

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Інформаційно-аналітична система визначення психічного стану  
людини»**

Виконав: студент 2 курсу, групи 2КІ-20м  
напряму підготовки (спеціальності)  
123 — «Комп'ютерна інженерія»  
\_\_\_\_\_ Григоришен О.О.

Керівник: к.т.н., доц. каф ОТ

\_\_\_\_\_ Колесник І.С.

«\_\_\_» \_\_\_\_\_ 2021 р.

Опонент: к.т.н., доц., зав. каф МБІС

\_\_\_\_\_ Карпінець В.В.

«\_\_\_» \_\_\_\_\_ 2021 р.

Допущено до захисту  
Завідувач кафедри ОТ  
д.т.н., проф. Азаров О.Д.  
«\_\_\_» \_\_\_\_\_ 2021 р.

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Освітньо-кваліфікаційний рівень: магістр  
Спеціальність 123 — «Комп'ютерна інженерія»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

обчислювальної техніки

\_\_\_\_\_ проф., д.т.н. О.Д. Азаров

«\_\_» \_\_\_\_\_ 2021 р.

**З А В Д А Н Н Я**  
**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**  
Григоришену Олександрю Олександровичу

1 Тема роботи «Інформаційно-аналітична система визначення психічного стану людини», керівник роботи Колесник Ірина Сергіївна, к.т.н., доцент кафедри ОТ затверджені наказом №227 вищого навчального закладу від 24.09.2021 р.

2 Строк подання студентом роботи 15.12.2021 р.

3 Вихідні дані до роботи — проаналізувати методи і засоби для розробки інформаційно-аналітичної системи визначення психічного стану людини. Розробити алгоритм, що дозволяє визначати психічне захворювання за наявними симптомами. Інформаційно-аналітичну систему створити у вигляді програмного продукту.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) — розглянути та проаналізувати методи визначення психічного стану людини. Провести порівняння з аналогами Розробити схеми алгоритму та структури інформаційно-аналітичної системи. Реалізувати систему у вигляді програмного продукту.

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

Схема структури інформаційно-аналітичної системи визначення психічного стану людини, блок-схема алгоритму, лістинг коду бази даних та основного коду програми.

6 Консультанти розділів роботи представлені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Колесник Ірина Сергіївна., к.т.н., доцент каф. ОТ		
4	Лесько О. Й., к.е.н., професор		

7 Дата видачі завдання 07.09.2021 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2 — Календарний план

№	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	01.09.21	
2	Огляд та аналіз джерел інформації	02.09-09.09.21	
3	Розробка технічного завдання	10.09-18.09.21	
4	Огляд засобів для визначення психічного стану людини	19.09-01.10.21	
5	Аналіз засобів для визначення психічного стану людини	02.10-18.10.21	
6	Розробка структурної схеми алгоритму	19.10-31.10.21	
7	Розробка блок-схеми алгоритму	01.11-10.11.21	
8	Практична реалізація інформаційно-аналітичної системи	11.11-16.11.21	
9	Розрахунок економічної частини роботи	17.11-30.11.21	
10	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12-06.12.21	
11	Аналіз виконання роботи, висновки, додатки	07.12-14.12.21	
12	Перевірка якості виконання магістерської роботи та усунення недоліків	15.12.21	

Студент \_\_\_\_\_ Григоришен О.О.

Керівник роботи \_\_\_\_\_ Колесник І.С.

## АНОТАЦІЯ

УДК 004.42

Григоришен О.О. Інформаційно-аналітична система визначення психічного стану людини. Магістерська кваліфікаційна робота зі спеціальності 123 – комп'ютерна інженерія, освітня програма – комп'ютерна інженерія. Вінниця: ВНТУ, 2021. 135с.

На укр.мові. Бібліогр.: 48 назв; рис.: 21 ; табл. 10.

У магістерській кваліфікаційній роботі розроблено інформаційно-аналітичну систему визначення психічного стану людини, яка дає змогу встановити діагноз індивіда за сукупністю певних симптомів, мінімізувавши час на обробку інформації.

Окрім того, систему можна використовувати для профілактики психічних захворювань та виявлення хвороби на ранній стадії, що значно підвищує ефективність успішного лікування.

У роботі розроблено алгоритм, який дає можливість виявити психічне захворювання за наявними симптомами та обрано інкрементну модель життєвого циклу, яка є зручною у використанні. Також для моделювання ПЗ були використанні контекстні діаграми.

Інформаційно-аналітична система визначення психічного стану розроблена мовою Java та використанням середовища розробки Android Studio.

В магістерській роботі були проведені економічні розрахунки для визначення доцільності та конкурентоспроможності створеного програмного продукту.

Ключові слова: інформаційно-аналітична система, психологічний стан, психоаналіз, психодіагностування.

## ANNOTATION

Grigorishen O.O. Information-analytical system for determining the mental state of man. Master's thesis in specialty 123 – computer engineering, educational program – computer engineering. Vinnytsia: VNTU, 2021 – 135 p.

In Ukrainian language. Bibliografer: 48 titles; fig .: 21; tabl. 10.

The master's qualification work has developed an information-analytical system for determining a person's mental state, which allows to diagnose an individual by a set of certain symptoms, minimizing the time for information processing.

In addition, the system can be used to prevent mental illness and detect disease at an early stage, which significantly increases the effectiveness of successful treatment.

The algorithm is developed in the work, which makes it possible to detect mental illness by the available symptoms, and an incremental life cycle model is selected, which is easy to use. Contextual diagrams were also used to model the software.

The information-analytical system for determining mental state was developed in Java with using Android Studio environment.

In the master's thesis, economic calculations were performed to determine the feasibility and competitiveness of the created software product.

Keywords: information-analytical system, psychological state, psychoanalysis, psychodiagnosis.

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....</b>	<b>8</b>
<b>ВСТУП.....</b>	<b>9</b>
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ СТВОРЕННЯ СИСТЕМИ... 12</b>	<b>12</b>
1.1 Загальні положення .....	12
1.2 Поняття емоції.....	14
1.3 Моделі оцінки психічного стану .....	15
1.4 Методи визначення психічного стану людини.....	16
1.5 Психологічне тестування .....	18
1.6 Комп'ютерне тестування .....	20
1.7 Загальний опис застосунків спрямованих на вирішення проблеми .....	24
1.7.1 Методи традиційної психіатрії.....	24
1.7.2 Тест Люшера.....	27
1.7.3 Тест Роршарха .....	28
<b>2 РОЗРОБКА МЕТОДУ І АЛГОРИТМУ ПРОГРАМНОГО ДОДАТКУ. 30</b>	<b>30</b>
2.1 Інкрементна модель алгоритму .....	30
2.2 Моделювання системи та побудова деревоподібної структури .....	32
2.3 Класифікація ключового функціоналу додатку .....	36
2.4 Принципи розробки та структура Android-додатків .....	41
<b>3 СТРУКТУРА ТА ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ .....</b>	<b>47</b>
3.1 Вибір інструментарію для розробки продукту .....	47
3.2 Початкова конфігурація додатку .....	50
3.3 Розробка структури програми .....	54
3.4 Розробка алгоритму роботи мобільного додатку .....	56
3.5 Підтримка баз даних інформаційно-аналітичної системи.....	58
3.6 Проектування додатку.....	61
3.7 Налаштування розмітки користувацького інтерфейсу .....	64
3.8 Опис розробленого програмного продукту .....	66

						<i>08-23.МКР.020.00.000 ПЗ</i>					
						Інформаційно-аналітична система визначення психічного стану людини. Пояснювальна записка.	<i>Лім.</i>	<i>Маса</i>	<i>Масштаб</i>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>							
<i>Розроб.</i>		<i>Григорішен О.О.</i>									
<i>Керівник</i>		<i>Колесник І.С.</i>									
<i>Реценз.</i>		<i>Карпінець В.В..</i>									
<i>Н. Контр.</i>		<i>Швець С.І.</i>			<i>ВНТУ, гр. 2КІ-20м</i>						
<i>Затверд.</i>		<i>Азаров О.Д.</i>									

<b>4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ВИЗНАЧЕННЯ ПСИХІЧНОГО СТАНУ ЛЮДИНИ .....</b>	<b>72</b>
4.1 Проведення технологічного аудиту науково-технічної розробки.....	72
4.2 Розрахунок витрат на здійснення науково-дослідної роботи .....	79
4.2.1 Витрати на оплату праці .....	79
4.2.2 Соціальні відрахування .....	80
4.2.3 Сировина та матеріали .....	81
4.2.4 Амортизація обладнання, програмних засобів та приміщень .....	81
4.2.5 Паливо та енергія для науково-виробничих цілей.....	82
4.2.6 Накладні (загальновиробничі) витрати та інші витрати .....	83
4.3 Розрахунок економічної ефективності науково-технічної розробки за її впровадження безпосередньо розробником (замовником) .....	84
<b>ВИСНОВКИ .....</b>	<b>89</b>
<b>ПЕРЕЛІК ДжЕРЕЛ ПОСИЛАННЯ.....</b>	<b>91</b>
<b>ДОДАТОК А Технічне завдання .....</b>	<b>96</b>
<b>ДОДАТОК Б Принципова схема інформаційно-аналітичної систем.....</b>	<b>100</b>
<b>ДОДАТОК В Деревоподібна структура процесів.....</b>	<b>101</b>
<b>ДОДАТОК Г Схема ключових компонентів ієрархії додатку .....</b>	<b>102</b>
<b>ДОДАТОК Д Алгоритм інформаційно-аналітичної системи.....</b>	<b>103</b>
<b>ДОДАТОК Е Лістинг коду Database.java .....</b>	<b>104</b>
<b>ДОДАТОК Ж Лістинг файлів програмного коду.....</b>	<b>121</b>
<b>ДОДАТОК И Лістинг коду розміток .....</b>	<b>130</b>
<b>ДОДАТОК К Протокол перевірки кваліфікаційної роботи .....</b>	<b>135</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД — база даних,

ЖЦ — життєвий цикл,

ЗМІ — засоби масової інформації,

ООП — об'єктно орієнтовна система,

ОС — операційна система,

ПЗ — програмне забезпечення,

ПК — персональний компютер,

ADT — Active Denial Technology,

API — Application Programming Interface,

CDK — Cloud Development Kit,

CLDC — Connected Limited Device,

DFD — Data Flow Diagrams,

MIDP — Mobile Information Device Profile,

SDK — Software Development Kit,

XML — Extensible Markup Language.



## ВСТУП

Мозок людини можна назвати чудовим біологічним комп'ютером, одним із найцінніших досягнень еволюції. Це може зробити нас успішними, підняти нас з бідності до багатства, дати змогу заробляти на життя, долати смертельні віруси, спілкуватися з іншими, ділитися почуттями та співпереживати.

Але, як і в будь-якому комп'ютері, у роботі мозку також виникають збої. Часто вони незначні і не впливають на наше повсякденне життя, проявляючись лише у вигляді головного болю чи безсоння. Але іноді трапляються збої, які призводять до радикальної зміни поведінки людини (відхилення від норми), часто роблячи її соціально неприйнятною. У таких випадках потрібно визначити в чому полягає відхилення і скласти план повернення до норми, тобто лікування.

Психічна хвороба — це розлад мозку, який спричиняє зміну психічного чи фізичного здоров'я людини, завдаючи дискомфорту пацієнту чи навколишньому середовищу. Часто такі незручності можуть заподіяти страждання та зашкодити працездатності людини. Однак своєчасна діагностика та правильне лікування можуть допомогти одужанню пацієнта і, у свою чергу, його повернення до суспільства та нормального життя.

Психічні розлади зазвичай можна класифікувати на психотичні розлади (психози), які включають марення, галюцинації та основні розлади емоцій і настрою, неврози, невротичні розлади, деякі психосоматичні розлади, алкоголізм, розумову відсталість, звички розладу особистості, деякі статеві спотворення.

У магістерській кваліфікаційній роботі розроблено та реалізовано інформаційно-аналітичну систему визначення психічного стану людини.

**Актуальність дослідження** пов'язана з необхідністю виявлення ситуацій, коли наслідки психічного захворювання і хвороби створюють обмеження життєдіяльності. Унікальна програма реабілітації використовує різноманітні психокорекційні методи для подолання соціально-психологічних

проблем пацієнтів, одним з яких є психологічне тестування. За допомогою комп'ютерної діагностики ми можемо спочатку швидко отримати необхідну інформацію про індивіда, відсіявши її з поміж іншої менш актуальної інформації.

**Метою роботи** є розробка інформаційно-аналітичної системи визначення психічного стану людини, яка може допомогти діагностувати комплекс наявних симптомів, заощаджуючи час на обробку наявної інформації. Крім того, таку систему можна використовувати для запобігання початку психічних захворювань на ранніх стадіях, що значно підвищує шанси на успішне лікування.

**Завдання дослідження** — скласти та проаналізувати тему роботи. Розкрити сутність основних понять. Створити алгоритм, що дозволяє діагностувати психічні захворювання за наявними симптомами.

Система повинна забезпечувати наступні функції:

- тестування для діагностики психічних захворювань;
- аналіз, огляд та обробку поточних результатів;
- функціонал для генерування висновку;
- виведення статистичних даних для розробника;
- підтримання сучасними ОС;
- забезпечення зручного для користувача графічного інтерфейсу.

**Методи дослідження** магістерської роботи: методи системного аналізу, які застосовувалися для дослідження психічного стану особи, методи причинно-наслідкового аналізу і прямого структурного аналізу.

**Об'єктом дослідження** є процес аналізу людської психіки, поведінки та емоційного стану.

**Предметом дослідження** є методи, моделі та інструменти створення інформаційно-аналітичних систем визначення психічного відхилення.

Методи розробки:

- мова програмування Java;
- використання SQL;
- середовище розробки Android Studio.

**Наукова новизна** полягає в комплексному використанні методів психологічного дослідження для отримання достовірної оцінки психіатричного стану людини.

**Практична цінність** роботи полягає в розробці алгоритму та програмного продукту, який відрізняється збільшеною швидкістю опрацювання інформації, легкою доступністю та дешевизною для лікування та профілактики психічних захворювань. Сферою застосування одержаних результатів роботи є комп'ютерна система для визначення психічного та емоційного діагнозу стану людини в умовах реального часу.

У першому розділі проведено техніко-економічне обґрунтування системи та порівняння її з аналогами. Також розглянуто основні теоретичні відомості про психологію в цілому та її аспекти, проаналізовано джерела вирішення проблем. Другий розділ описує вивчення інформаційної системи з метою визначення психічного стану людини з точки зору системного аналізу. У цьому розділі показано дерево цілей і показано, як працює система. В третьому розділі вибирається програмне забезпечення для вирішення проблеми, описується структура та архітектура програмного продукту. У четвертому розділі описано практичне виконання відповідно до встановлених критеріїв. П'ятий розділ демонструє проведені економічні розрахунки для визначення доцільності та конкурентоспроможності створеного програмного продукту.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ СТВОРЕННЯ СИСТЕМИ

## 1.1 Загальні положення

Психіатрія — курс медицини, який вивчає види психічних захворювань, методи лікування та організацію психіатричної допомоги. Вона є однією із складових клінічної медицини, у якій не є обов'язковим використання таких основних методів дослідження, як огляд, пальпація, аускультация, але для діагностики та оцінки психіки пацієнта використовують два методи: спостереження та опитування. У разі психічного розладу ви зможете, в результаті спостереження за пацієнтом, виявити унікальність його дій, поведінки та вчинків. Зокрема, якщо пацієнта турбують нюхові або слухові галюцинації, то він може закривати ніс або вуха [1].

Обстеженням можна виявити, що пацієнти герметизують вікна, щоб токсини, які викидають промислові підприємства, не потрапили в квартиру. Такі дії свідчать про те, що у особи можуть бути галюцинації. При умові наявності нав'язливих страхів пацієнти можуть виконувати незрозумілі жести або ритуальні рухи. Наприклад, постійне миття рук від бруду чи відмахування від нав'язливих комах. При розмові з психіатром хворий може розповісти про свої переживання, поганий настрій чи страхи, пояснити свою неправильну поведінку, а також описати неадекватні судження та помилкові уявлення [2].

Щоб точно оцінити становище людини, велику увагу слід приділяти збору інформації про її минуле життя та відношення до подій та стосунків з оточуючими людьми. Збір необхідної інформації характеризує хворобливість тлумачень певних явищ та подій. У цьому випадку можна говорити про анамнез хворого та його психічний стан.

Доволі часто психіатри зустрічаються з анозогнозією у пацієнта, що являє собою відмову від хвороби родичами, близькими та самим хворим. Це притаманно таким захворюванням, як шизофренія, епілепсія та олігофренія. У медицині трапляються випадки, коли батьки хворого, професійні лікарі, не бачать явних симптомів захворювання. Хоча хвороба іноді зникає, деякі з них

погоджуються пройти необхідну діагностику та лікування. У цьому випадку психіатр повинен продемонструвати найвищий рівень професіоналізму, гнучкості та ввічливості. Важливо лікувати хворобу, не уточнюючи, не вимагаючи та не звертаючи уваги на інтереси пацієнта. Іноді близькі родичі заперечують захворювання і відмовляються від необхідного лікування. Така поведінка може призвести до посилення симптомів та спричиняє перехід в хронічне захворювання [3].

На реакцію людини і захворювання впливає багато факторів:

- стан захворювання, швидкість розвитку та тяжкість хвороби;
- власна думка пацієнта про його захворювання;
- психотерапевтична ситуація в лікарні та методи лікування;
- психічний та емоційний стан людини;
- відношення хворого та родичів до захворювання.

Для персоналу важливо виявити зміни чутливості пацієнта та надати їм відповідний догляд. Коли пацієнт страждає від погіршення слуху, слід говорити голосніше, а у випадку світло боязні — затемнити кімнату. Потрібно пам'ятати, що безпідставні скарги хворого на смак їжі можливі, як наслідок не сприйняття смаку, а нарікання на неприємні запахи — загостренням нюху.

Закони чутливості включають поняття адаптації, виживання та чутливості. Адаптація викликає зміну чутливості внаслідок тривалих подразників. Наприклад, якщо людина заходить в темну кімнату з надвору в сонячний день. Спочатку вона нічого не бачить, але починає бачити все після адаптації зорового аналізатора.

Емоції відіграють важливу роль у житті людини. Емоції — найпростіший психічний процес. Коли ми говоримо про колірне сприйняття, ми говоримо про суцільний колір, віддаляючись від розмірів і форми об'єкта. Емоції становлять основу розумової діяльності і дають чуттєвий матеріал для побудови розумових образів. На них базуються основні пізнавальні процеси людини: відчуття, образи, мислення, уява. Емоції — це ніби «ворота» до нашого знання.

У хворих з психічними розладами межі чутливості знижуються, а схильність до чутливості підвищується. Стукіт взуття, запах парфумів, які відчуває пацієнт згубно впливають на психічний стан хворого. Почуття складають основу всіх психічних дій, особливо тих, які тісно пов'язані з емоціями. Вони можуть бути пов'язані як з нижчим рівнем, так і з несвідомими емоціями. Першою появою захворювання є часта зміна емоційних станів [4].

## 1.2 Поняття емоції

Існує досить велика кількість людських емоцій. У житті людини існують різноманітні емоційні переживання (радість, страх, здивування, пристрасть, сум, огида тощо), що відрізняються за характером і особливістю впливу на її роботу. Без емоцій будь-яка діяльність людини не можлива. Філософ Спіноза говорив, що емоції «зменшують чи збільшують спроможність самого тіла до дії».

Основні означення емоцій:

- психічний стан, який підкріплений деякими реакціями: руховими і секреторними;
- поведінка організму переважно за рахунок вісцерального компонента (тобто діяльності внутрішніх органів) [5].

Додаткові означення:

- психічний стан, що характеризується певними емоціями та руховою симптоматикою, зумовленими суб'єктивними причинами особистості;
- ментальний статус, який визначає його поведінку;
- стан афекту, що виникає внаслідок порушення чи затримки інстинктів людини;
- випадкова реакція організму.

Означення емоцій за концепцією Д.Креше та Р.С. Крутчфельд:

- почуття типу емоційного переживання;
- емоційні риси;
- зміни в фізіології організму.

Таким чином, емоції — це чудові психологічні явища, що виникають при оцінці ситуації або зовнішніх впливів. Склад суб'єктивного почуття полягає в усвідомленні важливості ситуації і поведінки. Окрім реагування на важливі ситуації та розуміння важливості, це також передбачає активізацію особистості до адаптації певних ситуацій [5].

### 1.3 Моделі оцінки психічного стану

На даний момент у вчених немає спільної думки щодо визначення означення емоцій. Зазвичай визначаються належним чином, та часто зображується переліком — тривога, ненависть, пригніченість, страх, збудження радість, смуток і здивування. Поняття емоція означає «тривале переживання, тривога, яка викликана переживанням радості, страху, огиди, смутку та ін». (Словник Лароссе, 1990), чи «психічні відчуття (наприклад, біль, бажання, надія тощо) на відміну від асиміляції чи волі» (Oxford English Dictionary, 1987). Хоча поняття емоції є складним, більшість дослідників підтверджують, що емоційно обумовлені гострі стани тривають недовго і пов'язані з певною подією чи об'єктом. Kreibig (2010) підсумовує емоцію, як багатокomпонентну реакцію, що призводить до зміни суб'єктивних емоцій та явища, що призводять до поведінкової та фізіологічної активації [6]. Зазвичай поняття емоцій визначають як різні категорії (базова модель) або безперервні виміри (безперервна модель). Добре відомою дискретною основою емоцій, за Екманом (1993), є шість основних або «базових» емоцій, а саме — смуток, опір, радості, страх, гнів і здивування. Зазначена модель впорядковує другорядні емоції навколо основних емоцій, які вважаються відправною точкою ландшафту емоційного стану, які обговорюють зважаючи на їх значення адаптації, як засобу вибору з точки зору часу та форми. Ви можете отримати комбінації або суміші, які утворюють величезні емоції (змінні емоції) з таблиці вихідних ситуацій [7].

Відмінно від базової моделі емоцій, вимірні (безперервні) моделі розглядають області або місця в багатовимірному просторі, які в основному

визначаються основними вимірами в обмеженій кількості відчуттів. Вони пов'язанні завдяки взаємному розміщенню їх в ефективному просторі.

Емоційні зміни відбуваються різними способами — у вигляді депресії, емоційної слабкості, байдужості, тощо. Депресія — психічне захворювання, що характеризується певним негативним емоційним фоном. Вона характеризується змінами настрою, розумовою відсталістю та руховою відсталістю. Пацієнти з депресією частіше відчують неприємні відчуття і відчують лише негативні емоції у всіх ситуаціях. Маніакальний синдром — захворювання, що характеризується афективним станом та постійними перепадами настрою, гіперактивністю і прискореною розумовою діяльністю.

Дисфорія — це психічний стан, що характеризується гнівом, злістю, пригніченим настроєм та надзвичайною чутливістю до дій інших, які схильні до агресії. Емоційна нерівність — це суперечливість ряду емоційних установок до певного предмета, які відчуває людина одночасно. Апатія — психічний розлад, який характеризується відсутню інтересу до оточення, байдужістю до явищ навколишнього середовища та відсутністю мотивації. Він інколи може бути викликаний деякими ураженнями мозку, деменцією та соматичними захворюваннями. Така зміна емоцій характерна хворим шизофренією. Відзначаються й інші зміни емоційних станів, як-от — гнів, інфантильність (якщо хворий наслідує поведінку дитини), невміння контролювати емоції, швидке перемикання зі сміху на плач, постійний гнів без причини [8]. Депресія потребує особливої уваги, якщо емоційно тривожна. Хворі на депресією часто залишаються на самоті, з пригніченою руховою та розумовою активністю. У них часто виникають суїцидальні міркування, пориви та намагання, відповідно вони потребують особливого догляду. Пацієнти, які знаходяться у маніакальному стані занадто мобільні, галасливі, порушують дисципліну у медичних закладках, створюючи проблеми для оточуючого середовища та незручності для медичних працівників.

#### 1.4 Методи визначення психічного стану людини

Для визначення психічного стану, застосовують методи оцінки:



- експеримент;
- дослідження;
- психологічне тестування;
- психофізіологічний метод;
- суб'єктивний метод;
- комп'ютерне тестування.

В даній магістерській кваліфікаційній роботі увагу буде приділено методам психологічного та комп'ютерного тестування. Суб'єктивний метод дозволяє провести аналіз та спостереження того, як індивід може дати характеристику свого психічного стану. Для дослідження психічного стану пацієнта, цим методом, застосовують ведення записів у щоденнику, у яких ведуть запис усіх думок та емоцій за увесь день [9].

Оскільки стан психіки змінюється протягом невеликого періоду та залежить від ситуації, то в тестах-опитувальниках потрібно давати відповідь на питання стосовно психічного стану на даний період. Виділяють однорівневі та багаторівневі тести. Однорівневий тест дає можливість визначити рівень вираженості властивості. Багаторівневий у свою чергу направлений на визначення двох і більше властивостей. Найвідомішими тестами визначення психічних станів є — «Шкала тривожності Цунга», «Методика Тейлора», «Шкала депресії Бека» та інші.

Кожен з наведених методів має деякі недоліки, зокрема нездатність аналізувати свої дії, поведінку, рефлекси, стан та почуття. На достовірність результатів тесту також будуть впливати особливості стану людини, її індивідуальні риси. Наприклад, песимісти характеризують свій стан, менш позитивно, ніж його характеризують оптимісти [10]. Особливостями психофізіологічних методів є те, що вони поділяються на психологічно-проективні методики та фізіологічні. Фізіологічні методи, застосовуються для оцінки працездатного стану людини шляхом визначення психофізіологічних параметрів, наприклад частота дихання, серцебиття, пульсу, тощо. Також такі методи дозволяють описати зміни у функціонуванні психіки особи, і нерідко

ці об'єктивні характеристики, засновані на фізіологічній інформації вважаються достовірнішими та надійнішими за суб'єктивні [11].

Психологічне та комп'ютерне тестування — це спеціалізовані методи психодіагностичного тестування, за допомогою яких можна отримати точні кількісні чи якісні ознаки досліджуваного явища. Тести різняться від інших методів тим, що забезпечують точний процес збору та обробки вхідних даних, а також унікальність подальшої інтерпретації. Досліджуючи та порівнюючи психологію різних людей за допомогою тестів, можна робити диференційовані та порівняльні оцінки [12].

