AuthActivaty : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_auth_activaty)

        val userLoginAuth: EditText = findViewById(R.id.user_login_auth)
        val userPassAuth: EditText = findViewById(R.id.user_password_auth)
        val button: Button = findViewById(R.id.button)
        val linkToReg: TextView = findViewById(R.id.link_to_reg)

        linkToReg.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }

        button.setOnClickListener {
            val login = userLoginAuth.text.toString().trim()
            val pass = userPassAuth.text.toString().trim()

            if (login == "" || pass == "")
                Toast.makeText(this, "Не всі поля заповнені!!!",
Toast.LENGTH_LONG).show()
            else {
                val db = DbHelper(this, null)
                val isAush = db.getUser(login, pass)

                if (isAush) {
                    Toast.makeText(this, "Користувач $login авторизован",
```

```
Toast.LENGTH_LONG).show()
                    val intent = Intent(this, MainMenu::class.java)
                    startActivity(intent)
                } else {
                    Toast.makeText(this, "Користувач $login НЕ авторизован",
Toast.LENGTH_LONG)
                        .show()
                }
            }
        }
    }
}
```

activity_auth_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#FCF49C"
    tools:context=".AuthActivaty">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/inter_bold"
        android:textColor="#574F2F"
        android:layout_gravity="center"
        android:layout_marginTop="200dp"
        android:textSize="30sp"
        android:textStyle="bold"
        android:text="Авторізація" />

    <EditText
        android:id="@+id/user_login_auth"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="text"
        android:textColorHint="#6A6A6A"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:textColor="#574F2F"
        android:hint="Введіть логін" />

    <EditText
        android:id="@+id/user_password_auth"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword"
        android:textColorHint="#6A6A6A"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:textColor="#574F2F"
        android:hint="Введіть пароль" />

    <Button
```

```
        android:id="@+id/button"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:layout_gravity="center"
        android:backgroundTint="#C69736"
        android:textColor="#FFFFFF"
        android:textStyle="bold"
        android:textSize="18sp"
        android:text="Увійти" />

    <TextView
        android:id="@+id/link_to_reg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="120dp"
        android:fontFamily="@font/inter"
        android:text="Зареєструватись"
        android:textColor="#574F2F"
        android:textSize="17sp" />


</LinearLayout>
```

DbHelper.kt

```
package com.example.myapplication

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DbHelper(val context: Context, val factory:
SQLiteDatabase.CursorFactory?) :
    SQLiteOpenHelper(context, "app", factory, 1) {
    override fun onCreate(db: SQLiteDatabase?) {
        val query = "CREATE TABLE users (id INT PRIMARY KEY, login TEXT, pass
TEXT)"
        db!!.execSQL(query)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion:
Int) {
        db!!.execSQL("DROP TABLE IF EXISTS users")
        onCreate(db)
    }

    fun addUser(user: User) {
        val values = ContentValues()
        values.put("login", user.login)
        values.put("pass", user.pass)

        val db = this.writableDatabase
        db.insert("users", null, values)

        db.close()
    }

    fun getUser(login: String, pass: String): Boolean {
        val db = this.readableDatabase
```

```
        val result = db.rawQuery("SELECT * FROM users WHERE login = '$login'
AND pass = '$pass'", null)
        return result.moveToFirst()
    }

    fun isLoginExists(login: String): Boolean {
        val db = this.readableDatabase
        val cursor = db.rawQuery("SELECT * FROM users WHERE login =
'$login'", null)
        val exists = cursor.count > 0
        cursor.close()
        return exists
    }

}
```

User.kt

```
package com.example.myapplication

class User(val login: String, val pass: String) {
}
```

MainMenu.kt

```
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainMenu : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main_menu)

        val buttonAnxiety: Button = findViewById(R.id.button_anxiety)
        val buttonDepression: Button = findViewById(R.id.button_depression)
        val buttonAgression: Button = findViewById(R.id.button_aggression)
        val buttonProcrastination: Button =
findViewById(R.id.button_procrastination)

        buttonAnxiety.setOnClickListener {
            val intent = Intent(this, Anxiety::class.java)
            startActivity(intent)
        }

    }
}
```

activity_main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FCF49C"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_marginTop="60dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="@font/inter_bold"
        android:text="Оберіть тест"
        android:textColor="#574F2F"
        android:textSize="40sp"
        android:textStyle="bold"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="98dp"
        tools:layout_editor_absoluteY="72dp" />

    <Button
        android:id="@+id/button_procrastination"
        android:layout_width="380dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="30dp"
        android:background="@drawable/button_selector"
        android:text="Шкала прокрастинації Такмана"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="bold"
        app:backgroundTint="@null"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="44dp"
        tools:layout_editor_absoluteY="213dp" />

    <Button
        android:id="@+id/button_anxiety"
        android:layout_width="380dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:background="@drawable/button_selector"
        android:text="Шкала тривожності Спілбергера-Ханіна Призначення"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="bold"
        app:backgroundTint="@null"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="322dp"
        tools:layout_editor_absoluteY="273dp" />

