

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Ялика Вадима Ігоровича
Прізвище, ім'я, по батькові студента(ки)

на здобуття ступеня вищої освіти Бакалавра


Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор
Назва теми


Галузь знань 12 «Інформаційні технології»


Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.200177.01.26.ПЗ

Виконав студент IV курсу, група ІПЗ-20-1  Вадим ЯЛИК
Ім'я, ПРІЗВИЩЕ

Керівник канд. пед. наук, доцент  Наталія ПРАВОРСЬКА
Науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Нормоконтролер канд.тех. наук, доцент  Оксана ЯШИНА
Посада Ім'я, ПРІЗВИЩЕ


До захисту допускаю:
Завідувач кафедри інженерії програмного забезпечення  Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

4 червня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри 173
Л. П. Бедратюк 
02 01 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Ялик Вадим Ігорович
Прізвище, ім'я, по батькові студента

1. Тема роботи Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор

Керівник кваліфікаційної роботи Праворська Наталія Іванівна канд. пед. наук, доцент
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2024 р. № 6

2. Строк подання студентом роботи на кафедру 01.06.2024 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

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

Дослідження предметної області та постановка задачі

Проектування програмного забезпечення

Програмна реалізація та тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

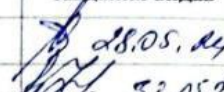
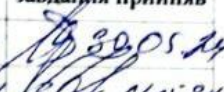
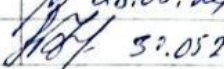

Три креслення:

1. Діаграма декомпозиції застосунку

2. UML-діаграма послідовності

3. Діаграма класів застосунку

6. Консультанти розділів дипломного проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Яшина О.М., доцент	 28.05.24	 29.05.24
Антиплагіат	Форкун Ю. В., доцент	 30.05.24	 06.05.24

7. Дата видачі завдання «02» січня 2024р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою кваліфікаційних робіт (КвР), визначення та узгодження індивідуальної теми	01.12 – 31.12.2023	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	01.01 – 20.02.2024	
3 Проектування програмного забезпечення	21.02 – 20.03.2024	
4 Програмна реалізація з використанням відповідних засобів розробки. Тестування ПЗ	21.03 – 30.04.2024	
5 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог стандартів	01.05 – 25.05.2024	
6 Попередній захист КвР	Травень 2024	Згідно графіка
7 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2024	
8 Здача КвР на кафедрі; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	з 01.06.2024	

Студент


Підпис

Вадим ЯЛИК
Ім'я, ПРІЗВИЩЕ

Керівник роботи


Підпис

Наталія ПРАВОРСЬКА
Ім'я, ПРІЗВИЩЕ

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.200177.01.26.ПЗ	Пояснювальна записка	62		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4		Презентаційні матеріали	13		
5	A3		Креслення	3		

КвРІПЗ.200177.01.26.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав.		Ялик В.І.	<i>[Підпис]</i>	06.06
Керівник.		Праворська Н.І.	<i>[Підпис]</i>	06.06
Рецензент			<i>[Підпис]</i>	
Н. Контр.		Яшина О.М.	<i>[Підпис]</i>	06.06
Затверд.		Бедратюк Л.П.	<i>[Підпис]</i>	06.06
Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор Пояснювальна записка				
		Лім.	Арк.	Аркуші
			5	62
ХНУ, ІПЗ-20-1				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор».

Автор роботи: Ялик Вадим Ігорович.

Керівник роботи: Праворська Наталія Іванівна

Пояснювальна записка: 62 с., 18 рис., 5 табл., 3 дод., 30 джерел.

Графічна частина: 15 слайдів

МОБІЛЬНИЙ ЗАСТОСУНОК, НАВЧАННЯ, МАТЕМАТИКА, ГРА,
ANDROID, FIREBASE, JAVA.

Метою роботи є створення програмного застосунку для навчання математиці учнів початкових класів з використанням інтерактивних ігор.

У кваліфікаційній роботі проведено аналіз предметної області та її інформаційного забезпечення, визначені вимоги до програмного застосунку для навчання математиці, розроблена загальна архітектура застосунку, спроектована структура бази даних та структура застосунку.

Для розробки програмного застосунку використано мову програмування Java, сервер бази даних Firebase.

У результаті проектування була здійснена програмна реалізація програмного застосунку для навчання математиці учнів початкових класів з використанням інтерактивних ігор.