### 1.5 Психологічне тестування

Психологічне тестування використовується під час профорієнтації, вибору професії, психологічному консультуванні, складанні плану роботи, дослідницькій роботі тощо. Тестування — це метод психологічної діагностики, основним організаційним аспектом якого є використання певною мірою стандартних та тестових запитань [13]. Воно застосовується для стандартизації вимірювання особистісних відмінностей. Існує три головні сфери використання методу тестування: навчання та освіта, психодіагностичне консультування, психотерапія тощо. Метод тестування забезпечує необхідні навички, знання, особистісні характеристики та дає змогу з певною ймовірністю визначити сучасний рівень розвитку [14].

Розрізняють три етапи психологічного тестування:

- вибір тесту (зважаючи на мету тесту та надійність тесту);
- поведінка (визначається тестовими інструкціями);
- пояснення результатів (визначаються рядом теоретичних припущень щодо теми тесту) [15].

В основі тестування лежить стандартизована тестова система, яка дає можливість виміряти конкретні психологічні відмінності в якості та кількості. У Стародавньому Китаї, Греції та В'єтнамі спеціальні тести використовувалися для відбору учнів для навчання та державних службовців.

Витоки наукового використання експериментів сягають імен британського психолога Джеймса МакКінна Кетла (1860-1944) і англійського вченого Френсіса Гальтона (1822-1911). Дж. Кеттель запровадив визначення поведінки людини на основі «середньої» людської ідеї. Його параметри визначаються середнім значенням, отриманим при великій кількості пошуків. У 1884 році Гальтон заснував лабораторію, яка досліджувала 17 показників людини.

Тести з визначення розумових здібностей були проведені у Франції Альфредом Біне (1857-1911) і Теодулом Сімоном (1873-1961) та використовувалися для відбору дітей до школи. Коефіцієнт інтелекту людини — IQ ("IQ") був запропонований у 1912 р. В. Штерном [16]. Використання завдань стандартизації, можливість одночасно вивчати багато людей, нетривалість процедури та легкість обробки результатів допомагали у поширенні тестових методів у вивченні різних психічних особливостей. Проте слабка розробленість методології та способів використання зазнала різкої критики. Критика тестів, та її часткові методи, необґрунтованість процесу оцінювання тестів, не звертають уваги на індивідуальні особливості особистості, можливість компенсації.

Науково необґрунтовані експерименти та їх широке застосування завдали великої шкоди розвитку психології в нашій країні [17]. У 1936 році випробування були заборонені. В Україні почали використовувати тести для вивчення своєї професії лише у другій половині ХХ ст. [18]. Діагностична значимість тесту значно залежить від наукових тестів, правдивість базової психологічної реальності, професійної підготовки вченого. Розроблено та випробувано експериментальні методики, що застосовуються для вивчення різноманітних психічних процесів та рис людини [19].

Тест — стандартизований підхід психологічного вимірювання, створений для визначення вираженості особистості та психічних особливостей чи умов при розв'язанні конкретного завдання. Він нормується в межах між індивідуальними характеристиками. Тест — це серія коротких експериментів

(питань, задач, тощо). Результати тестування є показником психічних характеристик або ситуацій [20].

Розрізняють два види тестів — психологічні та тести успішності (знань, компетенцій, умінь, загальної чи професійної підготовки).

За специфікою застосування тестів їх розділяють [21]:

- за темою діагностики — розумові тести, творчі тести, особистісні тести, тести навичок, психодіагностичні тести;
- за способом виконання — маніпуляційні, тест «ручка», асоціативні, комп'ютерні;
- аналітичні випробування і тест кейси, а також індивідуальні та групові випробування;
- дошкільні, клінічні, професійні.

Тести являють собою спеціальні методики для психодіагностичного вивчення, за допомогою яких можна отримати точні кількісні чи якісні ознаки явища, яке вивчають. Вони відрізняються від інших методів дослідження тим, що забезпечують точний процес збору та обробки початкових відомостей, а також унікальність подальшого розвитку [22]. Досліджуючи та порівнюючи психологію різних людей за допомогою тестів, вони можуть робити персональні та порівняльні оцінки. Різновиди тесту — тестовий опитувальник та тестова задача. Тестовий опитувальник створюється за системою попередньо продуманої, ретельно відібраної і перевіреної з урахуванням психологічних особливостей запитань, які характеризуються валідністю та надійністю. Тестова задача являє собою оцінку психологічних критеріїв та поведінку особи в залежності від того, що вона робить. У цьому виді експерименту досліджуваному висувається ряд конкретних питань, результати яких оцінюються на предмет наявності чи відсутності досліджуваної якості та рівня розвитку [23].

## 1.6 Комп'ютерне тестування

Комп'ютерне тестування — це сучасний тип психодіагностичного дослідження, що використовує комп'ютерні технології. Застосування цього

виду дослідження зумовлена швидкими темпами розвитку інформаційних та хмарних технологій.

Автоматизація відображення матеріалу та статистична обробка набула широкого розвитку з 1930-х років, але тільки через 40 років відбувся справжній поштовх у розвитку комп'ютерного тестування, завдяки появі нового апаратного устаткування. Починаючи з 1980 року комп'ютерні експерименти йдуть повним ходом: перш за все — це комп'ютерна версія відомої безкоштовної техніки, а з 1990-го року — персональні методи, які враховують здатність сучасних технологій і не використовують стандартні шаблони, так як вони вирізняються за складністю, простором і трудомісткістю матеріалу, мають особливі звукові сповіщення та інше [24].

На початку 21 століття більша частина керування тестового контролю передається на ПК [25]. Хоча деякі етапи дослідження, такі як подання матеріалу, впорядкування даних та інтерпретація кінцевих результатів, були автоматизовані, відповідно в наш час дуже часто зустрічаються програми, що зменшують потребу психолога, під час проходження психодіагностичних тестів.

Основні переваги комп'ютерного тестування:

- інтенсивне виконання — швидкий темп і бездоганна обробка інформації, що дає змогу отримати результат відразу;
- стандартизація усіх пунктів проходження тесту в однакових умовах, точний контроль за процесом (можна зафіксувати кожен час відповіді, що є пріоритетним для інтелектуальних тестів);
- можливість виключити фактор присутності лікаря, як психологічного подразника (особливо важливо під час тестування);
- наочність процесу та допитливість (найважливіше для навчальних програм — зосередження уваги кольором, звуковим супроводом, елементами гри) [26];
- зручне архівування висновків;
- можливість комбінувати тести (програмні пакети) з одностороннім поясненням;

- зручність накопичення даних (усі інструменти на одному диску);
- доступ до публічних досліджень (використання мережі Інтернет).

Комп'ютерне тестування має такі недоліки:

- важкість, об'ємність та вартість розробки програмного забезпечення;
- потреба в дорогому комп'ютерному обладнанні;
- не мобільність застосування обладнання в дорожніх умовах [27];
- необхідність підготовки предмета до роботи з програмними засобами;
- складність у роботі з невербальними матеріалами, особливі труднощі при перекладі проектних тестів у комп'ютерну форму;
- відсутність спеціального підходу до тесту (не збереження психодіагностичних даних, які були отримані під час співбесіди та моніторингу);
- затримки в обробці та збереженні даних (контроль цього процесу в повній мірі є залежним від програмного забезпечення).

Деякі суб'єкти можуть відчувати наслідки «психологічної відволікання» або «надмірної впевненості» під час роботи з комп'ютером. Відповідно інформація про точність, надійність та представленість безкоштовних пробних версій не має змоги автоматичної передачі комп'ютерним колегам, що призводить до вимушеної стандартизації тестування [28]. Недоліками під час комп'ютерних іспитів проявляються у тому, що психологи ставляться до них насторожено. Таке тестування рідко застосовують в медико клінічній психології, де значення помилки дуже велике.

Відомий психотерапевт Л. С. Віготський [29] визначив основні стадії психодіагностичних досліджень:

- симптоматичний;
- етіологічний;
- типологічний.

Зараз комп'ютерне психодіагностування перебуває на низько-пріоритетному рівні — на рівні визначення причин і припущень у

симптоматичному діагнозі. Більшість з цих недоліків ліквідуються шляхом прогресу сучасних високотехнологічних програм та вдосконаленням процесу дослідження хворих у вигляді тестів. Поштовхом до розвитку тестування є постійне зростання інтересу до наукового та практичного застосування діагностики, що нараховує більше 1000 комп'ютерних експериментів [30].

Поміж доступних варіантів тестів переважають наступні типи [31]:

- за будовою — аналоги порожніх тестів;
- за кількістю перевірених — індивідуальні та групові тести;
- рівень автоматизації тестів — автоматизація одного або декількох раундів анкетування та автоматизація всіх опитувань;
- за призначенням — дослідження та навчання;
- за заявником — кваліфікаційний та комбінований.

Кваліфікаційні тести використовує психолог, відповідно їх створюють спеціалізовані лабораторії або комп'ютерні психодіагностичні центри [32].

Вони мають кілька відмінних рис:

- наявність бази даних;
- містять пароль для доступу до тесту або архіву для умови конфіденційності;
- детальне пояснення висновків за допомогою професійних термінів і коефіцієнтів зі структурою графіків;
- наявність відомостей про методологічних засновників, відомостей про безпечність, інформаційних матеріалів до теоретичних правил, що становлять основу методики.

Комбіновані тести на комп'ютерні навички орієнтуються на професіоналів у суміжних сферах, таких як викладачі та керівники персоналу. Ці тести найчастіше мають пояснення, що полегшують навчання та роботу без використання спеціального лексикону. Тести такого рівня може використовувати і непрофесіонал, звичайний користувач персональних комп'ютерів, яких цікавить психологія.

В результаті з'являються і непрофесійні комп'ютерні тести, налаштовані на просування ідей психодіагностики або просто розважальних цілей.

Використовуючи кваліфікаційні або комбіновані комп'ютерні тести, слід дотримуватися принципів, що й у шаблонному тестуванні. Важливо не ділитися результатами тесту, насамперед якщо на вашому комп'ютері створено декілька облікових записів, щоб захистити ваші файли. Найголовніше, пам'ятайте, що комп'ютерне тестування має лише один інструмент, утиліту та сферу використання [33].

### 1.7 Загальний опис застосунків спрямованих на вирішення проблеми

В даний час не існує інформаційно-аналітичних систем, що можуть самостійно аналізувати дані та вирішувати проблеми. Для пошуку та діагностики психічних розладів в основному використовують традиційні методи, такі як консультація з психологом або психіатром. Для діагностики психічних розладів використовують методики психоаналізу, графічні методики та тестування.

#### 1.7.1 Методи традиційної психіатрії

Одним з базових завдань традиційної психіатрії являється дослідження, виявлення та лікування розладів психіки, їх клінічних форм та передчасне запобігання загострень. Сфера психіатрії також включає боротьбу з різноманітними навичками, проведення перед тестових іспитів та допомогу в дослідженні.

Психоаналіз — це метод психіатрії, який є доповненням до загального знання в області психоіндустрії і покликаний сформувати психологію як самостійну галузь, що не залежить від інших предметів, наприклад, біології. Винятки вважаються найшвидшим і найточнішим способом діагностики припин психічного захворювання. Зигмунд Фрейд являвся провідним спеціалістом з дослідження розладів психоаналізу, на честь якого напрямок отримав назву «фрейдизм». Це спрямування («фрейдизм») міг бути набутий тодішніми психіатрами з метою суттєвого поповнення наявних знань і створення на основі цих знань теоретичної бази. Відповідно до цього Фрейд,



свою працю, розпочав збором й систематизацією психотерапевтичних досліджень, а пізніше на основі здобутих даних розробив психоаналітичну теорію [34]. Такі поняття, як симптоми та синдроми, відіграють ключову роль у психіатрії. Психопатологічний симптом — специфічний симптом певного розладу чи діагнозу. Наприклад, тривога, страх, усі види галюцинацій, хвороби зв'язані з анорексією, панічні атаки, тощо.

Психопатологічний синдром — переривання людської діяльності, що виникає внаслідок сукупності взаємозв'язаних, а іноді і взаємопов'язаних симптомів. Загалом зазначені симптоми утворюють систему стійких станів, яка характеризує стан та поведінку хворого та його усвідомлення навколишнього середовища. Часто ці синдроми мають складні патогенетичні механізми. Застосовують нозологічний принцип для діагностування психопатологічних станів. Цей метод використовується для призначення дуже безпечних і необхідних методів лікування та для запобігання психічного захворювання. Нозологічний принцип заснований на клінічній картині відхилень і їх можливому розвитку. Знання про етіологічні зміни та зміни патогенезу, а також патологоанатомічні зміни організму людини дозволяють призначити повне і точне лікування виявленого психічного захворювання.

Принцип синдрому також використовується для діагностування та класифікації психічних розладів. Цей принцип включає класифікацію клінічних патологій і психічних розладів та тяжкості ураження, а також розділяє патологічні прояви на окремі захворювання та патологічні стани [35]. Шизофренія є одним із найпоширеніших психічних розладів сучасності. На шизофренію припадає менше 1% від загальної чисельності населення, але пацієнти на шизофренію складають половину усіх у психіатричних клініках через тривалість захворювання. Такі пацієнти за часту пригнічені, без бажання жити, тому близько 10% таких хворих потенційні самогубці. У хворих на шизофренію мінімізується кількість інформації, яка направляється до певних ділянок кори головного мозку. При шизофренії процес галюцинацій може бути подібним. Шизофреники не бачать і не чують, в залежності від того, яка область кори активніша. Звичайно в ряді досліджень професора Кана з Утрехта

показали, що електромагнітне збудження мозку пом'якшує галюцинації у шизофреніків. У чоловіків стан шизофренії частіший аніж у жінок і протікає більш важче. Це спричинено тим, що у чоловіків та жінок є відмінності у роботі головного мозку, а саме кількість нервових клітин у жінок переважає кількість клітин у чоловіків.

Згідно відомостей Всесвітньої організації з охорони здоров'я, біля 400 мільйонів людей страждають депресією. Вона також є одним із найбільш поширених психічних захворювань в Україні. Депресія — це стан, при якому людина довгий проміжок часу (не менше двох тижнів), втрачає інтерес до діяльності, не може виконувати повсякденні завдання. Розвиток її спричинений як наслідок складного взаємозв'язку таких факторів, психологічний, біологічний, емоціональний. Негативні випадки та явища найчастіше спричинюють появу депресії, а саме — смерть близьких, психологічна травма, втрата коханого та інше. Вона послаблює вашу реакцію на психологічну травму, порушує ваше звичне життя і погіршує ваш життєвий стан, що призведе до погіршення депресивного стану.

Олігофренія — це вроджений або ранній (до 3-річного віку) хворобливий психічний стан, який визначається стандартними клінічними ознаками. З олігофренією пов'язана загальна розумова відсталість, що виникає не тільки в розвитку розуму і мислення, а й інших психологічних функціях. Вона відрізняється від усіх інших форм психічних захворювань. Це проявляється початковою стадією відсталості, а не інтелектуальною деградацією [36]. Олігофренія може викликати збої в роботі або раннє пошкодження мозку, що пов'язане зі спадковими особливостями людини, а саме дефекти батьківських гамет, що хворі на олігофренію, чи через радіоактивний вплив, отруєння у здорових осіб. На ранніх стадіях внутрішньо-черевного розвитку дитини, різні несприятливі впливи, наприклад, алкоголізм, наркозалежність матері призводять до виникнення олігофренії. Близько одну третю випадків захворювання на олігофренію неможливо діагностувати.

### 1.7.2 Тест Люшера

Великої уваги заслуговує колірна діагностика розроблена доктором М. Люшером, яка отримала назву «Колірний тест Люшера». Це психологічний тест, що дає можливість визначати психофізіологічний стан людини, його стійкість до стресових ситуацій, соціальний стан і комунікативні навички. За допомогою нього діагностують виникнення психологічного стресу, що призводить виникнення клінічних симптомів. Науковець виявив, що кольоро-сприйняття є об'єктивним і універсальним для людини, але конкретні бажання вибору кольору є суб'єктивними. Це дозволяє визначати суб'єктивний стан за допомогою тестових кольорів.

Серед 36 тисяч точних означень відібрано 23 окремих індикатора для вимірювання вибору тестових кольорів. Окремі з них показників мають стосунку до свідомості людини [35]. Тест нерідко використовується, хоча інколи не має необхідності. В переважній кількості випадків тому що це дає багато інформації для роздумів. Не має необхідності у ситуаціях коли діагностується стан, а не риса особистості.

Експерименти показують, що протягом декількох місяців люди ділять такі списки зовсім по-іншому. Ви повинні бути більш розбірливими з допомогою, яку ви надаєте іншим людям.

Процес проведення тесту заключається в сортуванні кольорів об'єктів за ступенем суб'єктивного задоволення. Цей принцип хоче відійти від тривіальності, зв'язаних з модою, звичаями, спільними смаками, уявленнями і спробувати вибрати кольори, виходячи лише з ваших особистих поглядів. На сьогодні є два типи тестів Люшера: короткий і повний. Під час використання короткої версії тесту затосовує восьми колірний набір (таблиця 1.1).

Таблиця 1.1 — Набір кольорів тесту Люшера

Умовний номер	Колір
0	Сірий

## Закінчення таблиці 1.1

1	Темно-синій
2	Синьо-зелений
3	Червоно-жовтий
4	Фіолетовий
5	Жовто-червоний
6	Коричневий
7	Чорний

Висновки діагностики дозволяють проаналізувати та укласти оцінку та поради експертів щодо того, як уникнути психологічного стресу та фізіологічних симптомів, які він викликає. Також тест Люшера доповнює інформацію додатковими відомостями та результатами різних видів терапій.

Так як виокремлення кольору пов'язане з несвідомими процесами психіки то, він відображає те, ким є людина насправді, а не тим, ким вона представляє чи хоче бути. Сторонники застосування тестування переконані в тому, що він дає змогу швидко і точно проаналізувати особистість, враховуючи інформацію, отриманої при класифікації спектра кольорів [35]. Однак є спосіб «обійти» тест, показавши ідеально збалансований стан. Для цього виберіть кольори в черговому порядку — червоний, жовтий, зелений, червоний, синій, коричневий, сірий, чорний.

### 1.7.3 Тест Роршаха

Тест Роршаха (відомий як чорнильний тест) — це метод розробки психологічного тесту, під час якого людину просять описати, що вона бачить із десяти точок чорнила (окремі чорні або сірі точки, решта кольорові). Тест був запроваджений у 1921 році, коли він широко використовувався швейцарським психіатром Г. Роршахом для оцінки освіти та особистості та визначення деяких психологічних станів, який набув популярності 60-х роках.

Історія тесту Роршаха дуже цікава. У 1917 році в невеликому місті Швейцарії Г. Роршах розпочав писати карти для вивчення функціонування розуму. Після опитування людей стосовно побаченого, психіатр виявив, що відповіді пацієнтів з шизофренією переплітаються, і стверджували, що психічний стан можна оцінити при допомозі чисісь обробки побаченої інформації.

На деяких картах пацієнти спостерігали рух людських фігур в чорнильних плямах. Психіатр вияснив, що у здорових індивідів уява особлива і відрізняється від уяви людей з психологічними вадами. Проводячи аналогію у змісті фантастичних асоціацій з відомими, Роршах встановив, що ці асоціації еквівалентні складу мрій. Отже, виявлено чорнильні плями, які виявляють глибокі бажання або проблеми, що базуються на основі людських конфліктів, які не вирішені [36].

Зі змісту асоціацій, створеного чорнильними плямами статистики, можна отримати ту інформацію, яка потрібна людині, яка її радує чи змушує сумує, ту, яка її хвилює, і ту, що має натиснути і перевести у стан фантазій. Своєрідні 10 картинок Роршаха були опубліковані всього за рік до смерті вченого. Після їх появи у Чикаго, картинки миттєво поширилися у США, у якості найпопулярнішого тесту визначення особистості. Друга половина століття характеризується напрямками та тенденціями методу Фрейда, який втратив актуальність і експеримент став напрямом псевдонауки, що спричинило накладання мораторію на його застосування [37].

Точність, повнота та ефективність тесту Роршаха ще не визначені. Однак це також допомагає психологам і психіатрам отримати важливу інформацію для діагностування особистості, а також психічних розладів, що можуть досліджуватися клінічно.

## 2 РОЗРОБКА МЕТОДУ І АЛГОРИТМУ ПРОГРАМНОГО ДОДАТКУ

### 2.1 Інкрементна модель алгоритму

Для інформаційно-аналітичної системи визначення психічного стану людини обрано інкрементну модель ЖЦ (життєвого циклу) програмного додатку (рисунок 2.1), оскільки вона може бути легко реалізовано.



Рисунок 2.1 — Інкрементна модель ЖЦ

Ця система поділяється на частини (інкременти). Кожна частина має окремий розділ про те, як працювати. Інкрементація містить 4 фази:

- початкова фаза;
- фаза розроблення планової документації;
- фаза проектування та побудови;
- фаза отримання кінцевого продукту.

Початкова фаза проекту встановлює його масштаб, доцільність та специфікацію проекту та складність алгоритму продукту. Фаза розроблення планової документації включає опис архітектури, термінів та цінової політики виконання, оформлення тестової документації та розробки тестових кейсів.

Фаза проектування та побудови включає в себе програмну реалізацію продукту та створюється шляхом аналізу, планування, розробки, впровадження та тестування функціональних особливостей програми.

Фаза отримання кінцевого продукту являє собою інтеграцію продукту у середовище. При виборі моделі ЖЦ обумовлювалося тим, що початкова версія ПЗ повинна відповідати певним вимогам, а наступна версія додає додаткові вимоги, і таким чином дозволяє розробляти всі потреби та програмне забезпечення, що сприяє високій ефективності розробки. Вибрана модель, що розвивається, ідеально підходить для «початківців», так як знижує ризики, зручно та надійно реалізує деякі функції шляхом побудови тимчасової версії програмного продукту, що важливо для адміністратора програми [38].

Переваги інкрементної моделі:

- засоби, які потрібні для розроблення додатку, не потребують лишніх фінансових витрат і мінімізують ризики для виробництва та впровадження;
- результатом кожної окремої ітерації є незалежний проміжний продукт, що дає можливість продемонструвати певну частину функціоналу;
- неможливість виникення поразок та збою продукту;
- сегментація ризиків різних частин проекту;
- удосконалення вхідних і вихідних даних, вимог продукту, які можливо реалізувати кожним інкрементом при практичному застосуванні;
- можливість розпізнавання важливих та цінних функціоналів продукту, замовником на початкових фазах розробки.

Основним призначенням інформаційно-аналітичної системи визначення психічного стану людини є швидка і точна методика його визначення на основі обробки даних процесором і аналізу певної інформації, що дає бажані результати. Для визначення психічного стану користувача ЖЦ інформаційної системи містить три базових фази і описується як гілчаста структура.

Принципова схема інформаційно-аналітичної системи визначення психічного стану людини зображена в Додатку Б. Першим кроком у цій системі є аналіз психічного стану користувача. Він передбачає збір інформації про психічний та емоційний стан людини, аналіз та обробку усіх даних.

Другим кроком принципової схеми є побудова аналітичної системи, являє наповнення інформаційних баз даними про розлади психіки, комбінації

симптомів, проектування базової структури системи, проведення діагностування та тестування. Третім кроком визначення діагнозу стану хворого є аналіз стану психіки із занесенням інформації у бланки, аналіз висновків, пропозицій і виведення кінцевих даних.

## 2.2 Моделювання системи та побудова деревоподібної структури

Під час процесу моделювання інформаційно-аналітичної системи використовувалися контекстні діаграми, а саме DFD (Data Flow Diagrams). Діаграми потоків даних являються ключовим інструмент для вимог моделювання системи діагностування. При їх допомозі зазначені вимоги розбиваються на функціональні складові та представлені в мережевому вигляді в поєднанні з потоками даних. Головна ціль — демонстрація процесів, які перетворюють вхідні дані у вихідні, і визначають зв'язок між даними операціями [39]. Використовуючи цю модель, система зображена як багаторівнева структура потоків, які характеризують асинхронний обмін інформації з моменту часу входу у систему до виходу з неї. З кожним послідуочим кроком операції покращуються, до того моменту коли слідкуючий крок не стане примітивним.

Головні складові діаграм DFD (таблиця 2.1):

- значення (сутності);
- обробка даних;
- контейнери вхідних і вихідних даних;
- потоки інформаційних даних.

Сутності — це об'єкти, які є ключем системних вхідних і вихідних значень. Показано у вигляді затіненого прямокутника, що вказує на те, що об'єкт розташований поза площиною графіка. Зовнішні значення на схемі поєднуються з сутностями, а також їх можна з'єднувати між собою.

Таблиця 2.1 — Складові DFD діаграми

Поняття	Поняття	Поняття	Поняття
---------	---------	---------	---------



Закінчення таблиці 2.1

Значення (сутності)	Зовнішній по відношенню до системи об'єкт, що обмінюється з нею потоками даних	Ім'я зовнішнього об'єкта	Номер Ім'я
Обробка даних	Дія, що виконується модельованою системою	Ім'я функції	Номер
			Ім'я
			Механізм
Потік інформаційних даних	Об'єкт, над яким виконується дія. Може бути інформаційним (логічним) або таким, що управляє. (Поток и, що управляють, позначаються пунктирною лінією із стрілкою)	Ім'я об'єкта →	Ім'я об'єкта →
Контейнери вхідних і вихідних даних	Структура для зберігання інформаційних об'єктів	Ім'я об'єкта	Ім'я об'єкта

Обробка даних полягає в перетворенні вхідних потоків даних у вихідні дані відповідно до певного алгоритму. Кожен обробка даних у системі має свій номер, який відповідає ідентифікаційному порядковому номеру виконавця. Так як і у випадку з функціональними діаграмами, фактичне перетворення здійснюється при наявності ПК, мануальним шляхом чи застосуванні зовнішніх апаратних пристроїв. При невизначеності процесів у верхньому шарі даних, використовується термінологія «система» і «підсистема», щоб описати систему як «ціле» або функціонально довершену частину.

Потік інформаційних даних — це передача певних відомостей від джерела до одержувача. Практично інформаційний потік може виконуватися вручну за допомогою програмного або програмно-керованого та кабельного сполучення. Контейнери вхідних та вихідних даних — це віртуальна область для сховища важливих інформаційних даних. Вони можуть подаватися у вигляді баз даних, файлових елементів, електронних та Google таблиць та у вигляді текстових сховищ. Контейнери є сховищами у яких постійно зберігаються дані, які на даний момент не потрібні для використання та зберігаються у постійному вигляді до часу їх потреби.