    <Button
        android:id="@+id/button_depression"
        android:layout_width="380dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:background="@drawable/button_selector"
        android:text="
Опитувач депресії Бека Призначення: Оцінка наявності та ступеня виразності
```

```
депресивних симптомів"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="bold"
        app:backgroundTint="@null"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="50dp"
        tools:layout_editor_absoluteY="403dp" />

    <Button
        android:id="@+id/button_aggression"
        android:layout_width="380dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_gravity="center"
        android:background="@drawable/button_selector"
        android:text="Опитувальник Басса-Дарки Призначення: Діагностика
агресивних та ворожих реакцій"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        app:backgroundTint="@null"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="318dp"
        tools:layout_editor_absoluteY="508dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="@font/inter_bold"
        android:layout_marginTop="100dp"
        android:text="Так було задумано. Не дивуйтесь:>"
        android:textColor="#574F2F"
        android:textSize="12sp"
        android:textStyle="bold"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="119dp"
        tools:layout_editor_absoluteY="785dp" />

</LinearLayout>
```

Anxiety.kt

```
package com.example.myapplication

import android.content.Intent
import android.graphics.Color
import android.os.Bundle
import android.util.Log
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.RadioButton
import android.widget.RadioGroup
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.snackbar.Snackbar

class Anxiety : AppCompatActivity() {

    private lateinit var questionTextView: TextView
    private lateinit var questionNumberTextView: TextView
```

```
    private lateinit var answerRadioGroup: RadioGroup
    private lateinit var nextButton: Button
    private lateinit var prevButton: Button
    private var scoreCT0 = 0
    private var scoreCT1 = 0
    private var scoreOT0 = 0
    private var scoreOT1 = 0

    private data class Question(val text: String, val type: String, val
correctAnswerIndex: Int)
    private data class Answer(
        val questionIndex: Int,
        val selectedAnswerIndex: Int,
        val scoreCT0Delta: Int,
        val scoreCT1Delta: Int,
        val scoreOT0Delta: Int,
        val scoreOT1Delta: Int
    )

    private val questions = listOf(
        Question("Чи почуваєтеся ви спокійним?", "CT", 1),
        Question("Чи нічого вам не загрожує?", "CT", 1),
        Question("Чи перебуваєте ви в напрузі?", "CT", 0),
        Question("Чи відчуваєте ви жаль?", "CT", 0),
        Question("Чи почуваєтеся ви вільно?", "CT", 1),
        Question("Чи засмучені ви?", "CT", 0),
        Question("Чи хвилюють вас можливі невдачі?", "CT", 0),
        Question("Чи почуваєтеся ви відпочилим?", "CT", 1),
        Question("Чи стривожені ви?", "CT", 0),
        Question("Чи відчуваєте ви почуття внутрішнього задоволення?", "CT",
1),
        Question("Чи впевнені ви у собі?", "CT", 1),
        Question("Чи нервуєте ви?", "CT", 0),
        Question("Чи не знаходите ви собі місця?", "CT", 0),
        Question("Чи напружені ви?", "CT", 0),
        Question("Чи не відчуваєте ви скутості, напруженості?", "CT", 1),
        Question("Чи задоволені ви?", "CT", 1),
        Question("Чи стурбовані ви?", "CT", 0),
        Question("Чи надто збуджені ви і чи незручно вам?", "CT", 0),
        Question("Чи радісно вам?", "CT", 1),
        Question("Чи приємно вам?", "CT", 1),
        Question("Чи відчуваєте ви задоволення?", "OT", 1),
        Question("Чи швидко ви втомлюєтесь?", "OT", 0),
        Question("Чи легко ви можете заплакати?", "OT", 0),
        Question("Чи хотіли б ви бути таким же щасливим, як і інші?", "OT",
0),
        Question("Чи часто ви програєте через те, що недостатньо швидко
приймаєте рішення?", "OT", 0),
        Question("Чи почуваєтеся ви зазвичай бадьорим?", "OT", 1),
        Question("Чи спокійні, холоднокровні та зібрані ви?", "OT", 1),
        Question("Чи дуже засмучують вас очікувані труднощі?", "OT", 0),
        Question("Чи занадто переживаєте ви через дрібниці?", "OT", 0),
        Question("Чи щасливі ви?", "OT", 1),
        Question("Чи приймаєте ви все занадто близько до серця?", "OT", 0),
        Question("Чи бракує вам впевненості в собі?", "OT", 0),
        Question("Чи почуваєтеся ви зазвичай у безпеці?", "OT", 1),
        Question("Чи намагаєтеся ви уникати критичних ситуацій і труднощів?",
"OT", 0),
        Question("Чи буває у вас хандра?", "OT", 0),
        Question("Я задоволений", "OT", 1),
        Question("Чи відволікають і хвилюють вас всякі дрібниці?", "OT", 0),
        Question("Чи сильно переживаєте ви свої розчарування, що потім довго
не можете про них забути?","OT",0),
        Question("Чи врівноважена ви людина?", "OT", 1),
```