Розроблений застосунок призначений для учнів початкових класів з метою покращення їх навичок у виконанні базових математичних операцій, а також покращення ряду інших навичок

06.06.2024
Дата

Я
Підпис

ЗМІСТ

ВСТУП	9
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	10
1.1 Змістовий аналіз предметної області, її структурних та функціональних особливостей	10
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.....	13
1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення.....	20
1.4 Висновки. Постановка задачі.....	24
2 Проектування програмного забезпечення.....	25
2.1 Архітектура та функціональна структура застосунку.....	25
2.2 Проектування структури бази даних.....	28
2.3 Проектування інтерфейсу користувача.....	30
2.4 Розроблення алгоритму роботи мобільного застосунку.....	34
2.5 Створення прототипу мобільного застосунку.....	37
2.6 Аналіз та вибір технологій і методів реалізації застосунку.....	40
2.7 Висновки.....	44
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	45
3.1 Реалізація логіки мобільного застосунку	45
3.2 Реалізація розмітки мобільного застосунку.....	50
3.3 Розроблення бази даних	54
3.4 Керівництво користувача.....	57
3.5 Технічні характеристики мобільного застосунку.....	60
3.6 Тестування мобільного застосунку.....	61
3.6.1 Аналіз методів тестування мобільного застосунку.....	62

					КвРІПЗ.200177.01.26.ПЗ			
Змн	Арк.	№ докум.	Підпис	Дата	Програмний застосунок для навчання математики учнів початкових класів з використанням інтерактивних ігор Пояснювальна записка	Літ.	Арк.	Аркуше
Виконав.		Ялик В.І.	<i>[Підпис]</i>	06.02			6	62
Керівник.		Праворська Н.І.	<i>[Підпис]</i>					
Рецензент			<i>[Підпис]</i>					
Н. Контр.		Яшина О.М.	<i>[Підпис]</i>	06.05				
Затверд.		Бедратюк Л.П.	<i>[Підпис]</i>	06.05				
						ХНУ, ІПЗ-20-1		

3.6.2 Тестування мобільного застосунку за допомогою емулятора.....	63
3.6.3 Аналіз результатів тестування мобільного застосунку.....	66
3.7 Висновки.....	66
ВИСНОВКИ	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	70
ДОДАТОК А	73
ДОДАТОК Б	77
ДОДАТОК В	91

					КвРІПЗ.200177.01.26.ПЗ			
Змн	Арк.	№ докум.	Підпис	Дата	Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор Пояснювальна записка	Лит.	Арк.	Аркуші
Виконав.		Ялик В.І.	<i>[Підпис]</i>	06.06			7	62
Керівник.		Праворська Н.І.	<i>[Підпис]</i>	06.06				
Рецензент			<i>[Підпис]</i>					
Н. Контр.		Яшина О.М.	<i>[Підпис]</i>	06.06				
Затверд.		Бедратюк Л.П.	<i>[Підпис]</i>	06.06				
						ХНУ, ІПЗ-20-1		

ВСТУП

У сучасному світі мобільні застосунки стають все більш популярними інструментами для навчання та розвитку дітей. Навчальні ігри на мобільних пристроях можуть стати ефективними інструментами для залучення учнів до навчання, особливо в математиці.

Проте існуючі мобільні застосунки не завжди відповідають потребам учнів та вчителів, тому в рамках даної кваліфікаційної роботи необхідним є дослідити, яким повинен бути мобільний застосунок, щоб задовольнити ці потреби.

Метою дослідження, яке проводиться під час даної кваліфікаційної роботи є отримання знань, які можуть бути необхідними для того, щоб в подальшому розробити мобільний застосунок, призначений для покращення рівня математичної грамотності та навичок учнів початкової школи. Застосунок повинен містити щось, що буде зацікавлювати його цільову аудиторію.

Одним з основних аспектів такого дослідження є аналіз уже існуючих рішень даної проблеми, після якого можна щось запозичити з цих рішень, зробити щось схоже, а також, по можливості уникнути проблем, з якими зіткнулись дані рішення.

Результати дослідження можуть бути корисними для освітніх установ, батьків та вчителів, які бажають покращити навчання та розвиток математичних навичок учнів. Вони також підкреслюють важливість використання мобільних технологій, як інструмента, який здатен покращити процес навчання в сучасному освітньому середовищі.

Актуальність теми "Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор" визначається насамперед наступними чинниками:

– цифрова трансформація в освіті: оскільки сучасна система освіти переживає бум у використанні цифрових технологій, через що мобільні застосунки стають невід'ємною частиною цього процесу, особливо в початковій школі, де діти вже звикли до смартфонів і планшетів;

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

– покращення математичної грамотності: математика є одним із ключових предметів у навчальному процесі. Розробка і впровадження мобільних застосунків, спрямованих на покращення математичної грамотності учнів початкової школи, має стратегічне значення для підвищення якості освіти;

– зручний доступ до навчального матеріалу: мобільні застосунки надають можливість учням навчатися в будь-який час і в будь-якому місці, що є досить корисним для підтримки самостійного навчання в своєму темпі.