Використовуються контейнери, де вхідні і вихідні дані перебувають у стані очікування, після чого зберігаються та збираються для наступної обробки. На одній і тій самій діаграмі аналогічні контейнери можуть відображатися декілька раз у окремих місцях діаграми. Це зроблено для того, щоб полегшити читання діаграм і зменшити кількість рядків у схемі. DFD демонструє, що інформаційні потоки, створені сутностями, змінюються певними процесами та зберігаються в одиницях виміру даних та транспортуються іншому користувачу, який отримує дані.

Результатом є приклад зберігання та обробки даних. Вона демонструє процес взаємодії інформаційно-аналітичної системи з отримувачами та поточним користувачами, не показуючи виконавця, а саме показує графічний інтерфейс системи та демонструє її [39]. Працюючи з інформаційно-аналітичною системою визначення стану психіки, учасник обирає режим тестування, після чого система запускає йому відповідний вид тесту. Інформаційно-аналітична система визначення психічного стану людини використовує спеціально обрану і створену модель (рисунок 2.2).

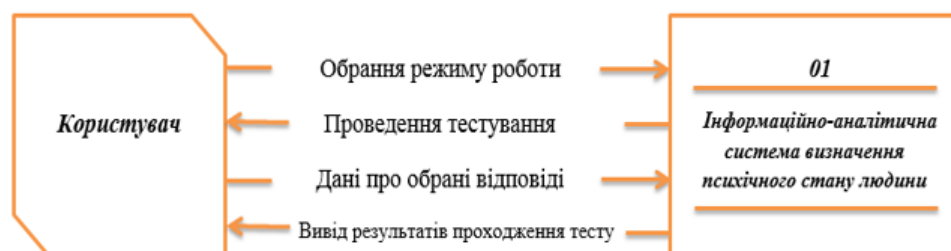


Рисунок 2.2 — Модель інформаційно-аналітичної системи

Протягом тестів система аналізує та упорядковує відповіді, зроблені користувачем. Наприкінці тесту, залежно від обраного вами типу тесту (тест Роршаха, тест Люшера або традиційний опитувальний тест) показує сформований результат. Користувач також має можливість, передивитися посібник у вигляді зображення із симптомами та супутніми захворюваннями.

На рисунку 2.3 показано DFD першого рівня. Діаграма першого порядку, що створена, має ряд пунктів, що згодом можливо буде розкласти на рівні DFD порядком нижчого, яка створює багаторівневність. Цей процес поділу триває, поки його не можна буде ефективно описати за допомогою коротких одно сторінкових специфікацій (особливостей процесу). Він демонструє два важливих процеса тестування: тест і результат тесту [39]. Перш за все користувач повинен обрати, який тест він хоче пройти, а потім у базу даних посилаються відповідні запити для надання переліку запитань.

Коли знайдено дані, користувачеві постійно задають запитання, а відповіді зберігаються в базі даних. Далі інформація у відповідях надсилається на обробку. Після проходження тестування користувачем, його відповіді зберігаються в базі даних, з якої вони надсилаються. Потім вони обробляються та аналізуються на основі результатів тесту. Після створення результатів система відображає їх у вигляді текстового повідомлення користувачеві.

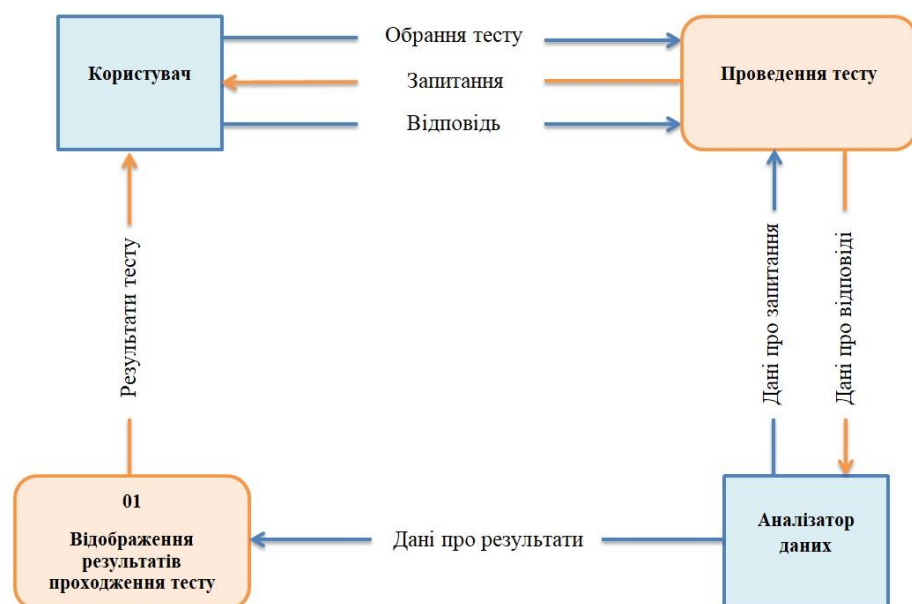


Рисунок 2.3 — DFD першого рівня

На рисунку 2.4 зображено модель процесу «Демонстрація проходження тесту та виведення результату».



Рисунок 2.4 — Модель проходження тесту та виведення результату

Для інформаційно-аналітичної системи визначення психічного стану людини була створена деревоподібна структуру процесів, що зображена у Додатку В. Деревоподібна структура процесів — це ієрархія, яка включає набір процесів і субпроцесів. В сукупності вони створюють дерево процесів, що пояснює функції в інформаційно-аналітичній системі.

Загальне призначення системи — визначення психічного стану людини. Першорівневими етапами є проходження тестів Роршарха та Люшера, які дозволяють провести дослідження сприйняття користувача та провести аналіз його особистості. Другий етап включає в себе збір, аналіз відомостей про симптоми, оприлюднення інформації з результатами пошуку, вивчення почуттів користувача та внесення пропозицій. Деревоподібна ієрархія дозволяє визначити основне призначення системи. Це дає можливість проглянути увесь процес і зосередитися лише на необхідних поточних проблемах. Процес побудови — це візуалізація набору дій або проблем, які дозволяють інформаційній системі працювати.

### 2.3 Класифікація ключового функціоналу додатку

Перед початком створення мобільного додатку необхідно чітко окреслити ключовий функціонал створюваного продукту. Час розробки, складність та вартість зумовлюються тим, які функції має здійснювати додаток. Розрізняють 4 базових типів мобільних додатків:

- корпоративні;
- системні;

- інформаційні;
- розважальні.

Корпоративним називають додаток, який призначений для спрощення діяльності компанії, швидкий обмін даними між співробітниками чи отримання інформації про саму компанію. Цільовою групою користувачів є в основному співробітники компанії, реальні та можливі клієнти та партнери. Вони мають спрощений процес створення, за рахунок застосування корпоративних елементів компаній (їх логотипів, веб-зображень). Інформаційним є додаток, який використовує різні види подання інформації (текстові, голосові повідомлення, відео зустрічі). Зазвичай їх розробка здійснюється для ЗМІ, таких як телеканали, газети, портали, радіо на канали новин.

Системні — це програми, які мають певну мету та повинні надавати спеціальні сервісні послуги, тобто вирішувати поточні задачі, які вимагають системного втручання у роботу додатку користувача. Вони мають складну постановку і вимагають великого технічного рівня для розробки, так як додатки повинні працювати бездоганно (не викликати збоїв). Розважальним — це мобільний застосунок, основна ціль якого не лише гра, але й розвиток моторики, логіки та ознайомлення із світом цифрових технологій. Їх використовують, як доповнення під час шкільних та дистанційних занять.

Для розробників має велике значення категорія створення мобільних додатків відповідно до її структури. Програми додатків можна розділити на гібридні та локальні категорії [40]. Гібриди додатки являють собою універсальні програми, що призначені одночасно для багатьох платформ і мають схожу функціональність, що не містять схожих платформ. Зауважимо, що гібридний додаток — це переважно типовий мобільний веб-сайт.

Основними перевагами є:

- кросплатформеність — додатки можуть створюватися для кількох платформних версій та при необхідності адаптуватися згідно вимог;

— мінімізація капіталу додатка — розробка такого типу програми легша і швидша, тому початкові витрати нижчі, що приваблює фірми з обмеженими можливостями;

— випуск продукту та конкурентоспроможність — у гнучкого гібридного зростання є ще одна перевага — менший час зростання, що дозволяє виробляти продукт швидше та отримувати максимально інтенсивно прибуток.

Локальні — це програми, при написанні яких використовувалися певні мови програмування, призначені для локальних платформ, які швидко й правильно розгортаються.

Переваги локальних додатків:

— з ними легко працювати — інтерфейс і графічна частина локальних програм мають класичний дизайн, підлаштований під конкретну операційну систему;

— швидкодія — програми з додатками оптимізовані для певних мобільних операційних систем, що робить їх швидшими та стабільнішими.;

— більш широкі можливості — ця версія програми включає в себе всі функції ОС та апаратні компоненти (камеру, GPS, адресну книгу, календар).

Інформаційно-аналітична система визначення психічного стану розроблена для дослідження психічного стану людини, є мобільним додатком, який надає послуги самостійно, так як написаний для певного середовища (Android) і використовується користувачем, щоб визначити його психічну урівноваженість та стан.

Розробка мобільного забезпечення — це процес створення додатків, зокрема смартфонам та планшетам різної роздільної здатності. Вони завантажуються на пристрої в процесі виробництва, інсталиуються користувачами через різні платформи розповсюдження ПЗ, або вони є веб-додатками, які обробляються на клієнтській чи серверній стороні. В наш час існує різноманітний вибір мов для розробки мобільних додатків [40]. Це тому, що різні операційні системи вимагають різних мов програмування для використання на кожному мобільному пристрої. Часто використовувани

платформи (або кілька платформ) iOS, Linux, Android і Windows значно впливають на мову, що використана у розробці. Наприклад, можливо створювати різне ПЗ для окремих платформ чи використовувати допоміжні інструменти для того, щоб оптимізувати програми у різних середовищах. Інший варіант — заощадити час і зусилля, але це може вплинути на зручність використання. Сучасні мобільні пристрої пропонують широкий спектр вибору типів дизайну, відповідно до цінової політики.

Розберемо основні технології, які використовуються для розробки додатків. Однією з них є Java two Macro Edition (J2ME). Ряд цих специфікацій і характеристик що розроблені для різних типів переносних апаратів. Виділяють такі області — конфігурація підключеного пристрою (CDC) і обмежена конфігурація підключеного пристрою (CLDC). Пристрої, які проводять конфігурацію підключеного пристрою, є більш досконалішими. Обмежена конфігурація підключеного пристрою містить звичайні мобільні телефони з більш скромними ресурсами. Він дозволяє налаштувати функцію конфігурації різних типів пристроїв при допомозі спеціальних режимів. Профіль мобільного інформаційного пристрою (MIDP) розроблено для портативних пристроїв CLDC, за допомогою яких можна спілкуватися. Реалізація J2ME базується на CLDC і MIDP. Java скрипти інтерпретуються безпосередньо пристроєм за допомогою віртуальної машини, яка працює на всіх пристроях, подібних до платформи Java, що дозволяє вільно поширювати програми. Java програмування залишається найпопулярнішим, тому що на більшості пристроїв (особливо мобільних телефонів) наявна Java платформа.

Qt технологія — ще одна популярна платформа, яка являється локально та дозволяє використовувати програми на різних ОС та мобільних пристроях, включаючи Linux, Symbian, Windows, Android, та багато інших. Із зростанням бази користувачів зростає попит на мобільні додатки та розробку інтерфейсу. Qt є однією з найуспішніших функцій C++. Встановлення мобільних пристроїв оптимізовано при допомозі емулятора як частини середовища. Тому є можливість написання більш складних для мобільних пристроїв, пристосовуючи бібліотеки C++ і платформній супровід. Для нормальної

роботи платформи на мобільних пристроях він повинен бути встановлений на базі сучасних фреймворків для розробки ПЗ.

Іншою технологією є розроблена технологія Windows Phone SDK. Розроблений програмний код порівнюється в XAML. Хоча цей файл є XML лише з мовою тегів XAML. Windows Phone — середовище для мобільних машин, що містить не тільки технічну експертизу, але й комплексну концепцію інтерактивності та інтерактивності з користувачем системи Metro. Оновлення для Windows Phone реалізовано в магазині додатків Store. Модифікації та тестування мобільних додатків Windows Phone виконуються за допомогою симулятора Windows Phone. Одним з технологічних досягнень є iPhone SDK. Оновлення до iOS доступне в Mac OS X.

Проте в Інтернеті можна отримати дані, як оновити на комп'ютерах Macintosh чи уявній машині. Відзначається, Apple пропонує безкоштовні інструменти, але приступна плата за підписку. Потрібно користуватися Objective-C для створення забезпечення на iPhone. Таким же чином можна кодувати на C і C ++ (вам потрібно змінити довжину та встановити дозволене розширення). Але повністю видалити Obj-C є неможливим, основна маса API є Obj-C, за винятком OpenGL (деякі рядки коду в Obj-C необхідно використовувати для початку), а стандартні репозиторії C / C ++ повністю доступні. Програма налаштовується за допомогою режиму XCode та вбудованого симулятора [40].

Можливе використання Android Studio з плагінами ADT і SDK для розробки для app. Розробка ведеться на Java, що є притаманною для даного середовища. Середовище можна налаштувати за допомогою вбудованого емулятора ADT або безпосередньо на своєму смартфоні. Версії SDK відрізняються для написання коду під різні прошивки Android. Доступна підтримка, всіх актуальних версій. Крім розробки на Java, можлива підтримка за допомогою Android NDK (Native Development Kit). Половина усього населення — користувачі Android платформи. На відмінну від веб-додатків акцент був зроблений на інтерфейсі, а також на його інтуїтивному та зручному



використанні. Аби не втратити прагнення до щоденного використання, важливо регулярно розробляти та покращувати продукт.

Android дає розробникам додаткову можливість вибрати ОС, в якій вони будуть створювати програми. Програмісти, які створюють на Android можуть використовувати Linux, Windows або Mac. При розробці інформаційно-аналітичної системи визначення психічного стану використовувалася платформа Android Studio, так як вона є найбільш адаптивною, зручною та багатофункціональною для створення мобільних додатків.

## 2.4 Принципи розробки та структура Android-додатків

Операційна система Android є безкоштовною системою, що базується на платформі Linux, яка використовує 2D і 3D графіку, а також інтерфейс програмування Java, доступний у SQLite, з використанням бібліотек Android і OpenGL. Програми працюють самостійно, тому є відокремленими від інших розроблених програм, які запобігають зламаній програмі спричиняти критичну шкоду другим програмам [41]. Android застосовує механізми руху на основі Intent. Коли необхідно виконати дію, наприклад надіслати електронний лист, здійснити дзвінок чи відправити SMS повідомлення, застосовуються функції Intent. Провайдери контенту застосовуються під час обміну інформації серед додатків. Щоб створити проект найчастіше приміняють середовище розробки Java та SDK Tools.

SDK Tools вміщує усе те, що необхідне для розробки, компіляції та дебагу програм на Android. Ці додатки в основному розроблені з використанням Java. Він містить виправлення Debug Bridge Android, що дає змогу інтегрувати та налагоджувати Debug Bridge Android на віртуальному або реальному пристрої та запускати програму. Архітектура додатків ґрунтується на основі багаторазового використання ключових компонентів так званого блоку структури.

Середовище розробки налагоджена так, що кожна програма може компілювати в собі частини інших програм. Щойно системний компонент запускається, він запускає власну програму та створює потрібні Intent. Тому

Android не має єдиної точки доступу, які є притаманні іншим системам. Так як кожна програма — індивідуальний продукт, то розробник може накладати обмеження на доступ до файлів та блокує змогу активувати частину іншої недоступної програми. Для його активації необхідно надіслати в систему сповіщення про бажання активувати частину функціоналу, після чого система почне працювати [42].

Ключові компоненти ієрархії додатку зображені на рисунку 2.5.

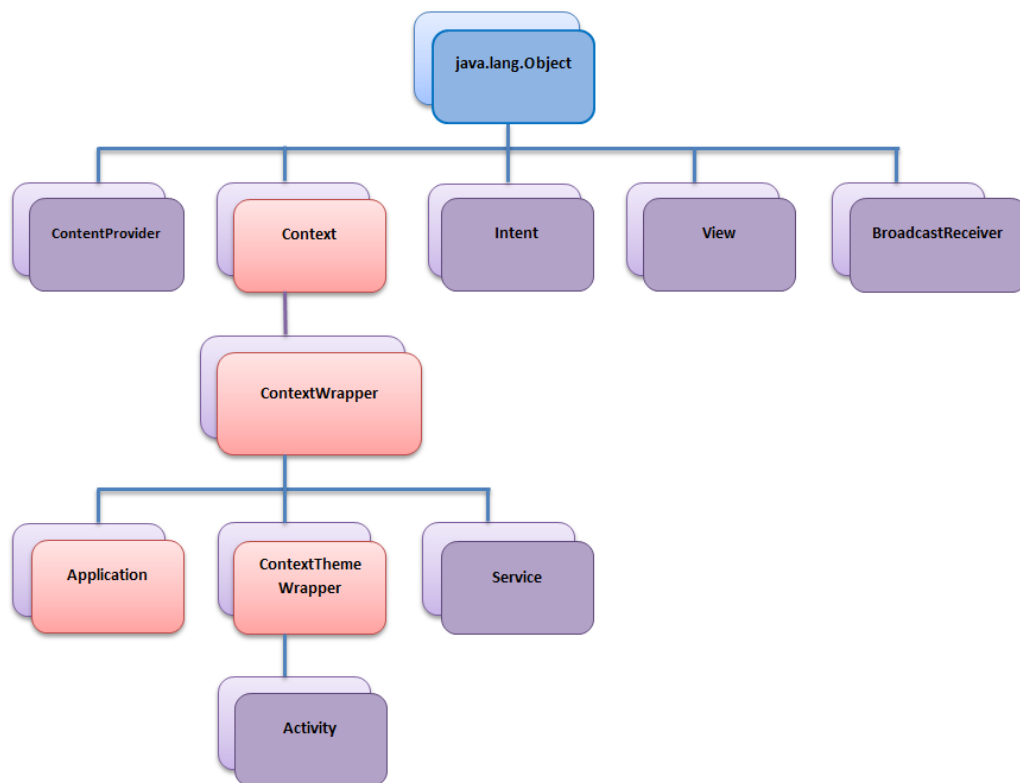


Рисунок 2.5 — Ключові компоненти ієрархії додатку

Загалом, додаток має такі складові:

- Java бібліотеки, що являються класами даних таких як (Content Provider, Activity, Context, Intent, BroadcastReceiver, Service), які не мають наявності завантаженого SDK за замовчуванням;
- `android.manifest`, список джерел (рядки, зображення тощо) та файли;
- діяльність (activity) — клас графічного інтерфейсу, що визначається програмно має розширення XML.

Програма для Android складається з ряду дій, які можна використовувати між ними під час виконання. Програма хоч і показана

повністю, але не всі види діяльності взаємозалежні. Тому є можливість запустити кожен елемент наступної програми, що може отримати доступ до діяльності цієї програми. Наприклад, програма Камера може мати дію, яка відображає обраний список швидкого доступу, для зручності та швидкості надсилання зображень близьким людям.

Класи `activity` формують порядок необхідних станів для життєвого циклу додатка:

- активний і виконуваний — хоча інтерфейс користувача знаходиться на передньому плані, він видимий і сфокусований;
- пауза — користувальницький інтерфейс досі видимий, але не в фокусі;
- аварійний — це, якщо користувальницький інтерфейс, вже невидимий.

Код не працює, так як додаток аварійно завершився. Якщо робота знаходиться в аварійному стані або в режимі паузи, то негарантовано, що об'єктні данні будуть повністю збереженні [43]. Служби(`Services`) — ключові компоненти додатку, які працюють у початкових етапах додатку та виконують локальні операції та працюють віддалено. Вони не відповідають за створення користувацького інтерфейсу, але мають змогу до відтворення зовнішніх аудіо на фоні, не перериваючи діяльності користувача, навіть в інших програмах. Є можливість запуску служби з іншого компонента, наприклад, активності, а потім працювати з ним.

Постачальники контенту (`Content-providers`) — це особи, які надають загальний потік інформації для програми. Накопичений контент міститься в базі даних `SQLite`, провіднику файлів, хмарному сховищі чи іншому місці для програмного забезпечення. `Content-providers` дозволяє робити запити чи змінювати інформацію, при умові конфіденційного доступу до додатку.

Приміром, `Android` надає постачальника контенту, що керує контактною інформацією юзера. Будь-яка програма, що має права доступу, має право вимагати від вас підтвердження надання особистої інформації, зчитування персональних даних з вашого мобільного пристрою. Постачальник контенту

— це підклас `ContentProvider`, що специфікує API, які дозволяють решті програмам здійснювати необхідний обмін.

`Broadcast Receiver` — це ключовий компонент додатку, що отримує широкоформатні мовні сповіщення, які виробляються системою. Наприклад повідомлення про те, що екран погас або заряд акумулятора низький. Такі повідомлення та додатки можна організувати. Наприклад, передається сповіщення, про надходження та доступ даних, іншими програмами для використання і обробки.

Такі сповіщення служать кабелями для інших компонентів і специфікуються для того щоб виконаний обмежений рівень роботи. Наприклад, для запуску служби, яка підходить для окремих подій. Програма має можливість зареєструватися як приймач певних подій і може бути виконана при наявності події, що виконується [44]. Намір — це клас, який використовує об'єкти для передачі повідомлень, які називаються `Intents`, між основними компонентами програми.

`Inteny` — це асинхронне повідомлення, яке дозволяє програмі запитувати функціональність від іншої служби або події. Програми можуть безпосередньо запитувати послуги чи дії або викликати служби та програми, зареєстровані на `Android`. `Intent`, як екземпляр — це форма даних, яка описує дію, яку потрібно виконати, і зазвичай використовується для запуску діяльності або служби. Для одержувачів цей запис містить характеристику події, яка сталася або була сповіщена. У відповідь `Android` знаходить відповідний компонент, наприклад прослуховував активності, службу або прослуховування трансляції, і запускає його за необхідності.

Сповіщення про `Intent`, відправлені конкретному компоненту, отримується виключно для нього. Коли одна з діяльність починає запускати іншу, то вона має намір, що описує дію, і надсилає його до методу `startActivity()`. Після чого система `Android` здійснює перевірку, чи відповідає програма наміру та знаходить те, що їй потрібно, після чого запускає кінцеву дію. Метод `onCreate()` виконується і надсилається в `Intent` (рисунок 2.6) [44].

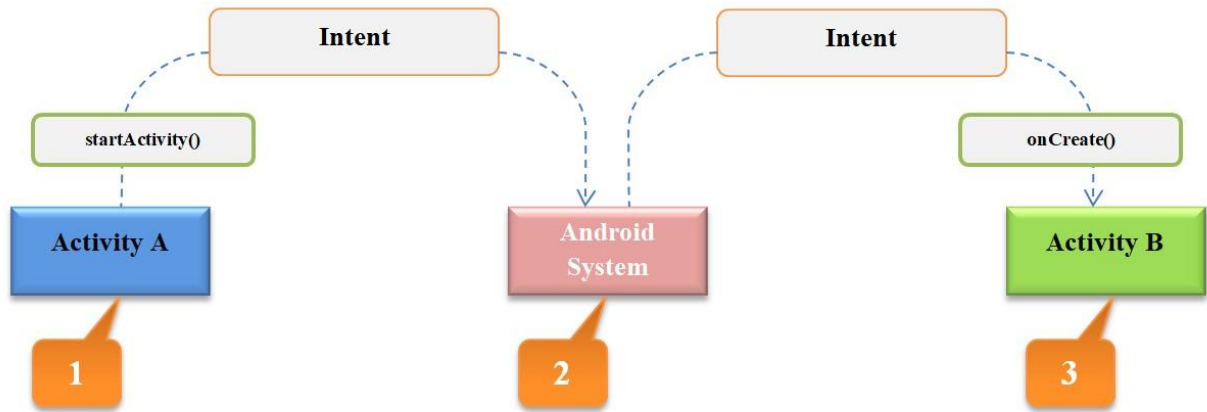


Рисунок 2.6 — Модель взаємодії Activity з Intent

Таким чином програма може отримати контактну інформацію або зробити запит через програму Intent на запуск браузера. Перегляди (view) — є класом, що створюють блок компонентів інтерфейсу користувача та визначають область екрана у вигляді прямокутника, яка відповідає за графічний та користувацький інтерфейс. View являється батьківським класом (кнопок, чек-боксів, полей форм вводу тощо) ViewGroup, що є невидимим для всіх спадкових елементів та розробляє специфікацію для місце розташування компонентів інтерфейсу користувача.

Користувацький інтерфейс є структурою зображень переглядів (view) та визначається при допомозі XML файлу, що автоматично передається під час виконання до дерева пов'язаних об'єктів (рисунк 2.7). Маніфест Android — це файл XML, що є головною частиною використання кількох функцій, а саме:

- вказує назву пакета Java програми, яка є унікальним ідентифікатором програми;
- описує складові програмного забезпечення, такі як діяльність, послуги, мовники та постачальники контенту, який визначає ім'я класу, який реалізує кожен компонент, і оголошує його функціональність;
- забезпечує процес, що містить програмні компоненти;
- оголошує необхідні дозволи на доступ до складових програми;
- повідомляє найменший рівень API, який необхідний для програми;
- перераховує бібліотеки, на які програма має посилатися.