```
        Question("Чи охоплює вас сильне занепокоєння, коли ви думаєте про
свої справи та турботи?","OT",0)
    )

    private val answerHistory = mutableListOf<Answer>()

    private var currentQuestionIndex = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_anxiety)

        questionNumberTextView = findViewById(R.id.questionNumberTextView)
        questionTextView = findViewById(R.id.questionTextView)
        answerRadioGroup = findViewById(R.id.answerRadioGroup)
        nextButton = findViewById(R.id.nextButton)
        prevButton = findViewById(R.id.prevButton)

        updateQuestion()

        nextButton.setOnClickListener {
            val selectedRadioButtonId = answerRadioGroup.checkedRadioButtonId

            if (selectedRadioButtonId == -1) {

                Snackbar.make(
                    findViewById(R.id.anxietyLinearLayout),
                    "Оберіть відповідь",
                    Snackbar.LENGTH_SHORT
                ).show()
            } else { // Ответ выбран
                val selectedAnswerIndex =
answerRadioGroup.indexOfChild(findViewById(selectedRadioButtonId))
                val currentQuestion = questions[currentQuestionIndex]


                var scoreCT0Delta = 0
                var scoreCT1Delta = 0
                var scoreOT0Delta = 0
                var scoreOT1Delta = 0

                when (currentQuestion.type) {
                    "CT" -> {
                        if (currentQuestion.correctAnswerIndex == 0) {
                            scoreCT0 += selectedAnswerIndex + 1
                            scoreCT0Delta = selectedAnswerIndex + 1
                        } else {
                            scoreCT1 += 4 - selectedAnswerIndex
                            scoreCT1Delta = 4 - selectedAnswerIndex
                        }
                    }
                    "OT" -> {
                        if (currentQuestion.correctAnswerIndex == 0) {
                            scoreOT0 += selectedAnswerIndex + 1
                            scoreOT0Delta = selectedAnswerIndex + 1
                        } else {
                            scoreOT1 += 4 - selectedAnswerIndex
                            scoreOT1Delta = 4 - selectedAnswerIndex
                        }
                    }
                }

                answerHistory.add(
                    Answer(
```

```
                        currentQuestionIndex,
                        selectedAnswerIndex,
                        scoreCT0Delta,
                        scoreCT1Delta,
                        scoreOT0Delta,
                        scoreOT1Delta
                    )
                )

                if (currentQuestionIndex < questions.size - 1) {
                    updateQuestion()
                } else {
                     showResults()
                }


                if (currentQuestionIndex < questions.size - 1) {
                    currentQuestionIndex++
                }
            }
            answerRadioGroup.clearCheck()
        }

        prevButton.setOnClickListener {
            if (currentQuestionIndex > 0 && answerHistory.isNotEmpty()) {

                val lastAnswer = answerHistory.removeLast()
                scoreCT0 -= lastAnswer.scoreCT0Delta
                scoreCT1 -= lastAnswer.scoreCT1Delta
                scoreOT0 -= lastAnswer.scoreOT0Delta
                scoreOT1 -= lastAnswer.scoreOT1Delta

                currentQuestionIndex = (currentQuestionIndex -
1).coerceAtLeast(0)
                updateQuestion()
            }
        }
    }

    private fun updateQuestion() {
        val currentQuestion = questions[currentQuestionIndex]
        questionTextView.text = currentQuestion.text


        answerRadioGroup.removeAllViews()
        val answerOptions = if (currentQuestion.type == "CT") {
            listOf("Ні, це не так", "Мабудь так", "Вірно", "Абсолютно вірно")
        } else { // "OT"
            listOf("Майже ніколи", "Іноді", "Часто", "Майже завжди")
        }

        questionNumberTextView.text = "Питання ${currentQuestionIndex + 1} з
${questions.size}"

        for (i in answerOptions.indices) {
            val radioButton = RadioButton(this)
            radioButton.text = answerOptions[i]
            radioButton.id = View.generateViewId()

            val parent = radioButton.parent as? ViewGroup
            parent?.removeView(radioButton)

            answerRadioGroup.addView(radioButton)
```

```
            radioButton.post {
                radioButton.setTextColor(Color.BLACK)
            }
        }
    }

    private fun showResults() {
        val situationalAnxiety = 35 + scoreCT0 - scoreCT1
        val personalAnxiety = 35 + scoreOT0 - scoreOT1

        // Создаем Intent для запуска ResultActivity
        val intent = Intent(this, res_anxiety::class.java)

        // Передаем результаты в Intent
        intent.putExtra("situationalAnxiety", situationalAnxiety)
        intent.putExtra("personalAnxiety", personalAnxiety)

        // Запускаем ResultActivity
        startActivity(intent)
    }
}
```

activity_anxiety.xml

```
<?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"
    android:id="@+id/anxietyLinearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FCF49C"
    android:padding="16dp">

    <TextView
        android:id="@+id/questionNumberTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Питання 1 з 10"
        android:textColor="#574F2F"
        android:fontFamily="@font/inter"
        android:textSize="16sp"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"/>