– розвиваючі ігри: використання гейміфікації в мобільних застосунках дозволяє зробити навчання математики захоплюючим та цікавим для дітей. Ігровий підхід сприяє більш ефективному засвоєнню.

Для того, щоб дослідження можна було назвати успішним, в даній кваліфікаційній роботі необхідно виконати наступні завдання:

– дослідити предметну область, з урахування особливостей того, як вона реалізовується та функціонує;

– виконати аналіз існуючих рішень на ринку, визначити їх переваги та недоліки і мінімальний функціонал;

– визначити вимоги до мобільного застосунку, який буде розроблюватись в подальшому, на основі отриманих з даного дослідження даних;

– на основі визначених вимог, та проаналізованих рішень на ринку, спроектувати архітектуру мобільного застосунку, обравши для цього найбільш підходяще архітектурне рішення;

– визначити яким має бути інтерфейс та дизайн розроблюваного застосунку використовуючи, як запозичені ідеї з готових рішень, так і власні;

– визначити те, як має працювати розроблюваний застосунок;

– обрати технології та інструменти необхідні для успішної розробки;

– на основі всіх зібраних даних розробити мобільний застосунок, та після його розробки, виконати тестування для того щоб перевірити чи відповідає він визначеним вимогам.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА

ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Перед початком розробки програмного застосунку, необхідним є дослідження предметної області. В даній кваліфікаційній роботі такою предметною областю виступає навчальна гра, що матиме мету покращити знання дітей в математиці. Для кращого розуміння потрібно дослідити, що з себе представляє навчальна гра.

Навчальна гра – є формою ігрової діяльності, що спрямована на покращення засвоєння навчального матеріалу шляхом використання ігрових компонентів. Такі ігри створюються з ціллю полегшити навчання у різних областях, таких як: вивчення конкретних тем; розвиток певних концепцій; усвідомлення історичних подій; культурного розвитку та набуття нових навичок через ігровий процес^[1].

Навчальні ігри мають досить довгу історію, що починається ще зі стародавнього часу. Ще в давніх цивілізаціях, таких як Єгипет, Греція та Рим, існували ігри, які використовувалися для навчання дітей основам математики, логіки та стратегічного мислення. Однією з таких була гра "Манкала". Вона виникла в Єгипті близько 1400 років до н.е. і використовувалася для розвитку обчислювальних навичок.

У середньовічній Європі теж з'явилися навчальні ігри, які використовувалися в освітніх закладах. Одним з відомих прикладів є гра "Шахи", яка використовувалась для навчання тактики, стратегії та логічного мислення.

З розвитком технологій і появою персональних комп'ютерів та мобільних пристроїв, навчальні ігри стали ще доступнішими і популярнішими. Вони здатні поєднувати навчання з розвагою, забезпечуючи певну інтерактивність та захоплюючий досвід.

					КвРІПЗ.200177.01.26.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Сучасні навчальні ігри розробляються з використанням різних технологій, таких як: віртуальна реальність, доповнена реальність та інші інноваційні підходи.

Вони можуть включати в себе різноманітні інтерактивні завдання, головоломки, певних віртуальних персонажів та можливість співпраці, а також здатність змагатися з іншими користувачами.

Як можна спостерігати, на даний момент час існує безліч різних навчальних ігор на різноманітну тематику, що використовують для своєї реалізації різні підходи та платформи, тому придумати щось абсолютно нове є достатньо важким, але можливим завданням.

Проте, не зважаючи на кількість таких ігор, можна з впевненістю сказати, що вони не вирішують всіх наявних проблем. Деякі з наявних проблем, що планується виправити в розроблюваному мобільному застосунку є наступні:

– відсутність якогось балансу між розвагою та навчанням. Іноді навчальні ігри можуть надмірно концентруватись на аспекті розваг, не надаючи достатньої кількості важливих навчальних матеріалів, або викликів, що може призвести до втрати основної мети, а саме навчання та зниження його ефективності;

– занадто широкий спектр тем. Деякі навчальні ігри можуть охоплювати багато різних тем, давати змогу вивчати різні предмети, але в результаті може постраждати глибина навчання. Тому концентрація на конкретній предметній області повинна бути більш корисною для учнів.

Окрім проблем самих навчальних ігор, є ще одна, яку б хотілося виділити окремо. Це проблема того, що люди по різних причинах можуть забувати деякі елементарні операції в математиці, або ж не мати змоги швидко їх виконувати, в результаті чого, можуть виникнути різні труднощі.