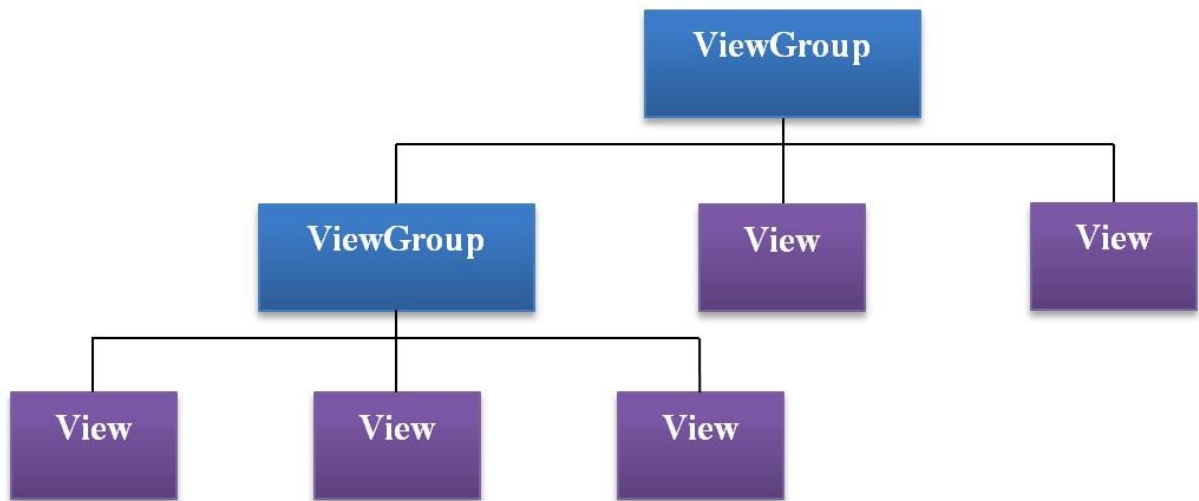


Рисунок 2.7 — Дерево пов'язаних об'єктів

Android додатки застосовують ресурси, такі як ілюстрації, графічність інтерфейсу, оголошення меню (файли XML) і текстові форми вводу. Ресурс визначається як рядок, що може містити шлях, назву файлу, та його зображення. Під час розробки додатку створюється абстрактний клас Java названий R, що вміщує статичний набір даних кінцевої точки. Кожен з них ссилається на відокремлений ресурс, що використовується програмним кодом для того щоб зв'язатися з ресурсом. Це допомагає знайти недоліки в посиланнях на ресурси під час обробки та створення програмних продуктів.

Додатки Android застосовують різні види файлів, включаючи файли загального призначення, файли бази даних, кеш-файли та двійкові blob (система кодових файлів, яку рекомендовано додати до продукту) [45].

## 3 СТРУКТУРА ТА ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

### 3.1 Вибір інструментарію для розробки продукту

Сучасний етап еволюції технологічного прогресу характеризується різноманіттям вибору мов програмування. Для розробки мобільних додатків під Android існують такі мови, як C++, Java та C#, які є одними з найуживаніших. Кожна з зазначених мов містить ряд відмінних характерних особливостей, що робить їх унікальними.

Створюючи жорсткі обмеження на варіанти передачі аргументів, обрана для інформаційно-аналітичної системи мова Java є найкращим вибором, яка значно полегшує розробку додатку, що значно відрізняється від популярної мови C++, де аргументи є обмежені та їх можна передавати виключно за значенням чи посиланням при врахуванні покажчиків, що дуже заплутує програмний код, роблячи його максимально нечитабельним.

Java переважає інші мови програмування, оскільки немає:

- аргументів з доданими покажчиками;
- абстрактних бібліотек та заголовних файлів;
- імплементації.

В результаті великого досвіду розробки мов програмування, зокрема таких як C++ та Java, було спроектовано C#, як мову програмування, яка є за своїми ознаками до мови Java. Мови Java і C# можна означити як вдалу спробу покращення мови C++. Функції C# та Java:

- програмний код є проміжним, який може бути інтегрований в спеціальні обрані для нього середовища та може адаптуватися під різні системи (не залежно від платформи);
- не валідність застосування покажчиків та сортування сміття (garbage) у мові Java та обмеження застосування покажчиків у C#;
- підтримка багатопоточності;
- розширені бібліотеки функцій та класів;
- легка реалізація користувацького інтерфейсу, за допомогою успадкування нащадків та класів;

- містять параметр «new» для динамічного створення об'єктів і реалізації принципів ООП;
- відсутність універсальних і безперервних функцій, елементи розрізняються лише в межах одного класу;
- масиви з можливістю динамічної обробки;
- оператори «->», «::» не використовуються;
- поняття null і boolean / bool можна легко використовувати;
- кожне значення має бути ініціалізовано перед використанням;
- нескінченні числа не використовуються для керування виразами.

Незважаючи на те, що C# і Java мають багато спільних функцій, їх не можна вважати ідентичними. C# аналогічним чином може імпортуватися з Java, але сьогодні він підтримується на значно меншій кількості платформ. Java дає змогу формувати додатки у байт-код, який застосовується у різних диференційних середовищах та здійснює компіляцію проміжного уявлення про додаток [45].

Головні найкращі функції мови програмування Java:

- незалежний контроль від платформи, на якій проектуються задачі (цей код можна запускати на Windows, Linux, Ubuntu чи Macintosh);
- має класичний набір символів, який походить від мови C, що дозволяє легко переходити з однієї мови програмування на іншу;
- є мовою ООП у основі якої лежать парадигми програмування;
- базується на об'єктах, крім деяких ( primitive types);
- безпечність, яка полягає у досконалій системі підказок, що була сформована з бази усіх відомих помилок різних мов програмування.

Для надійності мови Java було здійснено наступне:

- автоматична можливість звільнення явно виділеного механізму очищення сміття та вивільнення додаткової пам'яті;
- відсутність можливості очищення даних пам'яті;
- заборона покажчиків;
- різниця між операторами присвоєння та порівняльними операторами;



— введення можливості створення користувацьких інтерфейсів, замість множинного наслідування, уникаючи проблем при створенні складних структур класів.

Провівши аналіз зазначених переваг було прийнято рішення розробити інформаційно-аналітичну систему визначення психічного стану людини використовуючи Java. Найзручнішими середовищами для розробки мобільних додатків з використанням Java є Eclipse і Android Studio. Eclipse — є багатогранним середовищем для розробки. У ньому присутня можливість Java розробки. виправити дефекти можна в емуляторі ADT або мобільному пристрої безпосередньо за допомогою Android.

Для характеристики графічного інтерфейсу програмування використовується інша мова — XML (мова розмітки). Плагін ADT дає можливість проектувати інтерфейс додатків Android без ручного застосування XML. Також даний плагін інтегрує середовище Eclipse з симулятором пристрою Android. Симулятор є віртуальним аналогом смартфона з зменшеним функціоналом.

Android Studio є відносно новим середовищем для розробки програм Android, засноване на платформі IntelliJ IDEA JetBrains. Воно адаптоване для того щоб виконувати класичні задачі, які необхідно вирішувати під час створення програми для платформи Android.

Це середовище включає інструменти, які спрощують процес тестування ПЗ на сумістимість з різними варіантами платформи, а також інструментарій для проектування додатків, які діють на пристроях з різним розподільним розширенням екранів, зокрема мобільні телефони, нетбуки, VR, планшети, смарт-годинники та інше.

На додаток до функцій IntelliJ IDEA, Android Studio створила кілька додаткових функцій, таких як система для створення, та компіляції додатків використовуючи Gradle інструменти, які інтегровані в середовище [45].

Для створення програми застосовують емулятори для імітації роботи Android пристроїв, що дозволяють уникнути фізичної взаємодії з пристроями, та протестувати їх віртуально. Щоб пришвидшити процес розробки програми

було створено набір традиційних елементів інтерфейсу та візуальний редактор, який дозволяє легко бачити різні ситуації інтерфейсу програми (наприклад, в інтерфейсі можна побачити, як шукати різні версії. Android і екрани різних розмірів).

Існує майстер для розробки власних елементів дизайну, які підтримують використання шаблонів для створення незвичайних інтерфейсів [46]. Середовище розробки Android Studio було вибрано для розробки мобільних додатків, оскільки воно є більш актуальним, відмінно працює, легкий у проектуванні, включає програмування на Java та повністю відповідає вимогам розробки додатків.

### 3.2 Початкова конфігурація додатку

Мовою програмування для розробки додатку аналітично-інформаційної системи під Android вирішено обрати Java, а для проектування інтерфейсу користувача — XML. Створимо першу програму в середовищі Android Studio для операційної системи Android (рисунок 3.1). Відкриємо Android Studio і на початковому екрані оберемо пункт New Project:

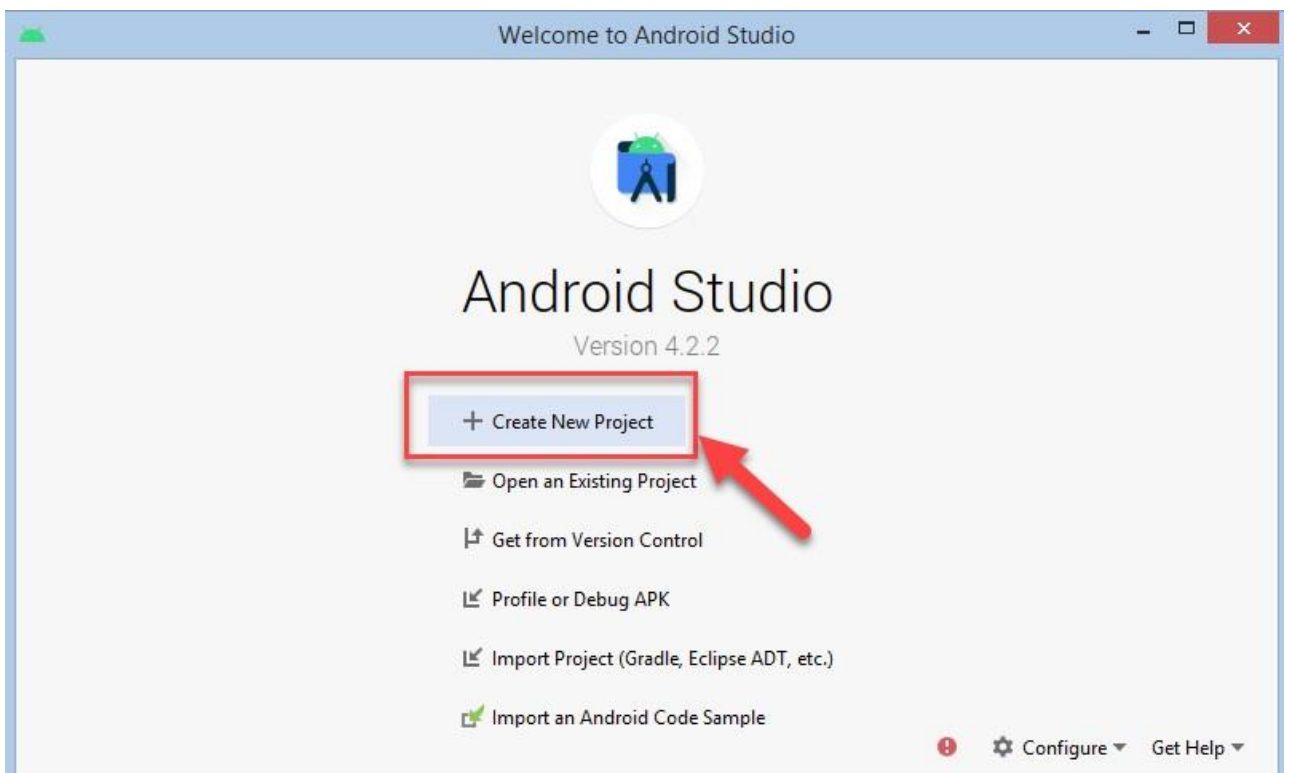


Рисунок 3.1 — Створення початкового проекту

Під час створення проекту середовище запропонує нам вибрати один з готових шаблонів проекту ( рисунок 3.2)

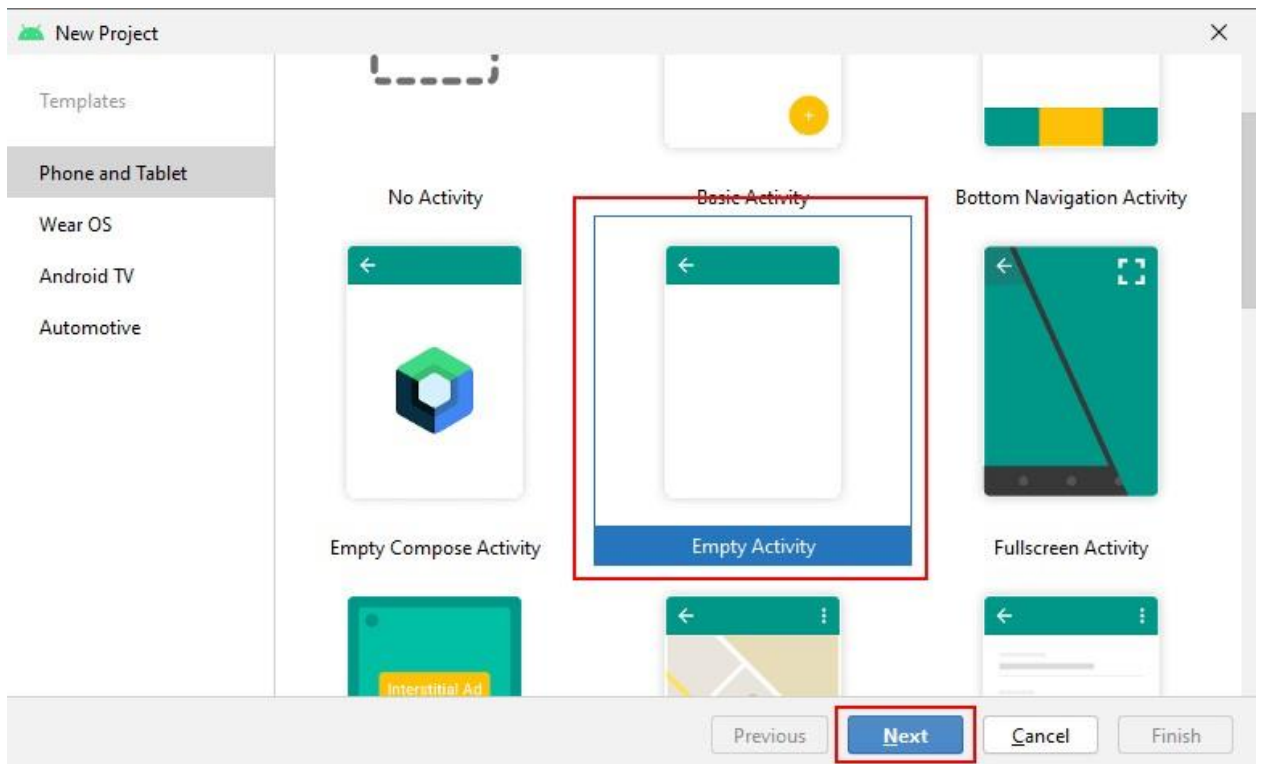


Рисунок 3.2 — Вікно вибору готових шаблонів проекту

Шаблони які пропонує програма значно економлять час впродовж написання примітивної частини коду. А також є можливість створення власного варіанту шаблону при виборі Add No Activity у якому набір шаблонів буде відсутній.

Варіанти шаблонів постійно змінюються і оновлюються.

Основні шаблони:

- Blank Activity — шаблон для стандартних смартфонів;
- Primary Activity — шаблон, який використовується у планшетах , та у режимах з поділом екрана на 2 частини;
- Fullscreen Activity — шаблон, який використовується при написанні ігор з великими розширеннями;
- Google Maps Activity — шаблони для роботи з картами [46].

Оберемо в цьому списку Empty Activity — шаблон, який містить найпростіший функціонал та натиснемо «Next».

Після цього можемо побачити сторінку створення проекту (рисунок 3.3).

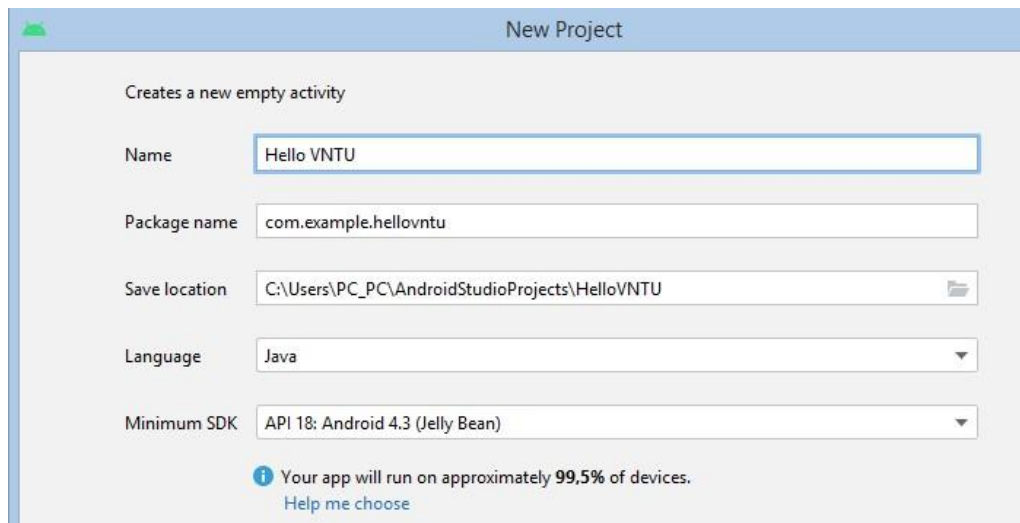


Рисунок 3.3 — Вікно головних налаштувань проекту

Вікно створення нового проекту містить наступні поля:

- назва програми — вкажемо, як назву Hello VNTU;
- ім'я пакета, де розміщуватиметься головний клас програми. В даному випадку для тестових проектів це значення не має значення, тому встановимо `com.example.hellovntu`;
- розташування файлів;
- мову Java, обрану для створення системи (за замовчуванням Kotlin);
- версія SDK.

Після завершення введення даних середовище створить новий проект та структуру, яка містить всі необхідні файли ( рисунок 3.4).

Перш за все бачимо ієрархію, яка складається з 3 основних папок:

- `manifests`, який зберігає в собі файл `AndroidManifest.xml`, що описує конфігурацію програми та визначає кожен із компонентів цієї програми;
- `java`, яка зберігає файли класів коду мовою java, впорядковані за окремими пакетами, таким чином, у папці `com.example.hellovntu` є за замовчуванням файл `MainActivity.java` з кодом на мові Java, який виражає клас `MainActivity`, що запускається за замовчуванням;
- `res`, який має ресурсні файли, які використовуються в додатку і діляться на різні файлові підпапки [46].

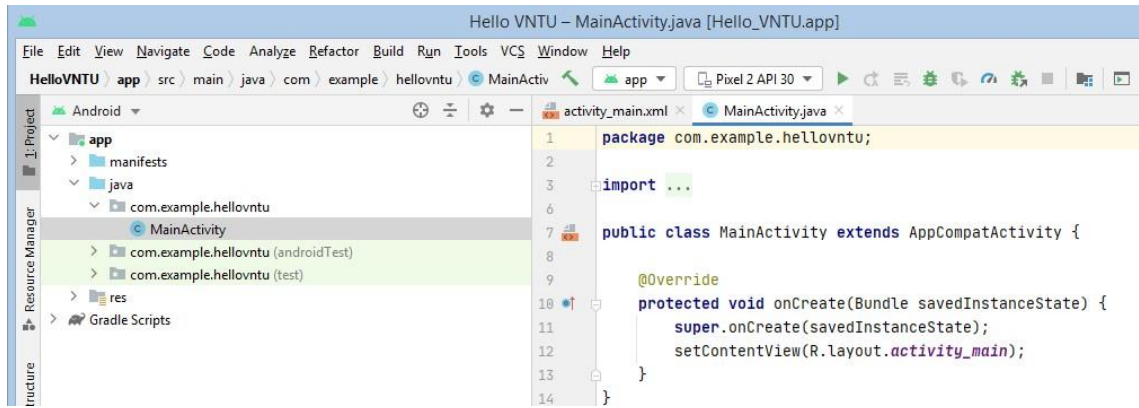


Рисунок 3.4 — Файлова структура створеного проекту

Відокремлений елемент Gradle Scripts включає в себе ряд скриптів, що можна використовувати при проектуванні програми. Вся ця структура має бути спільною з файлом MainActivity.java, який відкритий у студії Android і містить логіку програми, з якої починається реалізація програми. Ми також віділимо файл activity\_main.xml, що визначає користувацький інтерфейс. Система файлів Android пописується методами на базі XML. Файли приміток — це назва кожного класу зображення вузла та дерево елементів XML, розташоване в папці «res / layout». Під час програмування створіть кнопку, прапорець, текстове поле, тощо. Також він додає різні компоненти, які потім відображаються на екрані. Створити можна візуально, перетягнувши елементи мишею (дизайн програми в Android Studio) (рисунок 3.5).

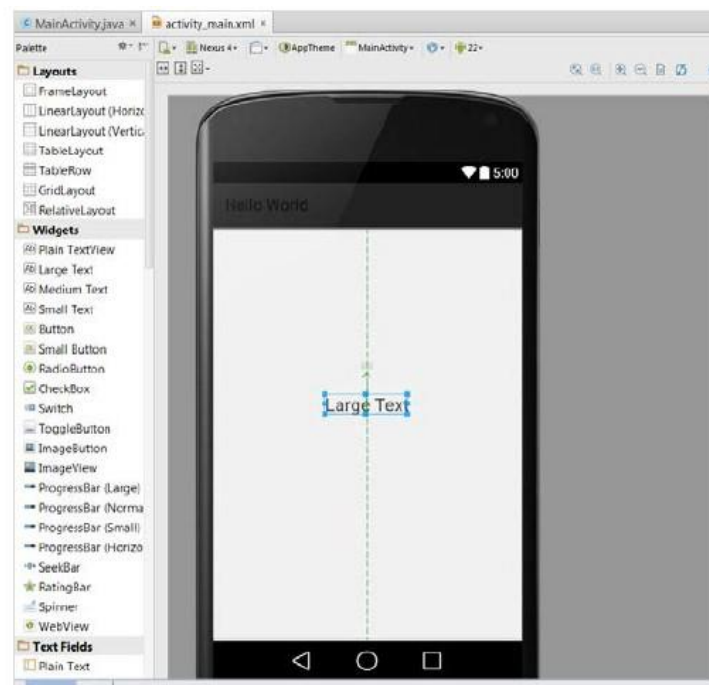


Рисунок 3.5 — Налаштування розмітки програми в Android Studio

Панель редагування елементів розташовано у лівій частині інтерфейсу програми. Вона дозволяє створювати та редагувати графічні елементи, зображення та повзунки [46]. У правій частині інтерфейсу розташовано дерево представлення елементів (Component Tree), що дозволяє задавати різноманітні властивості визначеного елемента (Properties). Дивлячись на текст, побачити проекцію правого боку екрана та переглянути вигляд додатку, після зміни властивостей чи додаванні нових компонентів. Рисунок 3.6 ілюструє, як компонент TextView змінили вручну, сформувавши нове текстове поле, додавши на нього кнопку зображення. Фон також був видозмінений.

Після завершення проектування є можливість провести симуляцію запуску телефону або встановити файл арк персонально на свій телефон.



Рисунок 3.6 — Налаштування текстових компонентів в Android Studio

### 3.3 Розробка структури програми

Еволюція сучасних інформаційних технологій розвивається на основах взаємозв'язку з різними сучасними базами даних. Вони дозволяють зберігати та впорядковувати великі об'єми даних, що дає змогу раціонально розташовувати та логічно систематизувати інформацію, яка дає можливість до

різного роду навчатися та взаємодії. БД лідирують на ринку програмного забезпечення серед усіх можливих носіїв даних.

Інформаційно-аналітична система визначення психічного стану, призначена для діагностики психічного стану людини, повинна містити назви симптом та опис психічного захворювання. Для вміщення великої кількості психодіагностичних даних необхідно використовувати бази даних.

БД — набір інформаційних даних, пов'язаних із загальною ознакою або параметром [47]. Android застосовує SQLite, найпопулярнішу, просту у використанні базу даних. У базі даних з'єднання таблиці створюються так, щоб між ними був взаємозв'язок. Інформаційні таблиці тісно пов'язані між собою, та при потребі мають здатність передавати сутності від одної таблиці до іншої.

SQLite — спрощена система керування БД, яка працює як бібліотека. Характерною ознакою є те, що вона не застосовує серверні параметри, тобто платформа SQLite надає бібліотеку, де програма і двигун є частиною програми, а не окремим процесом, з яким програма взаємодіє. Протоколом обмін є набір спеціальних функції для обробки SQL запитів. Виконання наступних функцій скорочує час відповіді та полегшує програму. Процедурою зберігання бази даних в цілому є процес її стандартизації на жорсткому носії, який містить усю інформацію (табличні значень, персональні дані, індексацію документації, перелік доступних програм). Проста реалізація реалізується тим, що обсяг файлу в цілому, в якому зберігається база даних, забороняється транзакцією [47]. Функції ACID в основному досягаються шляхом запису файлу журналу.

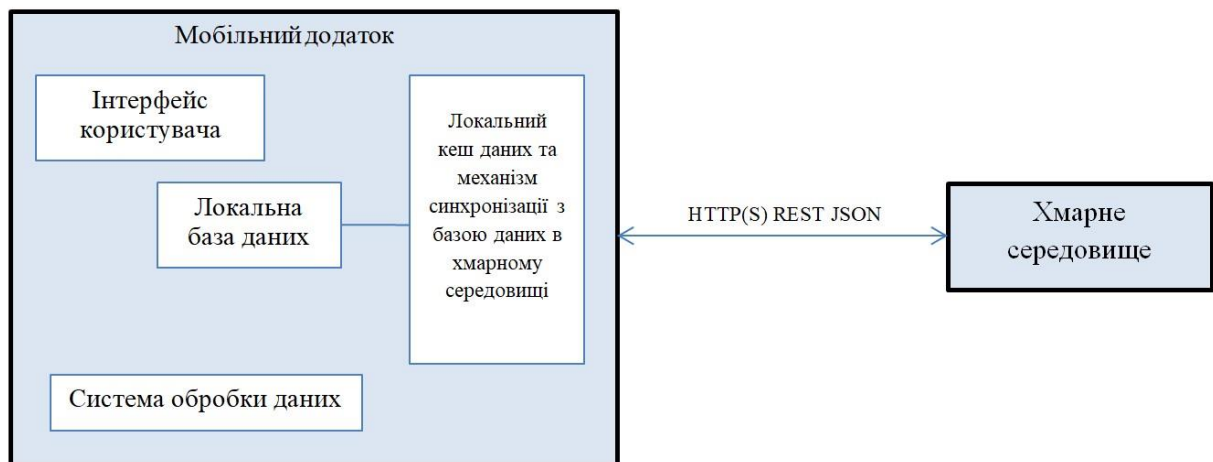
Функції системи керування БД:

- встановлення без налаштування, не вимагає встановлення чи керування;
- стандартизація усіх файлів на конкретному носію;
- нескладний і зручний API;
- відсутність залежності від зовнішніх факторів;
- Java файл, що можна інтегрувати в будь-який процес.