    <TextView
        android:id="@+id/questionTextView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Питання"
        android:textColor="#574F2F"
        android:fontFamily="@font/inter_bold"
        android:textSize="24sp"
        android:gravity="center"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toBottomOf="@id/questionNumberTextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <RadioGroup
```

```
        android:id="@+id/answerRadioGroup"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginTop="32dp"
        android:textColor="#000000"
        app:layout_constraintTop_toBottomOf="@id/questionTextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

    </RadioGroup>

    <Button
        android:id="@+id/prevButton"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Назад"
        android:textColor="#FFFFFF"
        android:backgroundTint="#C69736"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@id/nextButton"/>

    <Button
        android:id="@+id/nextButton"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Далі"
        android:textColor="#FFFFFF"
        android:backgroundTint="#C69736"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="@id/prevButton"
        app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

res_anxiety.kt

```
package com.example.myapplication

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class res_anxiety : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_res_anxiety)

        val situationalAnxietyTextView =
findViewById<TextView>(R.id.situationalAnxietyTextView)
        val personalAnxietyTextView =
findViewById<TextView>(R.id.personalAnxietyTextView)


        val situationalAnxiety = intent.getIntExtra("situationalAnxiety", 0)
        val personalAnxiety = intent.getIntExtra("personalAnxiety", 0)
```

```
        situationalAnxietyTextView.text = when {
            situationalAnxiety <= 30 -> "Низький рівень ситуативної
тривожності:\n" +
                    "Низький рівень ситуативної, або реактивної, тривожності
свідчить про загальний спокій та душевний комфорт особистості в даний
конкретний момент часу. Така людина не відчуває надмірного хвилювання,
напруженості чи занепокоєння через якусь певну ситуацію чи подію. Навпаки,
вона залишається врівноваженою, розслабленою та впевненою у своїх силах.
Низький рівень ситуативної тривожності допомагає зосередитись, ефективно
виконувати поставлені завдання та приймати зважені рішення, не піддаючись
імпульсивним емоційним реакціям. Проте, слід пам'ятати, що повна відсутність
ситуативної тривожності також може бути ознакою байдужості чи
незацікавленості ситуацією. Тому, оптимальним є помірний рівень ситуативного
хвилювання, який мобілізує ресурси особистості."
            situationalAnxiety in 31..44 -> "Середній рівень ситуативної
тривожності:\n" +
                    "Середній рівень ситуативної тривожності є цілком
нормальним та очікуваним станом для більшості людей у певних важливих
життєвих ситуаціях. Відчуття помірного хвилювання, неспокою чи напруженості
перед відповідальним завданням, іспитом, важливою зустріччю тощо є звичайною
реакцією організму на стресовий чинник. Такий рівень тривожності мобілізує
наші внутрішні ресурси, загострює увагу та пильність, що допомагає
підготуватись та сконцентруватись на майбутній події. Проте, якщо ситуативна
тривожність залишається на середньому рівні протягом тривалого часу, це може
призводити до зниження працездатності, роздратованості, порушень сну та інших
негативних наслідків. Тому важливо контролювати цей стан та вживати заходів
для його зменшення після завершення ситуації стресу."
            else -> "Високий рівень ситуативної тривожності:\n" +
                    "Високий рівень ситуативної тривожності свідчить про
надмірне, часто невиправдане, занепокоєння та перебільшене хвилювання щодо
певної конкретної ситуації чи події. Людина з високою ситуативною тривожністю
перебуває у стані сильного емоційного збудження, відчуває значний дискомфорт,
напругу, може страждати від нав'язливих думок та побоювань. Їй важко
зосередитись та раціонально мислити через постійні тривожні переживання та
сумніви. На фізичному рівні ця людина може відчувати прискорене серцебиття,
нудоту, тремтіння, запаморочення тощо. Високий рівень ситуативної тривожності
істотно знижує ефективність та продуктивність діяльності, негативно впливає
на міжособистісну взаємодію. Така людина потребує психологічної допомоги у
опануванні надмірної тривожності за допомогою психотерапевтичних та
поведінкових технік."
        }


        personalAnxietyTextView.text = when {
            personalAnxiety <= 30 -> "Низький рівень особистісної
тривожності:\n" +
                    "Низький рівень особистісної тривожності є ознакою
емоційної стійкості, врівноваженості та спокою як стійкої риси характеру
людини. Такі люди зазвичай не схильні до надмірного хвилювання, вони
залишаються холоднокровними та витриманими навіть в стресових ситуаціях. Вони
добре контролюють свої емоції, не піддаються паніці чи зневірі під впливом
негативних подій чи невдач. Низька особистісна тривожність допомагає
особистості бути впевненою у собі, ефективно справлятися з викликами, не
боятись труднощів та змін. Водночас, слід пам'ятати, що занадто низький
рівень особистісної тривоги може бути ознакою емоційної холодності,
байдужості та нечутливості до проблем оточуючих. Тому, оптимальним є помірний
рівень особистісної тривожності."
            personalAnxiety in 31..44 -> "Середній рівень особистісної
тривожності:\n" +
                    "Помірний, або середній, рівень особистісної тривожності
є типовим для більшості людей. Відчуваючи час від часу неспокій, сумніви,
легке занепокоєння, проте зберігаючи здатність контролювати ці емоційні стани
```

```
та опановувати їх за допомогою власного досвіду чи соціальної підтримки.
Людина із середнім рівнем особистісної тривожності здатна реагувати на
стресові події гнучко та конструктивно, не впадаючи в апатію чи панічний
страх. Помірна тривожність допомагає їй залишатись пильною, обережною та
передбачливою, проте без надмірного драматизму. В міжособистісних стосунках
такі люди спокійні, доброзичливі, проте можуть іноді бути дещо невпевненими.
В цілому, середній рівень особистісної тривоги є адаптивним та сприятливим
для нормального функціонування."
            else -> "Високий рівень особистісної тривожності:\n" +
                    "Високий рівень особистісної тривожності є стійкою рисою
характеру людини, що проявляється постійним відчуттям загрози, невпевненості,
схвильованості та неспокою. Така людина надміру реагує на будь-які незначні
стреси, проблемні ситуації чи дрібниці. Її переслідують невротичні страхи,
побоювання, нав'язливі думки про можливі негативні події. Звичайні життєві
труднощі вона сприймає як непереборні перешкоди, постійно очікує невдачі та
катастрофи. Внаслідок цього особа стає вкрай нерішучою, невпевненою у собі,
вразливою, легко впадає у безпорадний стан. Високий рівень особистісної
тривожності призводить до частих страхів, фобій, обсессивно-компульсивних
розладів та інших емоційних порушень. Така людина потребує кваліфікованої
психотерапевтичної допомоги для подолання свого стану та навчання прийомам
керування тривожністю."
        }
    }
}
```