Безумовні переваги, які має розробник та користувач при використанні SQLite:

- немає необхідності застосовувати додаткового ПЗ;
- підтримка Java функціоналу;
- передача структурних таблиць між системами управління;
- кодування програми SQL запитам, що є з зручним фактором при зміні та переносі програми у інше середовище.

Мобільний додаток вимагає локальної бази даних для зберігання списку захворювань і симптомів. Синхронізація відбувається запитам HTTP за допомогою серверу. Мобільний додаток відображає інформацію про психічний стан людини, проводить аналіз симптомів що, зберігаються у таблицях БД. На рисунку 3.7 показано принцип взаємодії додатку з хмарним середовищем за допомогою REST запитів [47].



Рисунку 3.7 — Модель взаємодії додатку з хмарним середовищем

### 3.4 Розробка алгоритму роботи мобільного додатку

Алгоритм — набір інструкцій, який точно і чітко описують черговість дій для досягнення результату. Розробка інструкцій включає в себе наступні компоненти:

- вихідний набір даних для завдання;
- потенційні користувачі;
- ціль розробки;



— перелік дій розробника.

Розрізняють 2 способи розробки алгоритмів. Метод частинних цілей полягає у тому, що вихідна складна задачі поділяється на підзадачі, які є окремими задачами і можуть бути вирішені просто. Якщо одна з таких підзадач виявляється складною то відбувається аналогічний її поділ на підзадачі. Цей процес триває до тих пір, поки підзадачі не стане легкою та простою. Після чого вирішення окремих елементарних підзадач поєднується у єдиний алгоритм розв'язання початкового завдання. Цей метод широко використовується, так як він дозволяє створити унікальний алгоритм для ряду програмістів, які працюють із вашими локальними підзадачами. Вона є необхідною для швидкого розвитку ПЗ.

Метод відпрацювання назад полягає у тому, що розв'язування завдання починається не зверху (тобто спочатку) а знизу (тобто з цільової точки). З моменту визначення цільової точки розробник досягає мети шляхом розробки «проміжних пунктів», як допоміжних алгоритмів. Метод спроб та помилок полягає у тому, що після перевірки усіх можливих варіантів розв'язку отримуємо бажаний результат після відкидання їх у випадку невдач.

Базові вимоги до алгоритму:

— точність — метод розрахунку повинен являти собою покрокове виконання ряду простих кроків, для виконання яких, розрахунок повинен займати обмежену кількість часу, а це означає, що перетворення вихідних даних у результат відбувається за певний час;

— визначеність — кожен послідовний крок у кожен момент точно визначається системним станом;

— наочність — алгоритм повинен мати лише ті кроки та ті команди, які в ньому є і включені в систему команд;

— досконалість — метод розрахунку з правильним набором даних повинен завершити роботу і дати кінцевий результат;

— універсальність — метод розрахунку повинен відповідати різним вихідним даним;

— ефективність — отримання кінцевого результату.

Залежно від призначення алгоритми поділяються на такі типи:

- механічний, що реалізує конкретні функції, позначає їх у єдиному та надійному порядку, таким чином дає чіткі, бажані результати, якщо суть є завданням, розробленим алгоритмом;
- ймовірнісний, дозволяє програмі розв'язувати задачу різними варіантами або методами, що приводять до потенційного досягнення мети;
- евристичний, дозволяє використовувати варіації міркувань без точного та чіткого пояснення;
- лінійний, дозволяє своєчасно виконувати набір перелік інструкцій одну за одну;
- гілчастий, що містить принаймні 1 умову, яка ділиться на декілька паралельних розгалужень;
- циклічний, що передбачає кілька повторів кожної ідентичної дії при нових початкових даних.

Інформаційно-аналітична система визначення психічного стану людини являє собою гілчастий алгоритм, так як має кілька умов для паралельного виконання. Алгоритм інформаційно-аналітичної системи визначення психічного стану зображена в Додатку Д.

### 3.5 Підтримка баз даних інформаційно-аналітичної системи

Вбудованою підтримкою середовища Android Studio є спрощена версія БД SQL, яка підтримує усі доступні функції SQLite і наявні версії API, що встановлюються з програмою за замовчуванням і є класичною версією. Програміст повинен виконати конфігурування усієї бази, реалізуючи побудову та перевірку версій [48]. Повинна бути зазначена назва та платформа зі з'єднуваною БД для встановлення зв'язку з програмним додатком.

Варіанти можливих подій при з'єднанні:

- відсутність БД, що часто трапляється при стартовій інсталяції програми — в цій ситуації, спочатку повинно бути проведено початкову конфігурацію таблиць та даних, записуючи їх в тимчасову кеш-пам'ять, після

чого ви зможете дістати необхідну вам інформацію для формування нової безпомилкової БД;

— не обслуговувана БД, що часто трапляється коли відбулось оновлення програми, без підтримки старої версії БД — у такій ситуації новостворена програма потребує обов'язкових полів для заповнення, які відсутні у новій версії, але присутні у попередній;

— нова оновлена БД — з'єднання з базою даних проходить вдало, а програма працює належним чином [48].

Для інформаційно-аналітичної ситсеми визначення психічного стану була інтегрована локальна спрощена БД, де був проведений запис усіх необхідних даних (хвороб, симптомів, діагнозів та можливих варіантів запитань для визначення психічного стану), частина яких перерахована далі. Весь перелік зображено у Додатку Е.

private static final int VISUAL\_HALLUCINATIONS = 1; // зорові галюцинації

private static final int EMOTIONS\_CHANGING = 2; // зміна емоційних станів

private static final int DELIRIUM = 3; // марення

private static final int AUDITORY\_HALLUCINATIONS = 4; // слухові галюцинації

private static final int INFANTILE = 5; // дитинячість

private static final int TOMFOOLERY = 6; // дуркуватість

private static final int APATHY = 7; // апатія

private static final int WEAKNESS = 8; // млявість (фізична слабкість)

private static final int PESSIMISM = 9; // песимізм

private static final int BAD\_MOOD = 10; // поганий настрій

private static final int LOW\_SELFESTEEM = 11; // низька самооцінка

private static final int BEDILITATION = 12; // виснаження

private static final int HELPLESSNESS = 13; // безпомічність

private static final int IRRITABILITY = 14; // дратівливість

private static final int RETICENCE = 15; // мовчазність

```

private static final int QUITL = 16; // почуття провини
private static final int KINDLING = 17; // збудження
private static final int EUPHORIA = 18; // ейфорія
private static final int ANGUISH = 19; // туга
private static final int EXALTATION = 20; // піднесення
private static final int PROBLEMS_APPETITE = 21; // втрата апетиту
private static final int AMNESIA = 22; // амнезія
private static final int PARALYSIS = 23; // параліч
private static final int FRISKINESS = 24; // грайливість, пустотливість
private static final int VOMITING = 25; // блювання

```

Усі варіанти симптом, що можливо статично добавляти використовуючи метод створення симптом, наведений нище.

```

private Symptom createSymptom(int id, String name, String question, String
description) {
    Symptom symptom = new Symptom();
    symptom.setId(id);
    symptom.setName(name);
    symptom.setQuestion(question);
    symptom.setDescription(description);
    return symptom.

```

При допомозі методу add.create(), список можна наповнити різними індексованими запитаннями та відповідними їм захворюваннями та закодованими системами симптом.

```

symptomList.add(createSymptom
(NEGATIVISM,
"Негативізм",
"Хворий робить протилежне тому, чого від нього вимагають?",
""))
);

```

Назва хвороби та перерахування особливих їй симптомів записуються при допомозі методу add.Disorder().

```

disorderList.add(createDisorder(5,
"Біполярний афективний розлад",
"Опис біполярного афективного розладу",
EUPHORIA, KINDLING, EXALTATION, ANGUISH)
);
disorderList.add(createDisorder(6,
"Циклотимія",
"Опис циклотимії",
KINDLING, EUPHORIA, EMOTIONS_CHANGING, EXALTATION,
ANGUISH)
);
disorderList.add(createDisorder(7,
"Синдром Корсакова",
"Опис синдрому Корсакова",
AMNESIA, PROBLEMS_ORIENTATION_TIME, DISORIENTATION,
PARALYSIS, FIXATION_AMNESIA,
KRYPTOMNESIA)
);

```

### 3.6 Проектування додатку

У процесі розробки програми за базовий клас діяльності приймається `activity`, який окреслює дії, що зможе здійснювати користувач, і візуально показує активність програми. При цьому, перш за все, утворюється об'єкт, після чого проходить запуск програми, який після тестування ліквідується, що дає можливість користувачу додатком перейти до наступного об'єкту. Приклад алгоритму створення об'єкту показаний на рисунку 3.8.

Метод `onCreate()` підтягується середовищем, після побудови нової функції `activity`. Зазначений метод ініціює початкову конфігурацію нової функціональності, з утворенням візуального інтерфейсу для `Windows` і зазначає його границі, кінцевим результатом якого є `Bundle`.

Об'єкт Bundle, що мав певний номер надає йому позицію порожнього значення, на відмінну від activity, що утворюється, як новий об'єкт. Якщо activity створювалась спочатку та знаходилась у приторможеному стану то Bundle вміщує основні дані про Activity.

Метод OnStart() викликається, лише після переходу у обраний режим activity. Після чого проходить OnResume() функції та надає можливість користувачу з ним співпрацювати. Якщо користувач приймає рішення про перехід до слідуючого Activity, системою активізується функція onPause().

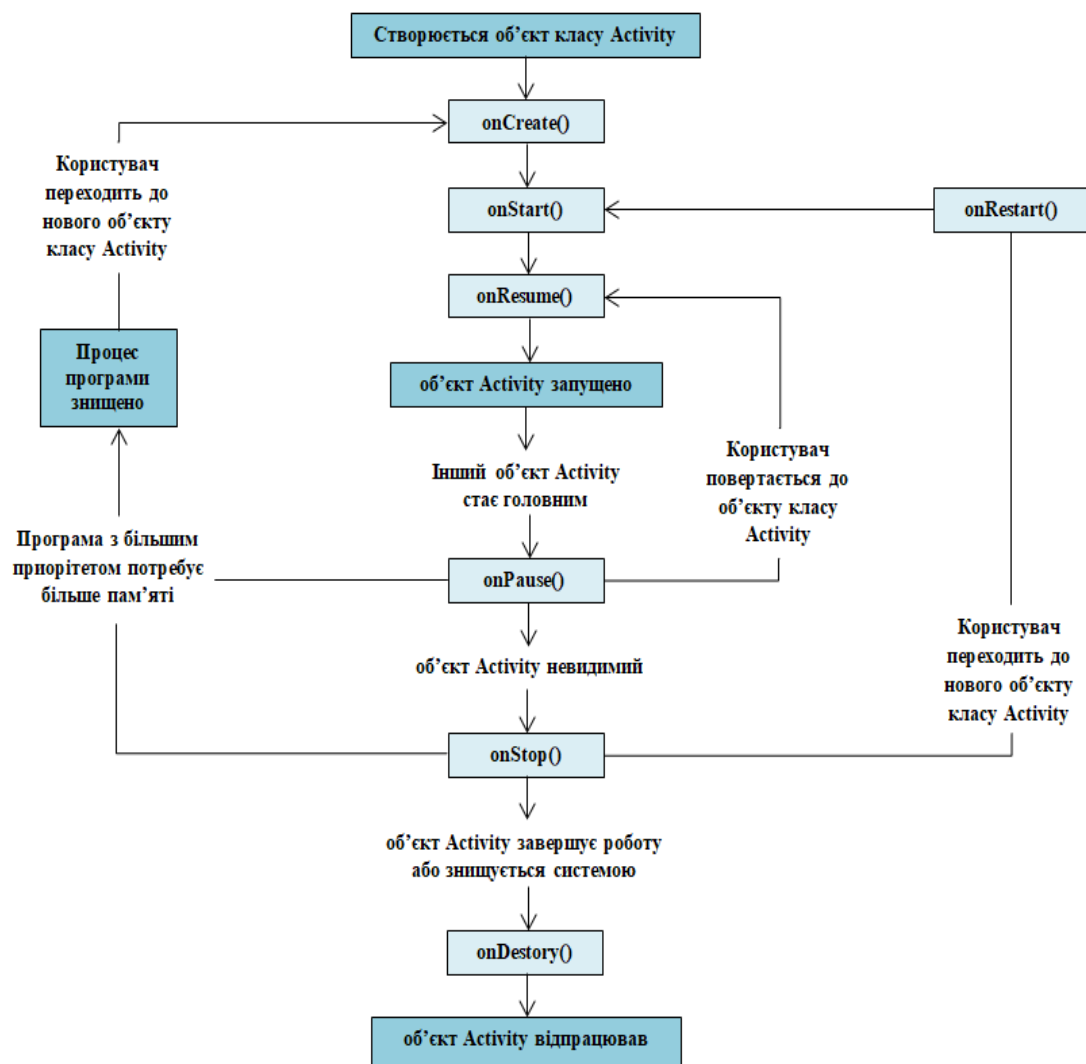


Рисунок 3.8 — Приклад алгоритму створення Activity

Якщо подія не з'являється на екрані, тобто її відключено, використовується метод onStop(). Якщо користувач має бажання повертатися до вихідної дії, після onStop(), тоді викликається системою метод onRestart().

Подія закінчується активуванням функції `OnDestroy()`, при умові коли додаток прийме рішення ліквідувати активність при виконанні методу `Complete()` [48].

Після запуску розробленого додатка до уваги користувача представляється головне вікно `MainActivity`, у якому необхідно обрати конкретний режим роботи з цим програмним забезпеченням.

Фрагмент коду дії `MainActivity` виглядає так:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    setContentView(R.layout.activity_main);super.onCreate(savedInstanceState;
    Button buttonStart = (Button) findViewById(R.id.buttonStart);
    Button button = (Button) findViewById(R.id.button);
    buttonStart.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intentStartDiagnostics = new Intent(MainActivity.this,
DiagnosticsActivity.class);
            startActivity(intentStartDiagnostics); }
        });
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intentStat = new Intent(MainActivity.this, TestActivity.class);
            startActivity(intentStat);
            public void onClick(View view) {
                Intent intentStat = new Intent(MainActivity.this, TestActivity.class);
                startActivity(intentStat);
            }; }
    }); }
```

Процес реалізації інтерфейсу користувача, взаємозв'язок з логічною послідовністю програмного продукту, забезпечують такі класи як

TestActivity.java і DiagnostictsActivity.java. Тим самим додаток реагує відкриттям тестів Люшера, Роршарха чи запитальник за вибором користувача.

Повний список коду представлений лістингом у Додатку Ж.

### 3.7 Налаштування розмітки користувацького інтерфейсу

У процесі створення програмного додатку для Android окрема частина візуальних полей виконується окремо. Форма макета дисплею розташована у файлі .xml папки проекту, що має назву «layout».

Ієрархія об'єктів View і ViewGroup застосовується для тегів додатків Android [49]. Зразки об'єктів — це утиліти користувацького інтерфейсу, що можуть бути представлені кнопками, символами чи текстовими формами. Існує декілька типових штамів тегів: GridLayout, LinearLayout, TableLayout, FrameLayout тощо.

GridLayout — різновид тегів розмітки, який на перший погляд може виглядати як TableLayout. Дійсно це дуже зручно. Позначення відноситься до android.widget.GridLayout, який складається зі стовпців, рядків, одиниць, так як і в TableLauot, де теги можна легко налаштувати. У GridLayout можливо вказати рядки та колонки для кожного розділу, і тут буде створена таблиця. Немає необхідності розкладати кожну клітинку за кожною клітинкою, потрібно це роботи тільки для тих клітин, де вони є актуальними. Розділи можна розділити на сегменти у таблиці, де в одну клітинку можливо вмістити декілька папок. LinearLayout має більш ніж один елемент. Як впливає з назви — це лінійний символ, де елементи вміщуються послідовно, та можуть мати вертикальне чи горизонтальне положення. У вертикальному положенні обраний тег займає окремий рядок, а в горизонтальному положенні — розташовуються в ряд, висота якого є максимально віддаленою для елемента нащадка.

TableLayout — відтворює похідні розділи у стрічках так колонках, так як в минулому вони застосовувалися. Цей вид розмітки не показує обмежень для своїх рядків, колонок або клітинок. Форма таблиці має стрічки з неоднаковими номерами комірок [49]. FrameLayout — простий вид розмітки, який вміщує



тільки одну похідну групу дочірніх елементів, наприклад View чи ViewGroup, організовану у верхньому лівому куті. Коли ви додаєте такі функції, вони зливаються.

Під час створення таблиць окремі клітинки можуть залишатися порожніми, коли це потрібно. При тегуванні стрічок зачасту застосовуються об'єкти такі як TableRow, що дочірніми елементами групи TableLayout . У ряду не має бути наявно багато клітинок, що являються хранилищами даних для інших. Також можна додати ще одну комірку LinearLayout чи TableLayout. Теги реалізуються у розмітці R.layout.main шляхом реалізації програмного коду. Фрагмент кодової розмітки зображена далі:

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="15sp"
    android:layout_alignParentStart="true"
    android:text="Будь ласка, оберіть необхідний функціонал:"
    android:layout_margin="20dp"/>
<Button
    android:id="@+id/buttonStart"
    android:layout_width="170dp"

```

```

    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="Діагностування">
</Button>
<Button
    android:id="@+id/button"
    android:layout_width="162dp"
    android:layout_height="wrap_content"
    android:text="Довідник"
    android:layout_margin="2dp">
</Button>

```

Користувацький інтерфейс, опублікований у XML файлах, дозволяє розділити формат програми на код, використовуючи процес форматування. Для екранів з різною роздільною здатністю, пристроїв різних неоднакових розмірів, мов додаток в змозі визначати розмітку у XML файлах, яка дозволить полегшити перегляд і створення текстового формату. Фрагмент коду розмітки поданий в Додатку І.

### 3.8 Опис розробленого програмного продукту

Створена інформаційно-аналітична система визначення психічного стану людини називається тест «Psyho», що являє собою набір тестувань для психодіагностики. Це програмне забезпечення є безкоштовним файлом з розширенням APK, розроблене у Android Studio та призначено для мобільних пристроїв. Існує три режими для здійснення видів психологічного діагностування. При обрані першого — користувач буде мати змогу пройти діагностування шляхом традиційного опитування. Обмеженням зазначеного режиму діагностики є те, що відхилення, яке не внесено у базу даних не можливо продіагностувати. Додаток до програми оснащений «Довідником» у якому перераховано усі додані симптоми та захворювання, що їм відповідають.

При обранні другого режиму — користувач буде мати змогу пройти діагностування особистості та діагностування психофізіологічного стану шляхом обрання тесту «Люшера», який полягає в обранні карточки асоціативних кольорів в даний час.

При обранні третього режиму — користувач буде мати змогу пройти діагностування за допомогою тесту «Роршарха». Тест «Psyho» спроектований у середовищі Android Studio за допомогою Java, яка використовували методи ООП. Деревоподібна ієрархія додатку, включає в себе всі необхідні конфігураційні файли, та дані спрощеної БД ( рисунок 3.9), а саме:

- application(app) — головна структурна папка додатку;
- manifestes — папка з конфігураційним файлом AndroidManifest.xml;
- java — olexandr.mentalist, data, model, source, ui, utils, папки що містять посилання на код;
- res (drawable, layout) — папки з xml тегами.

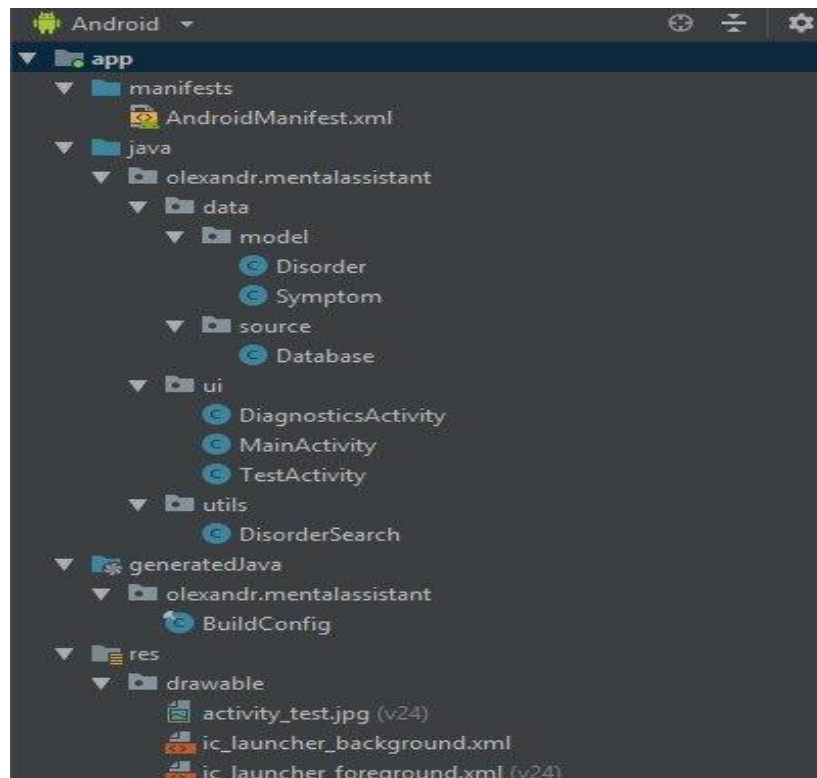


Рисунок 3.9 — Основні конфігураційні файли додатку

«Database.java» — клас, який вміщує у собі БД із переліком симптомів та відповідного психічного захворювання, що його описують. «Symptom.java», «Disorder.java» — це класи, які повинні взаємо зв'язати симптоми або їх

колекції з відповідними їм психічними станами. «TestActivity.java», «MainActivity.java», «DiagnosticsActivity.java» — класи, на які покладено функцію реалізувати користувацький інтерфейс, а також їх взаємозв'язок з конфігураційними алгоритмами.

«DisorderSearch.java» — клас, який повинен реалізувати алгоритм знаходження даних та їх фільтрацію. Під час запуску мобільного додатку обраного режиму користувачу виводиться на екран його головне меню. Обрано режим тесту «Роршаха» (рисунок 3.10).

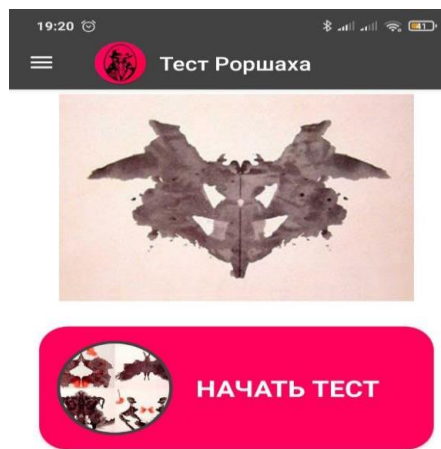


Рисунок 3.10 — Меню тесту «Роршарха»

Додаток відображає в послідовності десять різних зображень, серед яких необхідно обрати те, яке найбільше відповідає відчуттю користувача за значенням, кольором чи формою. По завершенні кожного етапу обрані користувачем відповіді архівуються. Після проходження пацієнтом останнього етапу усі архівовані відомості систематизуються, аналізуються та об'єднуються відповідно до заданих перерахованих шаблонів. Результатом проходження цього тесту є відображення кінцевих результатів, виведених на екран.

Обрано тест «Pshyho» (рисунок 3.11) у якому користувач може виявити психічне захворювання за відповідністю його наявних симптомів. Цей процес реалізується шляхом збору відповідей при допомозі пошукового алгоритму, який ґрунтується на аналогії притаманних симптомів певній хворобі після архівації відповідей у додатку.

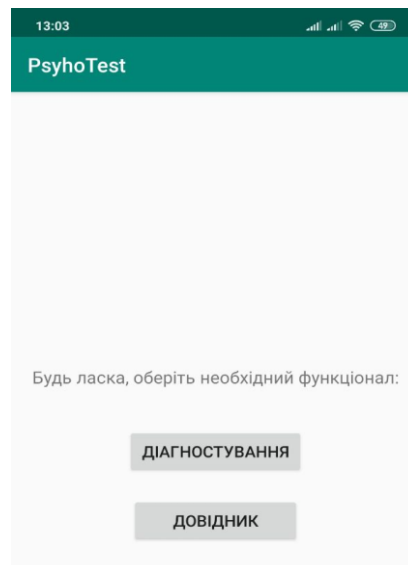


Рисунок 3.11 — Меню тесту «Psyho»

Додаток генерує 3 масиви базових даних. Перший масив, вміщує всі відомості про захворювання та їх симптоматику. Другий масив (проміжний), вміщує тимчасові відомості. Третій масив, містить відфільтровані відомості, тобто ті, що інформують користувача про отримання збігів при діагностиці окремих психічних відхилень.

При тестуванні «Psyho», на дисплеї з'являються питання стосовно притаманності психічного симптому особи, що досліджується, наприклад, «У хворого бувають марення?». Тестуючому висвічуються 3 можливих відповіді — «так», «невідомо» та «ні» (рисунок 3.12).

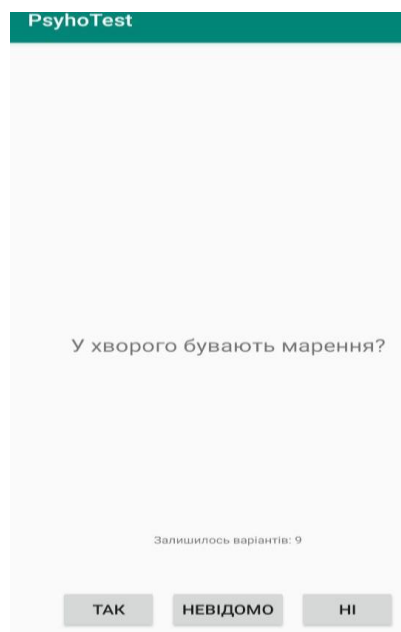


Рисунок 3.12 — Вікно вибору відповіді

При умові обрання варіанту «так», психічна хвороба, яка мала симптом с попереднього запитання, переміщується до бази проміжних даних, у якій зберігається до моменту остаточної перевірки. При умові обрання варіанту «невідомо», хвороба та симптом який їй відповідає не переміщується до жодної з баз проміжних даних та не приймаються до уваги програмою для подальших відповідностей. При умові обрання варіанту «ні», інформація про відсутність наявного симптому відмічається певною міткою у відповідному захворюванні. Хоча у будь якому випадку попадає у базу проміжних значень, де залишається до моменту остаточної перевірки, оскільки одна відповідь «ні» не дає гарантію остаточного діагнозу.

Раціональне фільтрування необхідних даних та значне зменшення переліку запитань досягається тим, що додаток проводить аналіз кожної обраної відповіді на питання, врахувавши притаманність чи відсутність симптомів, які зазначив пацієнт. Тривалість опитування сягає того моменту до поки буде визначено психічне відхилення чи до поки не вичерпаються запити у БД. Як результат, додаток виводить сповіщення, на екран мобільного пристрою, про вірогідне психічне захворювання. Відомості про симптоматику та психічний розлад можна у програмному додатку «Довідник» (рисунок 3.13).



Рисунок 3.13 — Програмний додаток «Довідник»

Обрано режим тесту «Люшера» (рисунок 3.14), де додаток відображає послідовно набори кольорових карток, у кожному з яких задається певна кількість різнокольорових прямокутників. У кожному такому запропонованому наборі, користувачу потрібно обрати методом відсіювання, один з кольорів, який подобається користувачу найбільше в даний момент часу. Характерним є те, що можливе повернення користувача на початковий етап тестування для підтвердження достовірності проходження. По завершенню на екрані смартфона з'являється опис характеристик: загальної настрою, самооцінки, збудження, психологічні та інші. Як підсумок виводиться актуальна емоційна проблема та показник стану тривожності.



Рисунок 3.14 — Меню тесту «Люшера»

Програмне забезпечення відкривається стандартним чином, запускаючи вибрану програму для Android, тобто один раз натиснувши на піктограму. При діагностуванні тесту «Psyho» кнопки «так», «невідомо», «ні» є індикаторами вхідних значень, та інтерпретуються у програму значеннями «1» та «0» для визначення діагнозу. У тестах «Роршаха» та «Люшера» це об'єкти, які обробляються функцією `isChecked()`, яка конвертує вибрані зображення у програмні параметри для отримання кінцевих графічних результатів, які виводяться на екран у вигляді текстових повідомлень.

## 4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ВИЗНАЧЕННЯ ПСИХІЧНОГО СТАНУ ЛЮДИНИ

Виконання дослідницької роботи передбачає певні витрати. Зазвичай це витрати на створення та впровадження конкретного програмного забезпечення. Такі витрати необхідно постійно зменшувати, адже у цьому полягає прогрес будь-якого суспільства. Без такого прогресу ніяка науково-технічна розробка не буде реалізована на практиці, адже така розробка не буде ефективнішою за існуючі на ринку аналоги.

На основі економічних розрахунків можна довести економічну доцільність та ефективність впровадження отриманих результатів виконаних науково-дослідних робіт у виробництво, тобто здійснити комерціалізацію наукових розробок [50]. Дана магістерська кваліфікаційна робота носить експериментальний характер, тобто прогнозується впровадження науково-технічної розробки безпосередньо розробником на підприємстві.

### 4.1 Проведення технологічного аудиту науково-технічної розробки

Метою оцінки потенціалу комерційного розвитку є оцінка потенціалу комерційного розвитку, що впливає з науково-технічних досліджень. За результатами оцінки робляться висновки про напрямки (особливості) організації в майбутньому її впровадження з урахуванням встановленої оцінки. У цілях проведення технологічного аудиту було залучено 3 незалежних експерти. Такими експертами є к.т.н., доц. Колесник Ірина Сергіївна, к.т.н., доц. Черняк Олександр Іванович та к.т.н., доц. Крупельницький Леонід Віталійович. Комерційний потенціал інвестицій буде оцінюватись відповідно до дванадцяти критеріїв, наведених у таблиці 4.1.

Таблиця 4.1 — Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-бальною шкалою)					
Критерій	0	1	2	3	4



Продовження таблиці 4.1

Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Багато аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
Практична здійсненність					
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Закінчення таблиці 4.1

8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні.	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідно регламентні документи та велика кількість дозвільних документів на виробництво продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

На основі таблиці різні експерти, у нашому випадку керівник магістерської роботи та викладачі кафедри ОТ, визначають різні результати. Результати цієї оцінки комерційного потенціалу узагальнено у таблиці 4.2.

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Експерт (ПІБ, посада)		
	1 Колесник І.С., к.т.н., доц. кафедри ОТ	2 Черняк О.І., к.т.н., доц. кафедри ОТ	3 Крупельницький Л.В., к.т.н., доц. кафедри ОТ
	Бали:		
1. Технічна здійсненність концепції	3	3	2
2. Ринкові переваги (наявність аналогів)	4	3	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	3	3	4
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	4	3	3
9. Практична здійсненність (наявність фінансів)	2	2	2
10. Практична здійсненність (необхідність нових матеріалів)	4	3	4
11. Практична здійсненність (термін реалізації)	4	4	3
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	СБ <sub>1</sub> = 39	СБ <sub>1</sub> = 37	СБ <sub>1</sub> = 37
Середньо-арифметична сума балів СБ <sub>с</sub>	$СБ_с = \frac{\sum_1^3 СБ_i}{3} = \frac{35 + 38 + 36}{3} = 38$		

Відповідно до таблиці 4.2, а також відповідно до рекомендацій, наведених у таблиці 4.3, можна зробити висновок про рівень потенціалу комерційного розвитку.

Таблиця 4.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 — 10	Низький
11 — 20	Нижче середнього
21 — 30	Середній
31 — 40	Вище середнього
41 — 48	Високий

З урахуванням середніх арифметичних балів  $СБ_c = 38$ , які були визначені експертами, можна зробити висновок, що рівень комерційного потенціалу цієї розробки буде вище середнього. На ринку дуже мало якісних програм аналогів для діагностування психічного стану.

Програмами аналогами є Psychology Today та Домашній психолог. Веб-застосунок, Psychology Today, де зібрали групи відомих психологів, науковців, психіатрів та письменників, щоб донести свої думки та ідеї. Присутнє проходження заздалегідь підготовлених психологічних опитувань та генерування висновку на основі пройденого раніше.

Також є можливість знайти собі спеціаліста в психологічному напрямку для подальшого лікування. Це рішення уже більш професійне в плані психології та психіатрії, тут присутні онлайн консультації у відомих психологів та науковців, також в наявності опитувальники, які націлені на психоаналіз людини та генерування висновку на результаті цього, також більш зручний та зрозумілий інтерфейс.

По-перше, один із головних мінусів застосунку, те що тести не можна створювати та редагувати, всі створенні опитувальники заздалегідь підготовленні адміністраторами. По-друге, варіативність питань мала, маємо тільки один вид питань. По-третє, немає змоги прослідкувати динаміку змін стану пацієнта. Якщо підсумувати все сказане вище цей застосунок не відповідає деяким вимогам — не маємо прив'язки пацієнт — терапевт, недостатній функціонал для проведення аналізу пацієнтів.

Має вартість 2300.

Домашній психолог. Це «продукт», який може застосовуватися для коригування психічного стану пацієнта для підвищення результату загальної терапії.

Позитивний характер впливу «Програми» виражаються в наступному:

- позбавлення від залежності,
- підвищення продуктивності розумової діяльності;
- поліпшення пам'яті і уваги;
- стабілізація емоційного стану;
- стабілізація настрою;
- загальне тонізуючу дію на стан здоров'я;
- зростання фізичної та сексуальної активності;
- повернення інтересу до життя.

На сучасному етапі інформатизації суспільства майже ніхто не обходиться без комп'ютера, смартфона, планшета, а це і є носії лікувальних програм, які зможуть допомагати людині в її повсякденному житті. Основним завданням нового ПЗ є швидка, дієва психологічна допомога і при цьому коректна робота на сучасних поширених версіях операційних систем Windows, Android, IOS. А однією з задач програмування є необхідність оновлювати софт до сучасного рівня і додавати необхідні функції. Основне призначення нашого програмного продукту — позитивний вплив на мозок через органи зору і слуху з метою коригування психіки людини.

Недоліки:

- дорога вартість (100 доларів за сеанс);
- складний інтерфейс для звичайного користувача;
- неможливість повноцінно купівлі програми;
- вузький спектр запитань.

Таблиця 4.4 — Порівняння характеристик розробки із аналогом

Показники	Розробка	Аналог1	Аналог2
Функціонал	7	7	8

Закінчення таблиці 4.4

Швидкодія	8	8	7
Надійність	8	8	8
Метод розповсюдження	9	7	6
Інтерфейс, простота використання	8	7	7
Ціна	—	2300 грн	2700 сеанс (немає можливості повноцінної купівлі)

Програма, яка розробляється, буде виконувати такі функції:

- проведення тестування для визначення психічного захворювання;
- аналізувати психічний стан хворого;
- швидко опрацьовувати данні;
- буде мати функціонал для генерування висновку;
- виведення статистичних даних для розробника.
- буде підтримуватися сучасними ОС;
- буде мати легкий та зручний для користувача графічний інтерфейс.

Продукт буде просуватися за допомогою реклами в соціальних мережах, пошукових системах та багатьох інших джерелах Інтернету. Використовуючи аналітику цих сервісів, можна буде націлити рекламу на цільову групу захисників інформації.

Наукова новизна дослідження полягає в тому, що вперше буде запропоновано та впроваджено комплексне використання методів психологічного дослідження для отримання достовірної оцінки психіатричного стану людини.

Даний рівень було досягнуто за рахунок покращення та/або розширення функціональних можливостей нової науково-технічної розробки порівняно з аналогічними розробками, існуючими в цей час на ринку.

## 4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Позитивний вплив на мозок через органи зору і слуху з метою коригування психіки людини. У магістерській роботі розглядається програмний продукт, який через органи зору і слуху коригує психічний стан людини, тому значну частину витрат складають витрати пов'язані з розробкою, а не виробництвом та відтворенням [51]. Відповідно, є певна специфіка розрахунків.

### 4.2.1 Витрати на оплату праці

Основна заробітна плата розробників, що працюють над проектом, визначена у формулі:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де  $k$  — кількість посад дослідників, залучених до процесу досліджень;

$M_{ni}$  — місячний посадовий оклад конкретного дослідника, грн;

$T_p$  — середня кількість робочих днів в місяці,  $T_p=20\dots23$  дні, обрано 20 днів;

$t_i$  — кількість днів роботи конкретного дослідника, дн.;

Над створенням розробки працював менеджер проекту та інженер програмного забезпечення, тому ми виконаємо для них усі необхідні розрахунки, і вносимо їх до таблиці 4.5:

$$Z_{o.k.} = \frac{14500 \cdot 8}{20} = 6000(\text{грн}).$$

$$Z_{o.v.} = \frac{10000 \cdot 32}{20} = 21800 (\text{грн}).$$

Додаткова винагорода ( $Z_{\text{дод.}}$ ) усіх розробників та працівників, які брали участь у цьому етапі роботи, обчислюється як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод.}} = (Z_o + Z_p) \cdot \frac{N_{\text{дод.}}}{100\%}, \quad (4.2)$$

де  $N_{\text{дод.}}$  — норма нарахування додаткової заробітної плати.

$$Z_{\text{дод.к.}} = \frac{12 \cdot 5800}{100} = 696 \text{ (грн)},$$

$$Z_{\text{дод.в.}} = \frac{12 \cdot 16000}{100} = 1920 \text{ (грн)},$$

$$Z_{\text{дод.}} = Z_{\text{дод.к.}} + Z_{\text{дод.в.}} = 2616 \text{ (грн)}.$$

Витрати на основну заробітну плату робітників за відповідними найменуваннями робіт відсутні, тобто  $Z_p = 0$ .

Таблиця 4.5 — Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	14500	725	9	5800
Старший інженер-програміст	10000	500	32	16000
Всього				21800

#### 4.2.2 Соціальні відрахування

Заробітна плата робітників відсутня, тому  $Z_p = 0$ . Нарахування на заробітну плату дослідників та нарахування на заробітну плату працівників, які брали участь у цьому етапі роботи, розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод.}}) \cdot \frac{N_{\text{зп}}}{100\%} \quad (4.3)$$

де  $N_{\text{зп}}$  — норма нарахування на заробітну плату.



$$Z_H = (21800 + 0 + 2616) \cdot \frac{22\%}{100\%} = 5371,52 \text{ (грн.)}$$

#### 4.2.3 Сировина та матеріали

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{Bj}, \quad (4.4)$$

де  $H_j$  — кількість матеріалу  $j$ -го виду, шт.;

$n$  — кількість видів матеріалу.

$C_j$  — ціна матеріалу  $j$ -го виду, грн;

$K_j$  — коефіцієнт транспортних витрат,  $K_j = (1,1 \dots 1,15)$ , обираємо  $K_j$  1,15;

$B_j$  — маса відходів  $j$ -го найменування, кг;

$C_{Bj}$  — вартість відходів  $j$ -го найменування, грн/кг.

Результати розрахунків занесено до таблиці 4.6.

Таблиця 4.6 — Витрати на матеріали

Найменування комплектуючих	Ціна за 1 штуку, Грн.	Кількість матеріалу, штук	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Ручка	25,00	1	0,05	0,09	24,91
Карта пам'яті	550,00	1	0	0,00	550,00
Пачка офісного папіру	105,00	1	0,5	18,75	86,25
Всього (з урахуванням транспортних витрат)					727,28

#### 4.2.4 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.5)$$

Таблиця 4.7 — Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців.	Амортизаційні відрахування, грн
ЕОМ	15000	2	1,5	937,5
Приміщення	100000	20	1,5	625
Всього				1562,5

#### 4.2.5 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховують за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{\text{впі}}}{\eta_i}, \quad (4.6)$$

де  $W_{yi}$  — встановлена потужність обладнання на певному етапі розробки, кВт;

$t_i$  — тривалість роботи обладнання на етапі дослідження, год;

$C_e$  — вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії),  $C_e = 4,62$  [52];

$K_{\text{впі}}$  — коефіцієнт, що враховує використання потужності,  $K_{\text{впі}} < 1$ ; обираємо  $K_{\text{впі}} 0,7$ ;

$\eta_i$  — коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = \sum_{i=1}^1 \frac{0,07 \cdot 256 \cdot 4,62 \cdot 0,7}{0,8} = 72,44 \text{ (грн.)},$$

Проведені розрахунки необхідно звести до таблиці 4.8.

Таблиця 4.8 — Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
ЕОМ	0,07	256	72,44
Всього			72,44

## 4.2.6 Накладні (загальновиробничі) витрати та інші витрати

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (z_{\text{о}} + z_{\text{р}}) \cdot \frac{N_{\text{ів}}}{100\%} \quad (4.7)$$

де  $N_{\text{ів}}$  — норма нарахування за статтею «Інші витрати».

$$I_{\text{в.к.}} = 5800 \cdot \frac{50\%}{100\%} = 2900 \text{ (грн.)},$$

$$I_{\text{в.в.}} = 16000 \cdot \frac{50\%}{100\%} = 8000 \text{ (грн.)},$$

$$I_{\text{в}} = I_{\text{в.к.}} + I_{\text{в.в.}} = 10900 \text{ (грн.)}.$$

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{нзв}} = (z_{\text{о}} + z_{\text{р}}) \cdot \frac{N_{\text{нзв}}}{100\%} \quad (4.8)$$

де  $N_{\text{нзв}}$  — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Беремо норму нарахування 150%.

$$V_{\text{нзв.к.}} = 5800 \cdot \frac{150\%}{100\%} = 8700 \text{ (грн.)},$$

$$V_{\text{нзв.в.}} = 16000 \cdot \frac{150\%}{100\%} = 24000 \text{ (грн.)},$$

$$V_{\text{нзв}} = V_{\text{нзв.к.}} + V_{\text{нзв.в.}} = 32700 \text{ (грн.)}.$$

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_0 + Z_{\text{дод}} + Z_{\text{н}} + M + K_{\text{в}} + B_{\text{спец}} + B_{\text{прг}} + A_{\text{обл}} + B_{\text{е}} + B_{\text{св}} + B_{\text{сп}} + I_{\text{в}} + B_{\text{нзв}} \quad (4.9)$$

У нашому випадку:

$Z_{\text{р}} = 0, K_{\text{в}} = 0, B_{\text{спец}} = 0, B_{\text{прг}} = 0, B_{\text{св}} = 0, B_{\text{сп}} = 0$ , тому отримаємо:

$$\begin{aligned} B_{\text{заг}} &= Z_0 + Z_{\text{дод}} + Z_{\text{н}} + M + A_{\text{обл}} + B_{\text{е}} + I_{\text{в}} + B_{\text{нзв}} \\ &= 21800 + 2616 + 5371,52 + 727,28 + 1562,5 + 72,44 + 10900 + 32700 \\ &= 75749,74 \text{ (грн)}. \end{aligned}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta}, \quad (4.10)$$

де  $\eta$  — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, обираємо його 0,5.

$$ЗВ = \frac{75749,74}{0,5} = 151499,48 \text{ (грн)}.$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її впровадження безпосередньо розробником (замовником)

Для оцінки та прогнозування зростання чистого прибутку від впровадження розробником нової науково-технічної продукції на своєму підприємстві (організації) використаємо метод прямої оцінки зростання чистого прибутку підприємства від впровадження результатів наукової розробки [53]. У цьому випадку збільшення чистого прибутку підприємства  $\Delta \Pi$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою 4.11:

$$\Delta \Pi_i = (\Delta \Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i, \quad (4.11)$$

де  $\Delta \Pi_{\text{я}}$  — покращення основного якісного показника від впровадження результатів розробки у даному році;

$N$  — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$P_{я}$  — основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо що в результаті впровадження наукових досліджень за допомогою реклами кількість учасників товариства зростає, що дозволяє проводити більшу кількість учасників проведення тестування за рік з 100 до 150, за другий рік з 150 до 300, за третій рік, з 30 до 40, а плату за організацію підвищити на 200 грн.

Орієнтовно — поповнення бюджету розробника (замовника) до впровадження проекту складало близько 15000 грн, кількість учасників проведення тестування на рік на 100.

Тоді збільшення чистого прибутку товариства  $\Delta P_i$  протягом першого року складе:

$$\Delta P_1 = 200 * 100 + (1500 + 200) * 50 = 105000$$

Збільшення чистого прибутку товариства  $\Delta P_i$  протягом другого року (відносно базового року, тобто року до впровадження результатів наукового дослідження) складе:

$$\Delta P_2 = 200 * 100 + (1500 + 200) * (50 + 150) = 360000$$

Збільшення чистого прибутку товариства  $\Delta P_i$  протягом третього року (відносно базового року, тобто року до впровадження результатів наукового дослідження) складе:

$$\Delta P_2 = 200 * 100 + (1500 + 200) * (50 + 150 + 100) = 530000$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків  $III$ , що їх може отримати розробник (замовник) від можливого впровадження науково-технічної розробки на власному підприємстві:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.12)$$

де  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ , обираємо  $\tau 0,1$ ;

$t$  — період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{105000}{(1 + 0,1)^2} + \frac{360000}{(1 + 0,1)^3} + \frac{530000}{(1 + 0,1)^4} = 719247,32 \text{ (грн)}.$$

Далі розраховують величину початкових інвестицій  $PV$ , які розробник (замовник) має вкласти для здійснення науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ, \quad (4.13)$$

де  $k_{\text{інв}}$  — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, які можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{\text{інв}}=2 \dots 5$ , але може бути і більшим. Обираємо даний коефіцієнт 2;

$ЗВ$  — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 \cdot 151499,48 = 302998,96 \text{ (грн.)}$$

Тоді абсолютний економічний ефект  $E_{абс}$  або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.14)$$

де ПП — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV — теперішня вартість початкових інвестицій, грн.

$$E_{абс} = 719247,32 - 302998,96 = 416248,17 \text{ (грн.)}$$

Оскільки  $E_{абс} > 0$ , то результат від проведення наукових досліджень та їх впровадження принесе прибуток.

Внутрішня економічна дохідність інвестицій  $E_B$ , які можуть бути вкладені розробником (замовником) у впровадження науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{ж}] \left( 1 + \frac{E_{абс}}{PV} \right) - 1, \quad (4.15)$$

де  $E_{абс}$  — абсолютний економічний ефект вкладених інвестицій, грн;

PV — теперішня вартість початкових інвестицій, грн;

$T_{ж}$  — життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$E_B = \sqrt[3] \left( 1 + \frac{416248,17}{302998,96} \right) - 1 = 0,35$$

Далі розраховуємо період окупності інвестицій  $T_{ок}$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_B}, \quad (4.17)$$

де  $E_v$  — внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = \frac{1}{0,35} = 2,8 \text{ року}$$

Оскільки  $T_{ок} = 2,8$  року, тоді розвиток доречний.



## ВИСНОВКИ

У магістерській кваліфікаційній роботі розроблено інформаційно-аналітичну систему визначення психічного стану за допомогою якої можливе виявлення психічного, емоційного та психофізіологічного розладу. Така система може бути використана також для діагностування психічних відхилень на початкових стадіях захворюваності, що сприяє збільшенню кількості успішного лікування.

У першому розділі магістерської роботи було проведено аналіз предметної області для створення системи. Наведено загальні положення про психіатрію, як науку. Проведено аналіз емоційних станів людини, моделей та методів оцінки психічного стану. Були розглянуті основні аспекти психологічного та комп'ютерного тестування. Проаналізовано методи традиційної психіатрії, методики тестів «Роршарха» та «Люшера».

У другому розділі магістерської роботи проаналізовано методи та основні алгоритми для розробки програмного продукту. Обрано інкременту модель алгоритму та змодельовано деревоподібну структуру програмного додатку. Описано класифікацію ключового функціоналу додатку та побудовано контекстну діаграму інформаційної системи діагностування психічного стану людини. Також основну увагу приділено принципам розробки та структури Android-додатку.

У третьому розділі магістерської роботи було обрано необхідний інструментарій для розробки продукту, зокрема мову програмування Java та середовище розробки Android Studio. Також було проведено порівняння середовищ Android Studio та Eclipse, де були виділені основні переваги Android Studio. Розглянута початкова конфігурації додатку, де описано розмітку xml файлу та його файлова структура. Проведено розробку та аналіз структури бази даних та наведено модель роботи мобільного додатку. Здійснено проектування інформаційно-аналітичної системи визначення психічного стану людини, де детально описано налаштування розмітки

графічного інтерфейсу. Продемонстровано 3 режими для здійснення психологічного діагностування.

У четвертому розділі магістерської її роботи було виконано розрахунок економічної доцільності створення інформаційно-аналітичної системи визначення психічного стану людини, проведено технологічний аудит для науково-технічної розробки, здійснено розрахунок потрібних економічних затрат (на оплату праці, відрахування на заробітню плату, сировину та матеріали амортизацію обладнання програмних засобів, на електроенергію), що необхідні для реалізації системи. Також проведено порівняння з аналогами та визначення економічної ефективності розробки за її впровадження безпосередньо розробником (замовником).

Розроблений програмний додаток є безкоштовним та надійним, що робить його конкурентоспроможним за рахунок:

- ціни та якості;
- надійності;
- швидкодії;
- інтегрованості;
- простоті інтерфейсу.

Інформаційно-аналітична система визначення психічного стану людини у повній мірі придатна до виконання і може бути використана у медицині, з метою діагностування, лікування та реабілітації психічних, емоційних та психофізіологічних захворювань. Для практичного застосування додатку були наведені всі необхідні інструкції. Додаток, також може бути вдосконалений при потребі. Результати розробки опубліковані у Міжнародному науково-технічному журналі «Інформаційні технології та комп'ютерна інженерія» у вигляді наукової статті.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Психология развития от рождения до смерти. Полный курс психологии развития / под редакцией члена-корреспондента РАО А.А. Реана. — СПб: «Прайм-ЕВРОЗНАК», 2005. — 416 с.
2. Рубинштейн С. Л. Человек и мир / С. Л. Рубинштейн // Бытие и сознание. Человек и мир. — СПб.: Питер, 2003. — С.282–427
3. Осипова А. А. Общая психокоррекция: учебное пособие для студентов вузов / А. А. Осипова. — М: ТЦ «Сфера», 2000. — 512 с.
4. Нартова-Бочавер С. К. Дифференциальная психология / С. К. НартоваБочавер. — М.: Флинта, НОУ ВПО Московский психолого-социальный университет, 2015. — 341 с.
5. Lidské emoce — Milan Nakonečný [Электронный ресурс]. Режим доступа: <https://www.kosmas.cz/knihy/63146/lidske-emoce/>
6. Computing emotion awareness through facial electromyography. Computer vision in human-computer interaction [Электронный ресурс]. Режим доступа: [https://link.springer.com/chapter/10.1007/11754336\\_6](https://link.springer.com/chapter/10.1007/11754336_6)
7. Eight dimensions for the emotions [Электронный ресурс]. Режим доступа: <http://journals.sagepub.com/doi/abs/10.1177/0539018409106198>
8. Multivariate discrimination of emotion-specific autonomic nervous system activity. [Электронный ресурс]. Режим доступа: <https://vtechworks.lib.vt.edu/bitstream/handle/10919/32903/Thesis1.pdf?sequence=1&isAllowed=y>
9. Е. В. Распопин. Методы изучения и оценки психических состояний [Электронный ресурс]. Режим доступа: <http://elar.urfu.ru/bitstream/10995/43809/1/iurp-2016-156-16.pdf> - 28.04.2016 р.
10. Класифікація психологічних тестів [Електронний ресурс]. Режим доступа: <https://studopedia.org/13-103872.html>
11. А. О. Прохоров. Практикум по психологии состояний: Учебное пособие. — СПб: Речь, 2004. — 480 с.