res_anxiety.xml

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FCF49C"
    android:padding="16dp">

    <TextView
        android:id="@+id/resultTitleTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Результати тесту"
        android:textColor="#574F2F"
        android:fontFamily="@font/inter_bold"
        android:textSize="24sp"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

    <TextView
        android:id="@+id/situationalAnxietyTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ситуативна тривожність: "
        android:textColor="#574F2F"
        android:fontFamily="@font/inter"
        android:textSize="18sp"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toBottomOf="@id/resultTitleTextView"
        app:layout_constraintStart_toStartOf="parent"/>

    <TextView
```

```
        android:id="@+id/personalAnxietyTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Особиста тривожність: "
        android:textColor="#574F2F"
        android:fontFamily="@font/inter"
        android:textSize="18sp"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toBottomOf="@id/situationalAnxietyTextView"
        app:layout_constraintStart_toStartOf="parent"/>

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

ДОДАТОК В

Питання та ключі до тестів

ПИТАННЯ ТА КЛЮЧІ ДО ТЕСТУ "ШКАЛА ТРИВОЖНОСТІ" (Тест 1):

Ситуативна тривожність (СТ):

1. Я спокійний (Зворотній рахунок)
2. Мені нічого не загрожує (Зворотній рахунок)
3. Я перебуваю в напрузі (Прямий рахунок)
4. Я відчуваю жаль (Прямий рахунок)
5. Я почуваюся вільно (Зворотній рахунок)
6. Я засмучений (Прямий рахунок)
7. Мене хвилюють можливі невдачі (Прямий рахунок)
8. Я почуваюся відпочилим (Зворотній рахунок)
9. Я стривожений (Прямий рахунок)
10. Я відчуваю почуття внутрішнього задоволення (Зворотній рахунок)
11. Я впевнений у собі (Зворотній рахунок)
12. Я нервую (Прямий рахунок)
13. Я не знаходжу собі місця (Прямий рахунок)
14. Я напружений (Прямий рахунок)
15. Я не відчуваю скутості, напруженості (Зворотній рахунок)
16. Я задоволений (Зворотній рахунок)
17. Я стурбований (Прямий рахунок)
18. Я надто збуджений і мені ніяково (Прямий рахунок)
19. Мені радісно (Зворотній рахунок)
20. Мені приємно (Зворотній рахунок)

Особистісна тривожність (ОТ):

21. Я відчуваю задоволення (Зворотній рахунок)
22. Я дуже швидко втомлююсь (Прямий рахунок)
23. Я легко можу заплакати (Прямий рахунок)
24. Я хотів би бути таким же щасливим, як і інші (Прямий рахунок)
25. Нерідко я програю через те, що недостатньо швидко приймаю рішення (Прямий рахунок)
26. Зазвичай я почуваюся бадьорим (Зворотній рахунок)
27. Я спокійний, холоднокровний і зібраний (Зворотній рахунок)

28. Очікувані труднощі зазвичай дуже турбують мене (Прямий рахунок)
29. Я надто переживаю через дрібниці (Прямий рахунок)
30. Я цілком щасливий (Зворотній рахунок)
31. Я приймаю все надто близько до серця (Прямий рахунок)
32. Мені бракує впевненості в собі (Прямий рахунок)
33. Зазвичай я почуваюся в безпеці (Зворотній рахунок)
34. Я намагаюся уникати критичних ситуацій і труднощів (Прямий рахунок)
35. У мене буває хандра (Прямий рахунок)
36. Я задоволений (Зворотній рахунок)
37. Всілякі дрібниці відволікають і хвилюють мене (Прямий рахунок)
38. Я так сильно переживаю свої розчарування, що потім довго не можу про них забути (Прямий рахунок)
39. Я врівноважена людина (Зворотній рахунок)
40. Мене охоплює сильне занепокоєння, коли я думаю про свої справи та турботи (Прямий рахунок)

Пам'ятайте, що при обробці результатів для запитань із зворотнім рахунком потрібно використовувати формулу: 5 - отриманий бал.

ПИТАННЯ ТА КЛЮЧІ: Опитувальник депресії Бека (тест 2)

Список 21 групи тверджень опитувальника депресії Бека з відповідними балами для кожного твердження. Пам'ятайте, що в кожній групі користувач повинен обрати лише одне твердження, яке найбільше відповідає його стану.

1. Настрій:

0 - Я не почуваюся засмученим, сумним.

1 - Я засмучений.

2 - Я весь час засмучений і не можу від цього відключитися.

3 - Я настільки засмучений і нещасний, що не можу це витримати.

2. Песимізм:

0 - Я не турбуюся про своє майбутнє.

1 - Я відчуваю, що стурбований майбутнім.

2 - Я відчуваю, що мене нічого не чекає в майбутньому.

3 - Моє майбутнє безнадійне, і нічого не може змінитися на краще.

3. Почуття невдачі:

0 - Я не відчуваю себе невдахою.

1 - Я відчуваю, що зазнав більше невдач, ніж інші люди.

2 - Коли я оглядаюся на своє життя, я бачу в ньому багато невдач.

3 - Я відчуваю, що як особистість я - повний невдаха.

4. Незадоволеність собою:

0 - Я отримую стільки ж задоволення від життя, як і раніше.

1 - Я не отримую стільки ж задоволення від життя, як раніше.

2 - Я більше не отримую задоволення ні від чого.

3 - Я повністю незадоволений життям і мені все набридло.

5. Почуття провини:

0 - Я не відчуваю себе винуватим у чомусь.

1 - Досить часто я відчуваю себе винуватим.

2 - Більшу частину часу я відчуваю себе винуватим.

3 - Я постійно відчуваю почуття провини.

6. Відчуття покарання:

0 - Я не відчуваю, що можу бути покараним за щось.

1 - Я відчуваю, що можу бути покараним.

2 - Я очікую, що можу бути покараним.

3 - Я відчуваю себе вже покараним.

7. Самозневага:

0 - Я не розчарований у собі.

1 - Я розчарований у собі.

2 - Я сам собі огидний.

3 - Я себе ненавиджу.

8. Самозвинувачення:

0 - Я знаю, що я не гірший за інших.

1 - Я критикую себе за помилки та слабкості.

2 - Я весь час звинувачую себе у своїх вчинках.

3 - Я звинувачую себе в усьому поганому, що відбувається.

9. Суїцидальні думки:

0 - Я ніколи не думав покінчити життя самогубством.

1 - До мене приходять думки покінчити життя самогубством, але я не буду їх здійснювати.

2 - Я хотів би покінчити життя самогубством.

3 - Я вбив би себе, якби трапилася нагода.

10. Плаксивість:

0 - Я плачу не більше, ніж зазвичай.

1 - Зараз я плачу частіше, ніж раніше.

2 - Тепер я увесь час плачу.

3 - Раніше я міг плакати, а тепер не можу, навіть якщо мені хочеться.

11. Дратівливість:

0 - Зараз я не більш дратівливий, ніж зазвичай.

1 - Я легше дратуюся, ніж раніше.

2 - Тепер я постійно відчуваю, що дратований.

3 - Я став байдужим до речей, які мене раніше дратували.

12. Порушення соціальних зв'язків:

0 - Я не втратив інтересу до інших людей.

1 - Я менше цікавлюся іншими людьми, ніж раніше.

2 - Я майже втратив інтерес до інших людей.

3 - Я повністю втратив інтерес до інших людей.

13. Нерішучість:

0 - Я відкладаю прийняття рішення іноді, як і раніше.

1 - Я частіше, ніж раніше, відкладаю прийняття рішення.

2 - Мені важче приймати рішення, ніж раніше.

3 - Я більше не можу приймати рішень.

14. Образ тіла:

0 - Я не відчуваю, що виглядаю гірше, ніж зазвичай.

1 - Мене турбує те, що я виглядаю старим і непривабливим.

2 - Я знаю, що в моїй зовнішності відбулися істотні зміни, що роблять мене непривабливим.

3 - Я знаю, що виглядаю потворно.

15. Труднощі в роботі:

0 - Я можу працювати так само добре, як і раніше.

1 - Мені необхідно докласти додаткових зусиль, щоб почати щось робити.

2 - Я з труднощами змушую себе щось робити.

3 - Я зовсім не можу виконувати жодну роботу.

16. Порушення сну:

0 - Я сплю так само добре, як і раніше.