12. Основи психології: Підручник для студ. вузів./ За заг. ред. О.В.Киричука, В.А.Роменця. — 3-є вид., стереотип. — К.: Либідь, 1997. — 630 с.
13. Математичні методи: [Електронний ресурс]. Режим доступу: [http://studopedia.su/8\\_1156\\_matematichni-metodi.html](http://studopedia.su/8_1156_matematichni-metodi.html)
14. Предмет психології її завдання та методи: [Електронний ресурс]. Режим доступу: [http://vvpc.com.ua/files/biblioteka/psih\\_sait/lekcija%201.htm](http://vvpc.com.ua/files/biblioteka/psih_sait/lekcija%201.htm)
15. Психодіагностика/ І. М. Галян. — К.: Академвидав, 2009. — 464 с
16. Українська педагогіка в персоналіях: У 2 кн. Кн. 2.: Навч. посібн./ За ред. О.В. Сухомлинської. — К.: Либідь, 2005. — 424 с.
17. Основи психології і педагогіки: Навч. Посібник./ Степанов О.М., Фіцула М.М. — 3-є вид., доповнене. — К.: Академвидав, 2012. — 528 с.
18. Вікова та педагогічна психологія: Навч. посіб./ О.В. Скрипченко, Л.В. Долинська, З.В. Огороднійчука та ін. — К.: Просвіта, 2001. — 416 с.
19. Класифікація психологічних тестів: [Електронний ресурс]. Режим доступу: <http://helpiks.org/2-84926.html>
20. Психологія./ Цимбалюк І. М. — К.: Професіонал, 2004. — 215 с.
21. Психологія особистості: Навч. посіб./ Ліфарева Н. В. — К.: Центр навчальної літератури, 2003. — 240 с.
22. Формування нового обліку вітчизняної науки: Міжнародна конференція, м. Київ, 28 лютого 2014р. Центр наукових публікацій. — 212 с.
23. Відродження духовності в сучасному світі: взаємодія церкви і освіти: Тези Всеукраїнської науково-практичної конференції, 24-25 березня 2011 року — Х.: ХНУ імені В. Н. Каразіна, 2011. — 392 с.
24. Експериментальна психологія: Конспект лекцій./ Коновалова М.Д. — Спб.: 2012. — 25 с.
25. Конспект лекцій з дисципліни «Інформатика, основи програмування і застосування ЕОМ» для студентів усіх спеціальностей./ Укл: В.Д. Гогунський, К.В. Колеснікова, І.А. Ярова. — Одеса: Наука і техніка, 2004. — 60 с.

26. Педагогіка та психологія в системі професійної підготовки фахівців з вищою технічною освітою./ Герасимова І. Г. — Вісник ВПІ, 2003. — № 5. — 178 с.
27. Погляди Л.С. Виготського і проблеми сучасної психодіагностики: [Електронний ресурс]. Режим доступу: <http://bukvar.su/psihologija/87300-Vzglyady-L-S-Vygotskogo-i-problemy-sovremennoiy-psihodiagnostiki.html>
28. Метод тестування в психології [Текст]: Навч. Посібник./ В. В. Нікандров, В. В. Новочадов; Інститут біології і психології людини. — СПб.: Мова, 2003. — 41 с.
29. Теоретична та контролююча частина програмного забезпечення: [Електронний ресурс]. Режим доступу: [http://ito.vspu.net/ENK/PPZ\\_shahina/8.htm#2.1](http://ito.vspu.net/ENK/PPZ_shahina/8.htm#2.1)
30. Психология и педагогика: Учебное пособие./ Столяренко А. М. — М.: ЮНИТИ-ДАНА, 2004. — 423 с.
31. Основи проектування баз даних. Навчальний посібник./ Гайна Г. А. — К.: КНУБА, 2005. — 204 с.
32. Загальна психологія. Модульно-рейтинговий курс для студентів вищих навчальних закладів./ Цимбалюк І. М., Яницька О. Ю. — К.: Професіонал, 2004. — 304 с.
33. Загальна та прикладна психологія: Навчальний посібник./ Цигульська Т. Ф. — К: Наукова думка, 2000. — 192 с.
34. Бурлачук, Л.Ф. Психодіагностика: Учебник для вузов / Л.Ф. Бурлачук. — СПб.: Питер, 2006. — 351 с.
35. Щербатых Ю.В. Психология труда и кадрового менеджмента в схемах и таблицах : справочное пособие / Ю.В. Щербатых. — М.: КНОРУС, 2011. — 248 с.
36. Большой психологический словарь / сост. и общ. ред. Б. Мещеряков, В. Зинченко. — СПб. : Прайм-ЕВРОЗНАК. — 2003. — 672 с. — (Проект «Психологическая энциклопедия»).

37. Божович Л.И . Проблемы формирования личности/Под ред. Д. И. Фельдштейна — 3-е изд. — М.: Московский психолого-социальный институт, Воронеж: НПО «МОДК», 2001. — 352 с.
38. Ипатов Э.Р. Методология и технологии системного проектирования информационных систем: учебник / Э.Р. Ипатов, Ю.В. Ипатов. — М. : Флинта : МПСИ, 2008. — 256 с.
39. Галямина И.Г Управление процессами: Учебник для вузов. Стандарт третьего поколения. — СПб.: Питер, 2013. — 304 с.: ил. — (Серия «Учебник для вузов»)
40. Класифікація мобільних додатків [Електронний ресурс].Режим доступу: [http://voroninstudio.eu/uk/service/razrabotka\\_mobilnih\\_prilozheniy](http://voroninstudio.eu/uk/service/razrabotka_mobilnih_prilozheniy)
41. Мови програмування. [Електронний ресурс]. Режим доступу: [http://zei.narod.ru/Comparison\\_C\\_\\_Java\\_Cpp\\_3.pdf](http://zei.narod.ru/Comparison_C__Java_Cpp_3.pdf)
42. Android Studio. [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/Android\\_Studio](https://uk.wikipedia.org/wiki/Android_Studio)
43. Голощапов А. Л. Google Android: программирование для мобильных устройств — СПб.: БХВ-Петербург, 2011 — 448 с.
44. Архитектура Android-приложений [Электронне джерело]. Режим доступу: URL <http://habrahabr.ru/post/141201/>. — Назва з екрану.
45. Виды приложений и их структура [Электронне джерело]. Режим доступу: URL <http://www.intuit.ru/studies/courses/12643/1191/lecture/21983?page=2>. Назва з екрану.
46. Учебник по Android [Электронне джерело]. Режим доступу: URL <http://startandroid.ru/ru/>. Назва з екрану.
47. SQLite. [Электронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/SQLite>
48. Алгоритми. [Электронний ресурс]. Режим доступу: <http://ru.wikipedia.org/wiki/Алгоритм>
49. Layout. [Электронний ресурс] Режим доступу: <http://developer.alexanderklimov.ru/android/theory/layout.php>

50. Кавецький В. В. Економічне обґрунтування інноваційних рішень: Практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. — ВНТУ, 2013. — 110 с.

51. Адлер О. О. Методичні вказівки до підготовки та написання курсової роботи з дисципліни «Економічне обґрунтування інноваційних рішень» / Уклад. О. О. Адлер, І. В. Причепа, Н. М. Тарасюк. — Вінниця: ВНТУ, 2014. — 38 с.

52. Тарифи на електроенергію [Електронний ресурс]. Режим доступу: <http://index.minfin.com.ua/tarif/electric.php>. Дата звернення: Листопад 28, 2021.

53. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.

## ДОДАТОК А

Міністерство освіти та науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

\_\_\_\_\_ проф., д.т.н. О. Д. Азаров

«\_\_\_» \_\_\_\_\_ 2021 р.

## ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи  
«Інформаційно-аналітична система визначення психічного стану людини»

08-23.МКР.020.00.000 ТЗ

Науковий керівник: к. т. н., доц. каф ОТ

\_\_\_\_\_ Колесник І.С.

Магістрант групи 2КІ-20м

\_\_\_\_\_ Григоришен О.О.

Опонент: к.т.н., доц., зав. каф МБІС

\_\_\_\_\_ Карпінець В.В.

Вінниця 2021



## 1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність дослідження пов'язана з необхідністю виявлення ситуацій, коли наслідки психічного захворювання і хвороби створюють обмеження життєдіяльності. Унікальна програма реабілітації використовує різноманітні психокорекційні методи для подолання соціально-психологічних проблем пацієнтів, одним з яких є психологічне тестування. За допомогою комп'ютерної діагностики ми можемо спочатку швидко отримати необхідну інформацію про індивіда.

1.2 Наказ про затвердження теми магістерської кваліфікаційної роботи.

## 2 Мета і призначення МКР

2.1 Мета магістерської роботи полягає у розробці інформаційно-аналітичної системи, яка може допомогти діагностувати комплекс наявних симптомів, заощаджуючи час на обробку наявної інформації.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

## 3 Вихідні дані для виконання МКР

Виконати розробку інформаційно-аналітичної системи визначення психічного стану людини. Схеми алгоритмів інформаційно-аналітичної системи та лістинги програми представити в додатках до роботи.

## 4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

- запропонувати інформаційно-аналітичну систему визначення психічного стану людини;
- розробити алгоритм та структуру для інформаційно-аналітичної системи;

- розробити функціонал для генерування висновку та виведення статистичних даних для розробника;
- підтримка сучасними ОС;
- забезпечення зручного та легкого для користувача графічного інтерфейсу;
- результат роботи, а саме мобільний додаток визначення психічного стану.

## 5 Етапи МКР та очікувані результати в таблиці А.1

Таблиця А.1 — Етапи виконання роботи

№	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	01.09.21	02.09.21	Вступ
2	Аналіз літературних джерел для розпізнавання особи	03.09.21	09.09.21	Розділ 1
3	Розробка технічного завдання	10.09.2021	24.09.21	Технічне завдання
3	Розробка структури інформаційно-аналітичної системи визначення психічного стану людини	25.09.21	08.10.21	Розділ 2, розробка структури
4	Розробка додатку, проектування програмного продукту	11.10.21	29.10.21	Розділ 3, розробка програми
5	Практична реалізація, результати.	01.11.21	14.11.21	Розділ 3
6	Розробка економічної частини	15.11.21	30.11.21	Розділ 4
7	Оформлення пояснювальної записки	02.12.21	15.12.21	ПЗ, презентація

## 6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив рецензента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

## 7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

## 8 Вимоги до оформлення МКР

Вимоги викладені в «Положенні про кваліфікаційну роботу у Вінницькому національному технічному університеті» з урахуванням змін, що подані у бюлетені ВАК України № 9-10, 2011р., а також в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008-2015, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

9 Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання до виконання отримав \_\_\_\_\_ Григоришен О.О.

## ДОДАТОК Б

### Принципова схема інформаційно-аналітичної системи

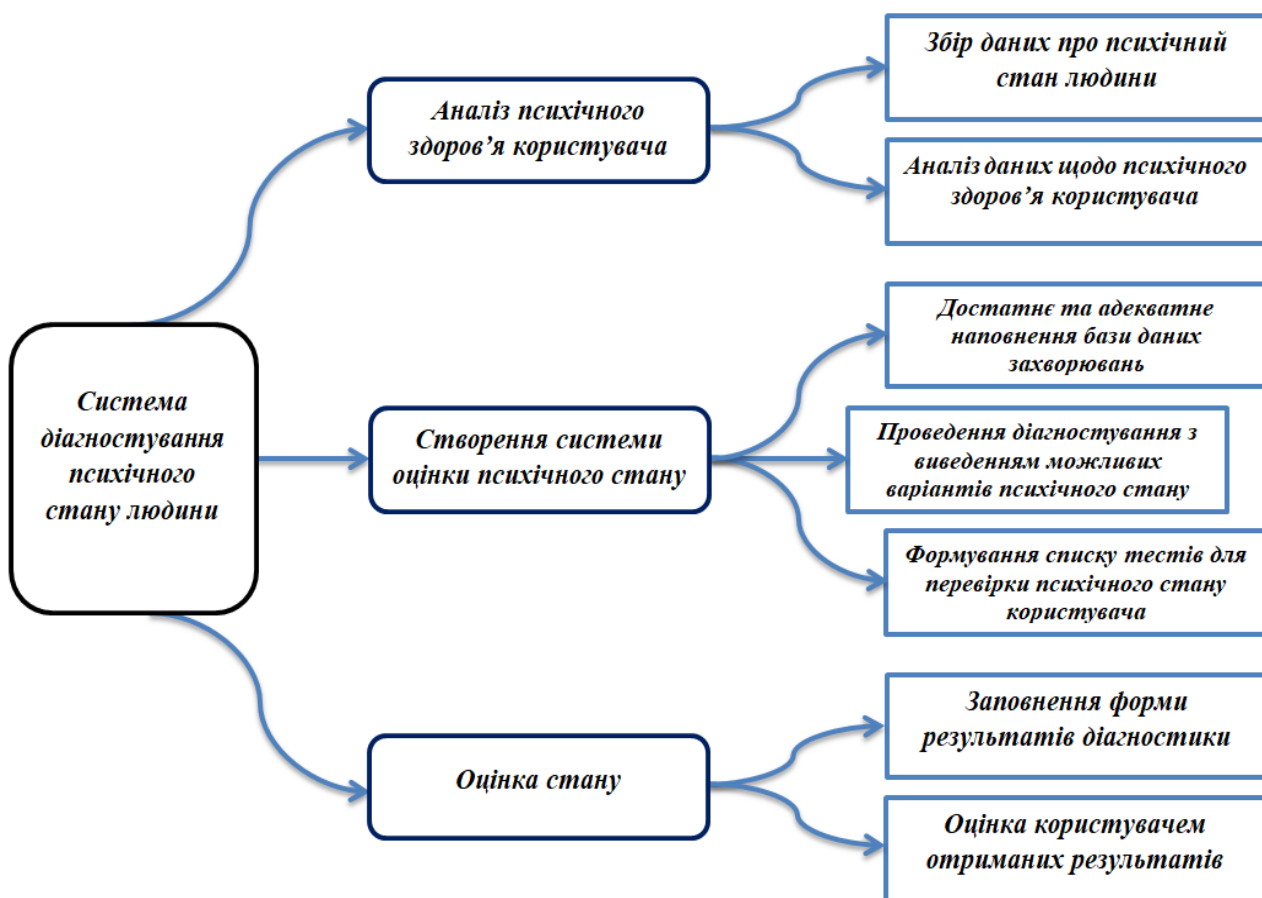


Рисунок Б.1 — Принципова схема інформаційно-аналітичної системи

## ДОДАТОК В

### Деревоподібна структура процесів

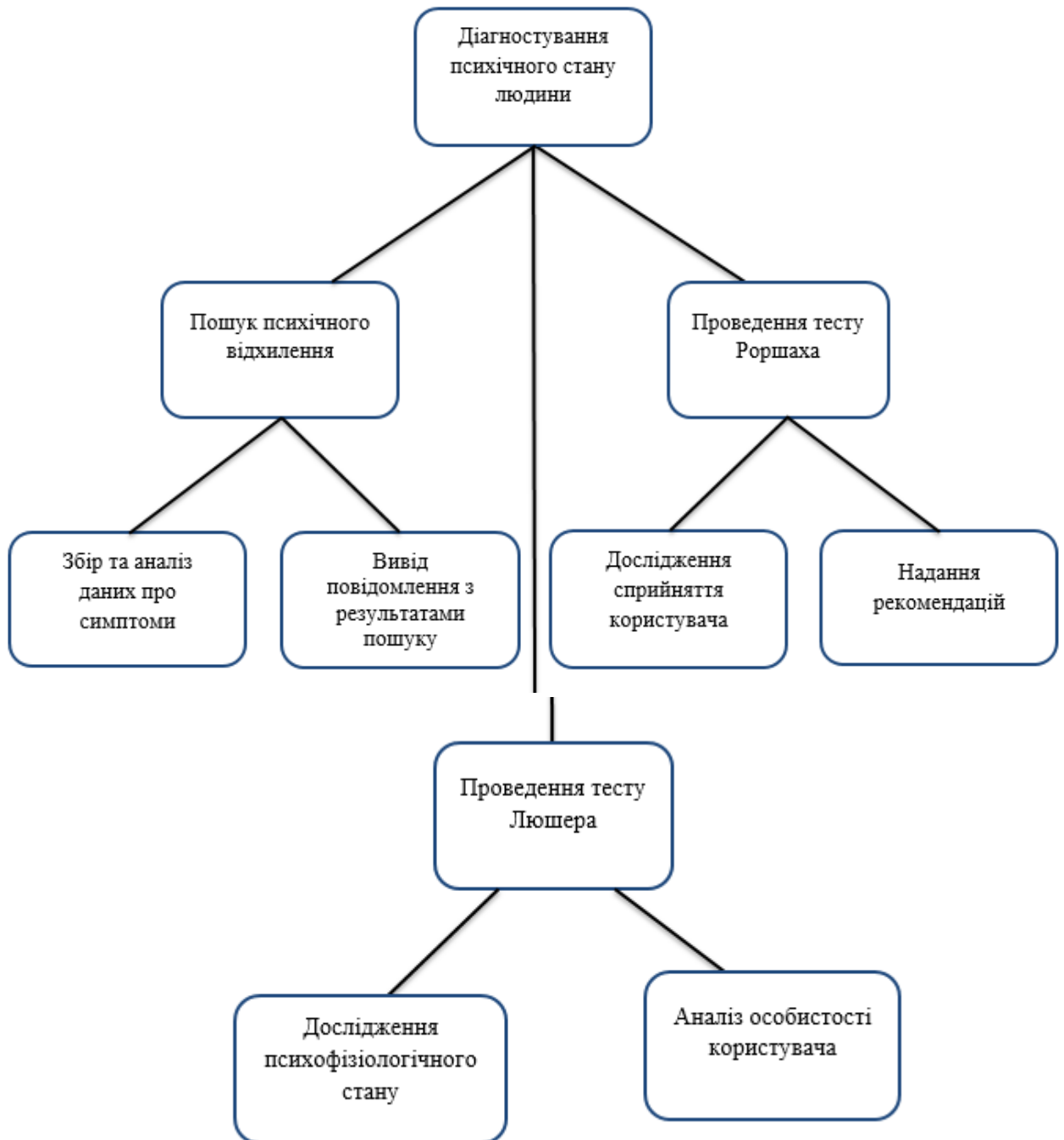


Рисунок В.1 — Деревоподібна структура процесів

## ДОДАТОК Г

### Схема ключових компонентів ієрархії додатку

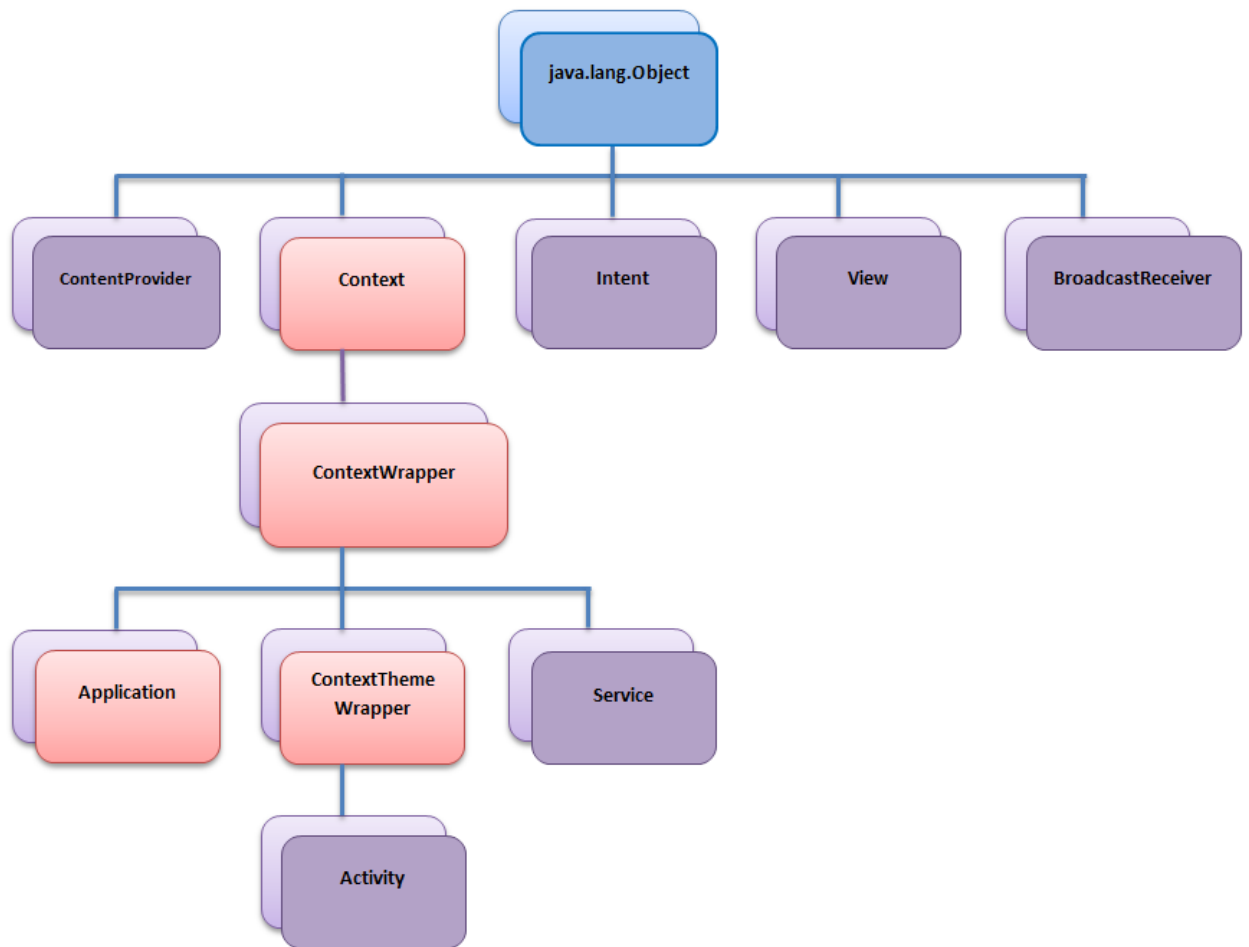


Рисунок Г.1 — Схема ключових компонентів ієрархії додатку

## ДОДАТОК Д

### Алгоритм інформаційно-аналітичної системи

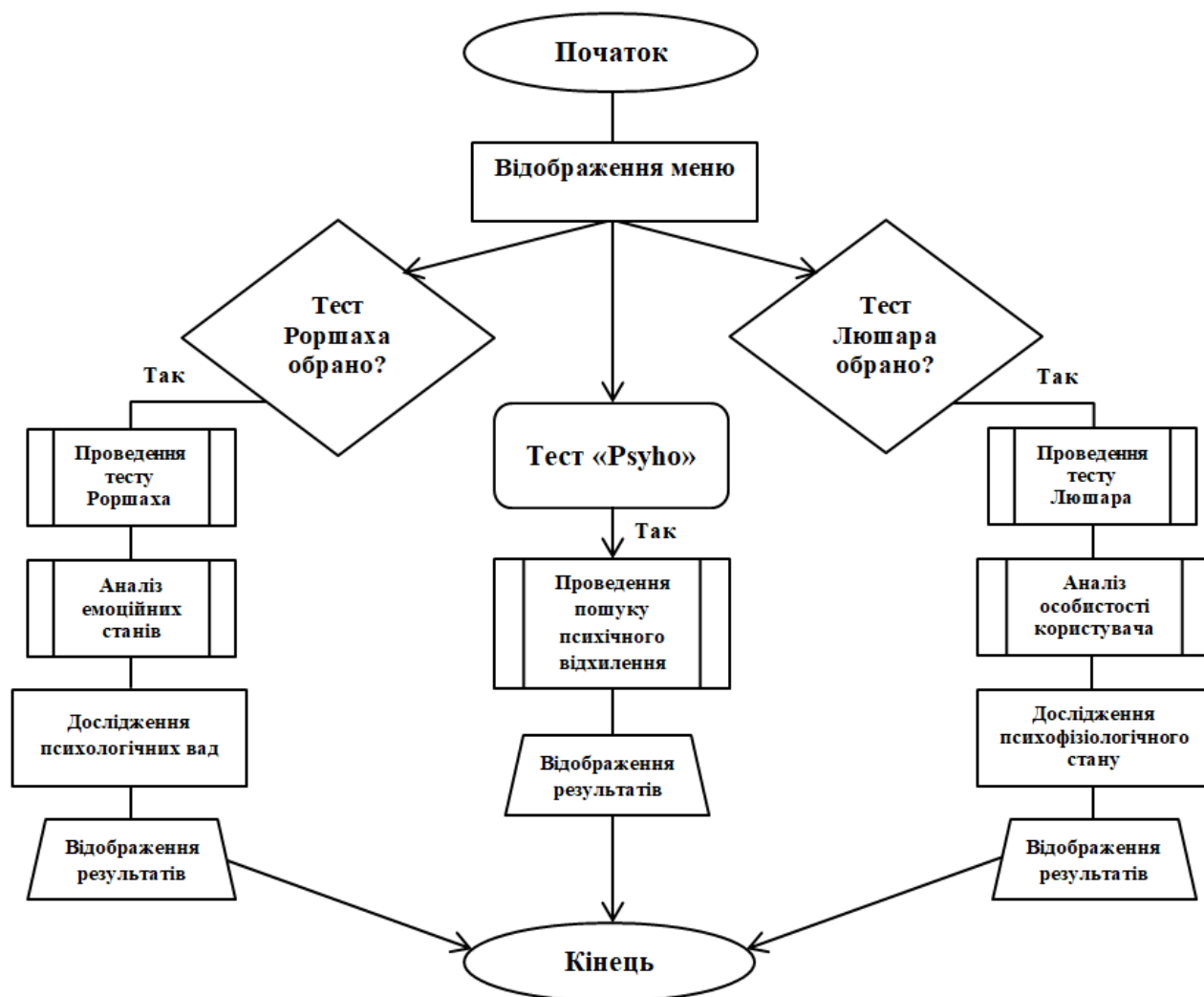


Рисунок Д.1 — Алгоритм інформаційно-аналітичної системи

## ДОДАТОК Е

### Лістинг коду Database.java

```
package olexandr.mentalassistant.data.source;
import java.util.ArrayList;
import java.util.List;
import olexandr.mentalassistant.data.model.Disorder;
import olexandr.mentalassistant.data.model.Symptom;
public class Database {
    private static final int VISUAL_HALLUCINATIONS = 1; // зорові
галюцинації
    private static final int EMOTIONS_CHANGING = 2; // зміна емоційних
станів
    private static final int DELIRIUM = 3; // марення
    private static final int AUDITORY_HALLUCINATIONS = 4; // слухові
галюцинації
    private static final int INFANTILE = 5; // дитинячість
    private static final int TOMFOOLERY = 6; // дуркуватість
    private static final int APATHY = 7; // апатія
    private static final int WEAKNESS = 8; // млявість (фізична слабкість)
    private static final int PESSIMISM = 9; // песимізм
    private static final int BAD_MOOD = 10; // поганий настрій
    private static final int LOW_SELFESTEEM = 11; // низька самооцінка
    private static final int BEDILITATION = 12; // виснаження
    private static final int HELPLESSNESS = 13; // безпомічність
    private static final int IRRITABILITY = 14; // дратівливість
    private static final int RETICENCE = 15; // мовчазність
    private static final int QUITL = 16; // почуття провини
    private static final int KINDLING = 17; // збудження
    private static final int EUPHORIA = 18; // ейфорія
```