1 - Зараз я сплю гірше, ніж раніше.

2 - Я прокидаюся на 1-2 години раніше, і мені важко заснути знову.

3 - Я прокидаюся на кілька годин раніше звичайного і більше не можу заснути.

17. Втомлюваність:

0 - Я втомлююся не більше, ніж зазвичай.

1 - Тепер я втомлююся швидше, ніж раніше.

2 - Я втомлююся майже від усього, що роблю.

3 - Я не можу нічого робити через втому.

18. Втрата апетиту:

0 - Мій апетит не гірший, ніж зазвичай.

1 - Мій апетит став гірший, ніж раніше.

2 - Мій апетит тепер значно гірший.

3 - У мене взагалі немає апетиту.

19. Втрата ваги:

0 - Останнім часом я не схуд або втрата ваги була незначною.

1 - За останній час я втратив більше 2 кг.

2 - Я втратив більше 5 кг.

3 - Я втратив більше 7 кг.

20. Зайнятість тілесними відчуттями:

0 - Я турбуюся про своє здоров'я не більше, ніж зазвичай.

1 - Мене хвилюють проблеми мого фізичного здоров'я, такі як біль, розлад шлунку, запори тощо.

2 - Я дуже стурбований своїм фізичним станом, і мені важко думати про що-небудь інше.

3 - Я настільки стурбований своїм фізичним станом, що більше ні про що не можу думати.

21. Втрата сексуального потягу:

0 - Я не помічав будь-яких змін у моїх сексуальних інтересах.

1 - Мене менше цікавлять проблеми сексу, ніж раніше.

2 - Зараз я значно менше цікавлюся сексуальними проблемами, ніж раніше.

3 - Я повністю втратив сексуальний інтерес.

ПИТАННЯ ТА КЛЮЧІ: Басс - Дарки (Тест 3)

1. Часом я не можу впоратися з бажанням завдати шкоди іншим (Так)

2. Іноді поплітковую про людей, яких не люблю (Так)

3. Я легко дратуюся, але швидко заспокоююся (Так)

4. Якщо мене не попросять по-хорошому, я не виконаю (Так)

5. Я не завжди отримую те, що мені належить (Так)

6. Я не знаю, що люди кажуть про мене за моєю спиною (Так)

7. Якщо я не схвалюю поведінку друзів, я даю їм це відчути (Так)

8. Коли мені траплялося когось обманути, я відчував мучительні докори сумління (Так)

9. Мені здається, що я не здатний вдарити людину (Ні)

10. Я ніколи не дратуюся настільки, щоб кидатися предметами (Ні)

11. Я завжди поблажливий до чужих вад (Ні)

12. Якщо мені не подобається встановлене правило, мені хочеться його порушити (Так)

13. Інші вміють майже завжди користуватися сприятливими обставинами (Так)

14. Я тримался насторожі з людьми, які ставляться до мене дещо дружніше, ніж я очікував (Так)

15. Я часто буваю не згоден з людьми (Так)

16. Іноді мені спадають на думку думки, яких я соромлюся (Так)

17. Якщо хтось перший вдарить мене, я не відповім йому (Ні)

18. Коли я дратуюся, я хлопаю дверима (Так)

19. Я набагато більш дратівливий, ніж здається (Так)

20. Якщо хтось уявляє себе начальником, я завжди роблю йому наперекір (Так)

21. Мене трохи засмучує моя доля (Так)

22. Я думаю, що багато людей не люблять мене (Так)

23. Я не можу утриматися від суперечки, якщо люди не погоджуються зі мною (Так)

24. Люди, які ухиляються від роботи, повинні відчувати провину (Так)

25. Той, хто ображає мене і мою сім'ю, напрошується на бійку (Так)

26. Я не здатний на грубі жарти (Ні)

27. Мене охоплює лють, коли наді мною насміхаються (Так)

28. Коли люди будують із себе начальників, я роблю все, щоб вони не зазнавалися (Так)

29. Майже щотижня я бачу когось, хто мені не подобається (Так)

30. Досить багато людей заздрять мені (Так)

31. Я вимагаю, щоб люди поважали мене (Так)

32. Мене гнітить те, що я мало роблю для своїх батьків (Так)

33. Люди, які постійно мучать вас, заслуговують на те, щоб їх "штовхнули в носа" (Так)

34. Я ніколи не буваю похмурим від злості (Ні)

35. Якщо до мене ставляться гірше, ніж я того заслуговую, я не засмучуюся (Ні)

36. Якщо хтось виводить мене з себе, я не звертаю уваги (Ні)

37. Хоча я й не показую цього, мене іноді точить заздрість (Так)

38. Іноді мені здається, що наді мною сміються (Так)

39. Навіть якщо я сердитий, я не вдаюся до "сильних" виразів (Ні)

40. Мені хочеться, щоб мої гріхи були прощені (Так)

41. Я рідко даю здачі, навіть якщо хтось вдарить мене (Ні)

42. Коли виходить не по-моєму, я іноді ображаюся (Так)

43. Іноді люди дратують мене самою своєю присутністю (Так)

44. Немає людей, яких би я по-справжньому ненавидів (Ні)

45. Мій принцип: "Ніколи не довіряти "чужинцям" (Так)

46. Якщо хтось дратує мене, я готовий сказати, що я про нього думаю (Так)

47. Я роблю багато такого, про що згодом шкодую (Так)

48. Якщо я розізлюся, я можу вдарити когось (Так)

49. З дитинства я ніколи не виявляв спалахів гніву (Ні)

50. Я часто відчуваю себе як порохова бочка, готова вибухнути (Так)

51. Якби всі знали, що я відчуваю, мене б вважали людиною, з якою нелегко працювати (Так)

52. Я завжди думаю про те, які приховані причини змушують людей робити що-небудь приємне для мене (Так)

53. Коли на мене кричать, я починаю кричати у відповідь (Так)

54. Невдачі засмучують мене (Так)

55. Я бійку не рідше і не частіше, ніж інші (Так)

56. Я можу згадати випадки, коли я був настільки злий, що хапав річ, яка потрапляла під руку, і ламав її (Так)

57. Іноді я відчуваю, що готовий першим почати бійку (Так)

58. Іноді я відчуваю, що життя ставиться до мене несправедливо (Так)

59. Раніше я думав, що більшість людей каже правду, але тепер я в це не вірю (Так)

60. Я лаюся тільки зі злості (Так)

61. Коли я вчиняю неправильно, мене мучить совість (Так)

62. Якщо для захисту своїх прав мені потрібно застосувати фізичну силу, я застосовую її (Так)

63. Іноді я виражаю свій гнів тим, що стукаю кулаком по столу (Так)

64. Я буваю грубуватим по відношенню до людей, які мені не подобаються (Так)

65. У мене немає ворогів, які б хотіли мені нашкодити (Ні66. Я не умею поставить человека на место, даже если он того заслуживает (Нет)

Ось переклад решти пунктів на українську:

67. Я часто думаю, що жив неправильно (Так)

68. Я знаю людей, які здатні довести мене до бійки (Так)

69. Я не засмучуюсь через дрібниці (Ні)

70. Мені рідко спадає на думку, що люди намагаються розізлити або образити мене (Ні)

71. Я часто тільки погрожую людям, хоча і не збираюся приводити погрози до виконання (Так)

72. Останнім часом я став зануда (Так)

73. В суперечці я часто підвищую голос (Так)

74. Я намагаюся зазвичай приховувати своє погане ставлення до людей (Так)

75. Я краще погоджуся з чим-небудь, ніж стану сперечатися (Ні)

Розподіл питань за 8 шкалами опитувальника Басса-Дарки:

1. Фізична агресія:

1, 9, 17, 25, 33, 41, 48, 55, 62, 68

2. Непряма агресія:

2, 10, 18, 26, 34, 42, 49, 56, 63

3. Роздратування:

3, 11, 19, 27, 35, 43, 50, 57, 64, 72

4. Негативізм:

4, 12, 20, 28, 36

5. Образа:

5, 13, 21, 29, 37, 44, 51, 58

6. Підозрілість:

6, 14, 22, 30, 38, 45, 52, 59, 65, 70

7. Вербальна агресія:

7, 15, 23, 31, 39, 46, 53, 60, 66, 71, 73, 74, 75

8. Почуття провини:

8, 16, 24, 32, 40, 47, 54, 61, 67, 69

Шкалу прокрастинації Такмана (Тест 4).

Пряме нарахування балів:

1. Я відкладаю на завтра те, що потрібно зробити сьогодні.

2. Я переглядаю рекламні матеріали або читаю щось цікаве, коли маю працювати.

3. Я потурання своїй лінощам.

4. Я займаюся справами, що приносять мені задоволення, коли потрібно сісти за рутинну роботу.

5. Я краще відпочину ще кілька хвилин, ніж відразу приступлю до виконання завдання.

6. Я відкладаю неприємні для мене справи.

7. Я відкладаю на завтра те, що повинен зробити сьогодні.

8. Я завжди кажу собі, що виконаю роботу "за хвилину".

9. Я пам'ятаю, що мав зробити щось важливе, коли вже стає запізно.

Зворотне нарахування балів:

10. Коли у мене з'являється вільний час від виконання домашніх справ або інших обов'язків, я намагаюся використати його для роботи над важливими завданнями.

11. Я намагаюся виконати неприємні завдання якомога швидше.

12. Незалежно від того, подобається мені завдання чи ні, я намагаюся виконати його якнайкраще.

13. Я віддаю перевагу завершити робочий проект за один присід, ніж повертатися до нього кілька разів.

14. Я дотримуюся девізу "Не відкладай на завтра те, що можна зробити сьогодні".

15. Я намагаюся не надто відволікатися на сторонні справи, коли займаюся важливим проектом.

16. Я взагалі не відкладаю справи на потім.

Таким чином, для тверджень 1-9 бали нараховуються прямо, а для тверджень 10-16 бали слід перераховувати за формулою "новий бал = 6 - старий бал".