private static final int ANGUISH = 19; // туга  
private static final int EXALTATION = 20; // піднесення  
private static final int PROBLEMS\_APPETITE = 21; // втрата апетиту  
private static final int AMNESIA = 22; // амнезія  
private static final int PARALYSIS = 23; // параліч  
private static final int FRISKINESS = 24; // грайливість, пустотливість  
private static final int VOMITING = 25; // блювання  
private static final int NYSTAGMUS = 26; // ністагм  
private static final int MURMUR = 27; // бурмотіння  
private static final int PROBLEMS\_MEMORY = 28; // проблеми з пам'яттю  
private static final int PROBLEMS\_LOGICAL = 29; // проблеми з логічним  
мисленням  
private static final int REDUCED\_SENSITIVITY = 30; // понижена  
чутливість  
private static final int TINGLING = 31; // поколювання  
private static final int PROBLEM\_COORDINATION = 32; // проблеми з  
координацією рухів  
private static final int BYSPNEA = 33; // задишка  
private static final int TACHICARDIA = 34; // тахікардія  
private static final int DISORIENTATION = 35; // дезорієнтація  
private static final int PARAMNESIA = 36; // парамнезія  
private static final int PROBLEMS\_ORIENTATION\_TIME = 37; // проблеми  
з орієнтацією в часі  
private static final int KRYPTOMNESIA = 38; // криптомнезія  
private static final int FIXATION\_AMNESIA = 39; // фіксаційна амнезія  
private static final int MENTAL\_RETARDATION = 40; // розумова  
відсталість  
private static final int DEMENTIA = 41; // слабоумство  
private static final int LOW\_INTELLIGENCE = 42; // низький рівень  
інтелекту

```
private static final int LOW_MENTALITY = 43; // низький рівень психіки
private static final int UNSOCIAL_BEHAVIOR = 44; // соціально
неприйнятна поведінка
private static final int JEALOUSY = 45; // ревності
private static final int PERSECUTION_MANIA = 46; // манія переслідування
private static final int DECREASED_PERFORMANSCE = 47; // зниження
працездатності
private static final int LOW_LEARNING = 48; // зниження рівня навчання
private static final int WITHOUT_WORLDINTERESTING = 49; // зниження
рівня інтересу до навколишнього світу
private static final int OLFACTORY_HALLUCINATIONS = 50; // нюхові
галюцинації
private static final int TACTILE_HALLUCINATIONS = 51; // тактильні
галюцинації
private static final int BODILY_HALLUCINATIONS = 52; // органічні
галюцинації
private static final int CATATINIA_STUPOR = 53; // кататонічний ступор
private static final int CATALEPSY = 54; // воскова гручкість (каталепсія)
private static final int STEREOTYPE = 55; // стереотипія
private static final int ECHOLALIA = 56; // ехолалія
private static final int ECHOPRAXIA = 57; // ехопраксія
private static final int NEGATIVISM = 58; // негативізм
private static final int AMBIVALENCE = 59; // амбівалентність
private static final int MUTISM = 60; // мутизм
private static final int HIGHT_INTELLIGENCE = 61; // високий рівень
інтелекту
private static final int
WITHOUT_UNDESTENDING_PERSONAL_MISTAKES = 62; // нездатність
бачили власні помилки
private List<Symptom> symptomList;
```

```

private List<Disorder> disorderList;
public Database() {
    symptomList = new ArrayList<>();
    disorderList = new ArrayList<>();
    createSymptomList();
    createDisorderList();
}
public List<Disorder> getDisorderList() {
    return disorderList;
}
public List<Symptom> getSymptomList() {
    return symptomList;
}
private Symptom createSymptom(int id, String name, String question, String
description) {
    Symptom symptom = new Symptom();
    symptom.setId(id);
    symptom.setName(name);
    symptom.setQuestion(question);
    symptom.setDescription(description);
    return symptom;
}
private Disorder createDisorder(int id, String name, String description, Integer...
symptoms) {
    Disorder disorder = new Disorder();
    disorder.setId(id);
    disorder.setName(name);
    disorder.setDescription(description);
    disorder.addSymptoms(symptoms);
    return disorder;
}

```

```

}
private void createSymptomList() {
    symptomList.add(createSymptom(VISUAL_HALLUCINATIONS,
        "Зорові галюцинації",
        "Чи присутні зорові галюцинації у хворого?",
        "У хворого викривлене візуальне сприйняття зовнішнього світу,
наприклад присутність об'єктів, " +
        "істот та явищ, яких насправді немає.");
    );
    symptomList.add(createSymptom(EMOTIONS_CHANGING,
        "Зміна емоційних станів",
        "Чи спостерігається у хворого зміна емоційних станів?",
        ""));
    );
    symptomList.add(createSymptom(DELIRIUM,
        "Марення",
        "У хворого бувають марення?",
        ""));
    );
    symptomList.add(createSymptom(AUDITORY_HALLUCINATIONS,
        "Слухові галюцинації",
        "Чи присутні слухові галюцинації у хворого?",
        ""));
    );
    symptomList.add(createSymptom(INFANTILE,
        "Дитячість",
        "Поведінка хворого схожа до поведінки дитини?",
        ""));
    );
    symptomList.add(createSymptom(TOMFOOLERY,

```

```
        "Дуркуватість",
        "У хворого присутня дуркуватість?",
        "")
);
symptomList.add(createSymptom(APATHY,
        "Апатія",
        "Чи є прояви апатії?",
        ""));
symptomList.add(createSymptom(WEAKNESS,
        "Млявість",
        "Чи присутня млявість під час рухів?",
        ""));
symptomList.add(createSymptom(PESSIMISM,
        "Песимізм",
        "У хворого спостерігаються прояви песимізму?",
        ""));
symptomList.add(createSymptom(FRISKINESS,
        "Грайливість",
        "Хворий поводить себе пустотливо та грайливо?",
        ""));
symptomList.add(createSymptom(BAD_MOOD,
        "Поганий настрій",
        "Чи часто хворий знаходиться у поганому настрої?",
        ""));
symptomList.add(createSymptom(LOW_SELFESTEEM,
```

```
        "Низька самооцінка",
        "У хворого низька самооцінка?",
        "")
);
symptomList.add(createSymptom(BEDILITATION,
        "Виснаження",
        "У хворого присутнє фізичне чи емоційне виснаження?",
        ""));
symptomList.add(createSymptom(HELPLESSNESS,
        "Безпомічність",
        "Хворий відчувається безпомічним?",
        ""));
symptomList.add(createSymptom(IRRITABILITY,
        "Дратівливість",
        "Чи підвищена дратівливість?",
        ""));
symptomList.add(createSymptom(RETICENCE,
        "Мовчазність",
        "Дана людина мовчазна?",
        ""));
symptomList.add(createSymptom(QUITL,
        "Почуття провини",
        "Чи присутнє у хворого почуття провини?",
        ""));
symptomList.add(createSymptom(KINDLING,
```

```
        "Збудження",
        "Чи проявляється у хворого загальне збудження?",
        "")
);
symptomList.add(createSymptom(EUPHORIA,
        "Ейфорія",
        "У хворого буває відчуття ейфорії?",
        ""));
symptomList.add(createSymptom(ANGUISH,
        "Туга",
        "У хворого буває відчуття туги?",
        ""));
symptomList.add(createSymptom(EXALTATION,
        "Піднесення",
        "У хворого буває піднесений настрій?",
        ""));
symptomList.add(createSymptom(PROBLEMS_APPETITE,
        "Втрата апетиту",
        "Чи спостерігається втрата апетиту?",
        ""));
symptomList.add(createSymptom(PARALYSIS,
        "Параліч",
        "Чи страждає хворий на параліч?",
        ""));
symptomList.add(createSymptom(MURMUR,
```

```
        "Бурмотіння",
        "Чи присутнє порушення мови у вигляді бурмотіння?",
        "")
);
symptomList.add(createSymptom(PROBLEMS_MEMORY,
        "Проблеми з пам'яттю",
        "Чи бувають проблеми з пам'яттю?",
        ""));
symptomList.add(createSymptom(PROBLEMS_LOGICAL,
        "Проблеми з логічним мисленням",
        "У хворого порушене логічне мислення? ",
        ""));
symptomList.add(createSymptom(REduced_sensitivity,
        "Понижена чутливість",
        "У хворого понижена чутливість?",
        ""));
symptomList.add(createSymptom(TINGLING,
        "Поколювання",
        "Чи бувають у хворого поколювання у різних частинах тіла?",
        ""));
symptomList.add(createSymptom(PROBLEM_COORDINATION,
        "Проблеми з координацією рухів",
        "Чи виникають складнощі з координацією рухів?",
        ""));
);
```



```
symptomList.add(createSymptom(BYSPNEA,  
    "Задишка",  
    "У хворого буває задишка?",  
    ""))  
);  
symptomList.add(createSymptom(TACHICARDIA,  
    "Тахікардія",  
    "Хворий страждає на тахікардію?",  
    ""))  
);  
symptomList.add(createSymptom(DISORIENTATION,  
    "Дизорієнтація",  
    "Чи виникають проблеми із орієнтацією у просторі?",  
    ""))  
);  
symptomList.add(createSymptom(PARAMNESIA,  
    "Парамнезія",  
    "Чи виникають у хворого хибні або ж перекручені спогади, що  
інколи супроводжуються зміщенням у часі?",  
    ""))  
);  
symptomList.add(createSymptom(PROBLEMS_ORIENTATION_TIME,  
    "Проблеми з орієнтацією в часі",  
    "Чи виникають проблеми із орієнтуванням у часі?",  
    ""))  
);  
symptomList.add(createSymptom(KRYPTOMNESIA,  
    "Криптомнезія",  
    "Чи бувають у хворого проблеми із визначенням джерела раніше  
отриманої інформації?",
```

```
    ""
);
symptomList.add(createSymptom(FIXATION_AMNESIA,
    "Фіксаційна амнезія",
    "Чи виникають у хворого проблеми із запам'ятовуванням подій, що
відбуваються у реальному часі?",
    ""
);
symptomList.add(createSymptom(MENTAL_RETARDATION,
    "Розумова відсталість",
    "Чи спостерігається розумова відсталість у хворого?",
    ""
);
symptomList.add(createSymptom(DEMENTIA,
    "Слабоумство",
    "Чи страждає на слабоумство?",
    ""
);
symptomList.add(createSymptom(LOW_INTELLIGENCE,
    "Низький рівень інтелекту",
    "У хворого низький рівень інтелекту?",
    ""
);
symptomList.add(createSymptom(LOW_MENTALITY,
    "Низький рівень психічного розвитку",
    "У хворого низький рівень психічного розвитку?",
    ""
);
symptomList.add(createSymptom(UNSOCIAL_BEHAVIOR,
    "Соціально неприйнятна поведінка",
```

```
        "Чи буває у хворого соціально неприйнятна поведінка?",
        """);
);
symptomList.add(createSymptom(JEALOUSY,
        "Ревнощі",
        "Чи буває у хворого приступ необґрунтованих ревнощів?",
        """));
);
symptomList.add(createSymptom(PERSECUTION_MANIA,
        "Манія переслідування",
        "Хворий страждає на манію переслідування?",
        """));
);
symptomList.add(createSymptom(DECREASED_PERFORMANSCE,
        "Зниження рівня працездатності",
        "Чи спостерігається зниження рівня працездатності?",
        """));
);
symptomList.add(createSymptom(LOW_LEARNING,
        "Зниження здатності навчатись",
        "Чи спостерігається зниження здатності до навчання?",
        """));
);
symptomList.add(createSymptom(WITHOUT_WORLDINTERESTING,
        "Відсутність інтересу до довколишнього світу",
        "Хворий поводить ся так, ніби його не цікавить довколишній світ?",
        """));
);
symptomList.add(createSymptom(OLFACTORY_HALLUCINATIONS,
        "Нюхові галюцинації",
```

```
        "Чи присутні нюхові галюцинації у хворого?",
        ""
    );
    symptomList.add(createSymptom(TACTILE_HALLUCINATIONS,
        "Тактильні галюцинації",
        "Чи присутні тактильні галюцинації у хворого?",
        ""
    );
    symptomList.add(createSymptom(BODILY_HALLUCINATIONS,
        "Органічні галюцинації",
        "Хворий жаліється на проблеми із внутрішніми органами?",
        ""
    );
    symptomList.add(createSymptom(CATATINIA_STUPOR,
        "Кататонічний ступор",
        "Чи застигає хворий у одній позі протягом тривалого часу?",
        ""
    );
    symptomList.add(createSymptom(CATALEPSY,
        "Каталепсія",
        "Чи зберігає хворий пози, механічно надані його тілу?",
        ""
    );
    symptomList.add(createSymptom(STEREOTYPE,
        "Стереотипія",
        "Чи спостерігаються моготонні розхитування чи інші одноманітні
рухи?",
        ""
    );
    symptomList.add(createSymptom(ECHOLALIA,
```

```
        "Ехолалія",
        "Чи повторює хворий слова, які він чує?",
        "")
);
symptomList.add(createSymptom(ECHOPRAXIA,
        "Ехопраксія",
        "Хворий повторює побачені дії?",
        ""));
symptomList.add(createSymptom(NEGATIVISM,
        "Негативізм",
        "Хворий робить протилежне тому, чого від нього вимагають?",
        ""));
symptomList.add(createSymptom(AMBIVALENCE,
        "Амбівалентність",
        "Чи спостерігаєть ставлення до когось чи до чогось одночасно
позитивно і негативно?",
        ""));
symptomList.add(createSymptom(MUTISM,
        "Мутизм",
        "У хворого спостерігався стан, коли він не відповідав на запитання
чи навіть знаками не давав" +
        " зрозуміти, що він згоден вступити в контакт з оточуючими?",
        ""));
symptomList.add(createSymptom(HIGHT_INTELLIGENCE,
        "Високий рівень інтелекту",
        "Спостерігається високий рівень інтелекту?",
```

```

        """);
    );
symptomList.add(createSymptom(WITHOUT_UNDESTENDING_PERSONAL_
MISTAKES,
        "Нездатність бачити власні помилки",
        "Хворий здатний помічати та усвідомлювати власні помилки та
проблеми?",
        """);
    );
}
private void createDisorderList() {
    disorderList.add(createDisorder(1,
        "Параноя",
        "Опис параної",
        VISUAL_HALLUCINATIONS, EMOTIONS_CHANGING,
        DELIRIUM, UNSOCIAL_BEHAVIOR, JEALOUSY,
        PERSECUTION_MANIA)
    );
    disorderList.add(createDisorder(2,
        "Проста шизофренія",
        "Опис простої шизофренії",
        VISUAL_HALLUCINATIONS, AUDITORY_HALLUCINATIONS,
        DECREASED_PERFORMANSCE, LOW_LEARNING,
        WITHOUT_WORLDINTERESTING, OLFATORY_HALLUCINATIONS,
        TACTILE_HALLUCINATIONS, BODILY_HALLUCINATIONS)
    );
    disorderList.add(createDisorder(3,
        "Гебефренічна шизофренія",
        "Опис Гебефренічної шизофренії",
        INFANTILE, TOMFOOLERY, FRISKINESS)
    );
}
}

```

```
);
disorderList.add(createDisorder(4,
    "Депресія",
    "Опис депресії",
    BAD_MOOD, LOW_SELFESTEEM, PESSIMISM, APATHY,
WEAKNESS, HELPLESSNESS, IRRITABILITY,
    RETICENCE, QUITL)
);
disorderList.add(createDisorder(5,
    "Біполярний афективний розлад",
    "Опис біполярного афективного розладу",
    EUPHORIA, KINDLING, EXALTATION, ANGUISH)
);
disorderList.add(createDisorder(6,
    "Циклотимія",
    "Опис циклотимії",
    KINDLING, EUPHORIA, EMOTIONS_CHANGING, EXALTATION,
ANGUISH)
);
disorderList.add(createDisorder(7,
    "Синдром Корсакова",
    "Опис синдрому Корсакова",
    AMNESIA, PROBLEMS_ORIENTATION_TIME,
DISORIENTATION, PARALYSIS, FIXATION_AMNESIA,
    KRYPTOMNESIA)
);
disorderList.add(createDisorder(8,
    "Кататонічна шизофренія",
    "Опис кататонічної шизофренії",
```

CATATINIA\_STUPOR, STEREOTYPE, CATALEPSY, ECHOLALIA,  
ECHOPRAXIA, NEGATIVISM,           AMBIVALENCE, MUTISM)

);

disorderList.add(createDisorder(9,

    "Параноїдальна шизофренія",

    "Опис параноїдальної шизофренії",

    HIGHT\_INTELLIGENCE,

WITHOUT\_UNDESTENDING\_PERSONAL\_MISTAKES,

PERSECUTION\_MANIA, UNSOCIAL\_BEHAVIOR)

); }

}



## ДОДАТОК Ж

Лістинг файлів програмного коду

Ж.1 — Disorder.java

```
package olexandr.mentalassistant.data.model;
import java.util.ArrayList;
import java.util.List;
public class Disorder {
    private String name;
    private int id;
    private List<Integer> symptomIdList;
    private String description;
    public Disorder() {
        symptomIdList = new ArrayList<>();
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public List<Integer> getSymptomIdList() {
        return symptomIdList;
    }
    public void setSymptomIdList(List<Integer> symptomIdList) {
        this.symptomIdList = symptomIdList;
    }
    public String getDescription() {
        return description;
    }
}
```

```

public void setDescription(String description) {
    this.description = description;
}
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public void addSymptoms(Integer... symptoms) {
    for (Integer symptom : symptoms) {
        symptomIdList.add(symptom);
    }
}
}

```

## Ж.2 – Symptom.java

```

package olexandr.mentalassistant.data.model;
public class Symptom {
    private int id;
    private String name;
    private String question;
    private String description;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {

```

```
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getQuestion() {
        return question;
    }
    public void setQuestion(String question) {
        this.question = question;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

### Ж.3 – DiagnosticsActivity.java

```
package olexandr.mentalassistant.ui;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.List;
import olexandr.mentalassistant.R;
```

```

import olexandr.mentalassistant.data.model.Disorder;
import olexandr.mentalassistant.data.source.Database;
import olexandr.mentalassistant.data.model.Symptom;
import olexandr.mentalassistant.utils.DisorderSearch;
public class DiagnosticsActivity extends AppCompatActivity {
    private TextView question;
    private TextView textStatus;
    private int currentQuestion;
    private List<Symptom> symptomList;
    private List<Integer> selectedSymptom;
    private List<Disorder> disorderList;
    private boolean symptomConfirmed;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(olexandr.mentalassistant.R.layout.activity_diagnostics);
        initView();
        Database database = new Database();
        disorderList = database.getDisorderList();
        symptomList = database.getSymptomList();
        selectedSymptom = new ArrayList<>();
        currentQuestion = 0;
        askQuestion();
    }
    private void initView() {
        textStatus = (TextView) findViewById(R.id.status);
        question = (TextView) findViewById(R.id.question);
        Button buttonYes = (Button) findViewById(R.id.buttonYes);
        buttonYes.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View view) {
    Symptom symptom = symptomList.get(currentQuestion);
    disorderList = DisorderSearch.filterDisorders(disorderList,
symptom.getId(), true);
    selectedSymptom.add(symptom.getId());
    textStatus.setText(Integer.toString(disorderList.size()));
    checkResults();
}
});
Button buttonNo = (Button) findViewById(R.id.buttonNo);
buttonNo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        textStatus.setText(Integer.toString(disorderList.size()));
        checkResults();
//        disorderList = Finder.filterDisorders(disorderList,
//            symptomList.get(currentQuestion).getId(), false);
//        textStatus.setText(Integer.toString(disorderList.size()));
//        checkResults();
    }
});
Button buttonUnknown = (Button) findViewById(R.id.buttonUnknown);
buttonUnknown.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        textStatus.setText(Integer.toString(disorderList.size()));
        checkResults();
    }
}

```

```

    });
} private void checkResults() {
    if (disorderList.size() == 0) {
        //disorder not found
        Toast.makeText(this, "disorder not found", Toast.LENGTH_SHORT).show();
    } else {
        currentQuestion++;
    }
    if (currentQuestion < symptomList.size()) {
        askQuestion();
    } else {
        for (Disorder disorder:disorderList) {
            if(selectedSymptom.containsAll(disorder.getSymptomIdList())){
                Toast.makeText(this, "Психічне відхилення встановлено: " +
                    disorderList.get(0).getName(), Toast.LENGTH_LONG).show();
                Intent intentStartDiagnostics = new Intent(DiagnosticsActivity.this,
MainActivity.class);
                startActivity(intentStartDiagnostics);
            } else{
                Toast.makeText(this, "Найбільш відрогідне психічне відхилення: " +
                    disorderList.get(0).getName(), Toast.LENGTH_LONG).show();
                Intent intentStartDiagnostics = new Intent(DiagnosticsActivity.this,
MainActivity.class);
                startActivity(intentStartDiagnostics);
            }
        }
    }
}
private void askQuestion() {

```

```

boolean found = false;
Symptom symptom = symptomList.get(currentQuestion);
for (Disorder disorder : disorderList) {
    if (disorder.getSymptomIdList().contains(symptom.getId())) {
        found = true;
        break;
    }
}
if (found) {
    question.setText(symptom.getQuestion());
} else {
    checkResults();
}
}
}

```

#### Ж.4 – MainActivity.java

```

package olexandr.mentalassistant.ui;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import olexandr.mentalassistant.R;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button buttonStart = (Button) findViewById(R.id.buttonStart);
    }
}

```

```

Button button = (Button) findViewById(R.id.button);
buttonStart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intentStartDiagnostics = new Intent(MainActivity.this,
DiagnosticsActivity.class);
        startActivity(intentStartDiagnostics);
    }
});
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intentStat = new Intent(MainActivity.this, TestActivity.class);
        startActivity(intentStat);
    }
}); }
}

```

#### Ж.5 – TestActivity.java

```

package olexandr.mentalassistant.ui;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import olexandr.mentalassistant.R;
public class TestActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_test);
    }
}

```



```
}
```

## Ж.6 – DisorderSearch.java

```
package olexandr.mentalassistant.utils;
import android.util.Log;
import java.util.ArrayList;
import java.util.List;
import olexandr.mentalassistant.data.model.Disorder;
public class DisorderSearch {
    public static List<Disorder> filterDisorders(List<Disorder> disorderList, int
currentSymptomId, boolean symptomContained) {
        List<Disorder> potentialDisorderList = new ArrayList<>();
        for (Disorder disorder : disorderList) {
            Log.d("DisorderSearch", "Disorder" + disorder.getName() + "\n" +
disorder.getSymptomIdList() + "\n" + currentSymptomId);
            List<Integer> symptomIdList = disorder.getSymptomIdList();
            if (symptomContained && symptomIdList.contains(currentSymptomId)) {
                potentialDisorderList.add(disorder);
            } else if (!symptomContained &&
!symptomIdList.contains(currentSymptomId)) {
                potentialDisorderList.add(disorder);
            }
        }
        return potentialDisorderList;
    }
}
```

## ДОДАТОК И

### Лістинг коду розміток

#### И.1 — AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="olexandr.mentalassistant">
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".ui.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".ui.DiagnosticsActivity"></activity>
    <activity android:name=".ui.TestActivity"></activity>
  </application>
</manifest>
```

#### И.2 — activity\_main.xml

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

```
<LinearLayout
```

```
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:gravity="center"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentStart="true">
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="15sp"  
    android:layout_alignParentStart="true"  
    android:text="Будь ласка, оберіть необхідний функціонал:"  
    android:layout_margin="20dp"/>
```

```
<Button
```

```
    android:id="@+id/buttonStart"  
    android:layout_width="170dp"  
    android:layout_height="wrap_content"  
    android:layout_margin="20dp"  
    android:text="Діагностування">
```

```
</Button>
```

```
<Button
```

```
    android:id="@+id/button"  
    android:layout_width="162dp"  
    android:layout_height="wrap_content"  
    android:text="Довідник"  
    android:layout_margin="2dp">
```

```
</Button>
```

```
</LinearLayout>
```

</RelativeLayout>

### И.3 — Activity\_diagnostics.xml

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

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="horizontal">
```

```
<TextView
    android:id="@+id/question"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="12dp"
    android:textSize="18sp"
    android:text="@string/app_name" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:gravity="bottom|center"
    android:orientation="horizontal"
    android:padding="120dp">
```

```
<TextView
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textSize="10sp"
        android:layout_alignParentStart="true"
        android:text="Залишилось варіантів: " />
<TextView
    android:id="@+id/status"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:textSize="10sp"
    android:text="10" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="bottom|center"
    android:orientation="horizontal"
    android:padding="21dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true">
    <Button
        android:id="@+id/buttonYes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:text="Так"></Button>
    <Button
        android:id="@+id/buttonUnknown"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:text="Невідомо"></Button>
<Button
    android:id="@+id/buttonNo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hi"></Button>
</LinearLayout>
</RelativeLayout>
```

#### И.4 — Activity\_test.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"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/activity_test"
        android:id="@+id/imageView" />
</LinearLayout>
```

# ДОДАТОК К

## Протокол перевірки кваліфікаційної роботи

### ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Інформаційно-аналітична система визначення психічного стану людини.

Тип роботи: магістерська кваліфікаційна робота

(кваліфікаційна роботи, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))

Підрозділ кафедра обчислювальної техніки

(кафедра, факультет (інститут), навчальна група)

Науковий керівник Колесник І.С.к.т.н., доц. каф.ОТ

(прізвище, ініціали, посада)

#### Показники звіту подібності

Plagiat.pl (StrikePlagiarism)		Unicheck	
КП1		Оригінальність	99,6
КП2			
Тривога/Білі знаки	/	Схожість	0,4

#### Аналіз звіту подібності (відмінити подібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності і відсутності самостійності її автора. Робот направити на доопрацювання.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений(-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор \_\_\_\_\_

(підпис)

Григорішен О.О.

(прізвище, ініціали)

#### Опис прийнятого рішення

Ступінь оригінальності роботи відповідає вимогам, що висуваються до МКР

Особа, відповідальна за перевірку \_\_\_\_\_

(підпис)

Захарченко С.М.

(прізвище, ініціали)

Експерт \_\_\_\_\_

(за потреби) (підпис)

(прізвище, ініціали)