Нехтування основними математичними навичками може мати досить негативний вплив на розвиток мислення, та загальну математичну грамотність. Тому знання таких елементарних операцій як: додавання, віднімання, множення та ділення чисел є фундаментальними для розв'язання більш складних задач та для розуміння складніших концепцій. Звичайно, що люди з часом можуть

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

забувати або втрачати впевненість у своїх навичках, якщо вони не використовують їх регулярно. Це може бути наслідком віддаленості від теми математики після закінчення школи чи університету, або ж недостатньої практики у повсякденному житті.

Зважаючи на це, розробка навчальних ігор, спрямованих на оновлення та підтримку елементарних математичних навичок, може бути досить корисною. Такі ігри можуть надати досить зручний і можливо захоплюючий спосіб для відновлення своїх вмінь та збереження інтересу до математики в цілому^[2].

Тому саме для вирішення даних проблем і створюється мобільний застосунок, який буде мати назву «MathMania». Для усунення проблеми балансу між розвагою та навчанням в розроблюваному застосунку, планується створити чотири тренувальних рівні, в яких користувач зможе попрактикуватись у виконанні базових математичних операцій.

В цих рівнях користувач повинен при введенні відповіді мати можливість переглянути її правильність. Елемент гри буде реалізований за допомогою рейтингової гри, в якій користувач на основі своїх знань, або ж того, що він встиг повторити під час виконання тренувальних вправ, виконуватиме ті ж самі стандартні математичні операції, проте тепер він це робитиме на швидкість, змагаючись з іншими користувачами.

Проблема з великою кількістю тем буде вирішуватись тим, що застосунок матиме лише одну тему, перевіряючи лише базові знання в математиці.

Проблема того, що люди можуть забути, або ж не досить швидко виконувати деякі елементарні математичні операції вирішуватиметься сама по собі, тому що це і є тим для чого даний застосунок розроблятиметься. Окрім цього, він ще й додатково тренуватиме деякі інші навички, такі як: уважність та моторика пальців. Також вирішення даної проблеми, робить цей застосунок корисним не тільки для його школярів початкових класів, але й для людей старшого покоління.

Відповідно на основі отриманих під час даного дослідження, даних про те, що таке навчальна гра, визначення проблематики даного типу ігор та ряду інших

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

проблем, а також надання варіанту того, як ці проблем можна вирішити в розроблюваному мобільному застосунку, можна приблизно уявити, яким повинен він має бути. Виходячи з усього він повинен:

- забезпечувати певну різноманітність завдань і активностей для підтримки цікавості користувача, уникнення монотонності;
- створювати систему заохочень, яка стимулюватиме користувачів до досягнення кращих результатів і прогресу в грі;
- використовувати естетично приємний дизайн інтерфейсу без яскравих або надто різких кольорів;
- мати інтуїтивно зрозумілий для користувача інтерфейс та правила;
- створювати систему лідерів, що дозволить користувачам змагатися один з одним та порівнювати свої досягнення для стимулювання конкуренції.

Розробка також була висвітлена в тезах Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023»^[3].

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Для того щоб краще зрозуміти те, яким повинен бути майбутній мобільний застосунок, необхідно дослідити додатки на дану тематику та які здатні вирішувати задані проблеми вже існують на даний момент. Це виконується для того, щоб дізнатись про недоліки, які є в даних застосунках і які по можливості необхідно усунути в розроблюваному програмному продукті, а також визначити їх переваги та певні цікаві рішення, що вони надають, які можна використати під час розробки.

Для розгляду було обрано два мобільних застосунки, тематика яких є схожою до того, що розроблятиметься в кваліфікаційній роботі, та які містять функціонал схожий з тим, що планується впровадити в ньому.

Першим розглядався такий мобільний застосунок, як «Math Workout». Він містить досить схожий з задуманим у розроблюваному застосунку функціонал.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

При вході в застосунок користувача зустрічає екран з своєрідним головним меню. В верхній панелі даного додатку міститься його назва, а справа є кнопка у вигляді шестерні, яка відповідає за відкриття екрану з налаштуваннями.

В нижній панелі розміщено три кнопки, які відповідають за такі функції: оцінити застосунок, де йому можна поставити певну кількість зірок; купити про-версію застосунку, що відкриватиме доступ до різного оформлення, яке існує в ньому та блокує всю рекламу.

В центральній частині екрану розміщено кнопки, які відповідають за відкриття екранів з певними завданнями. По кількості даних кнопок можна побачити, що функціоналу в даному застосунку більше ніж того, що планується впровадити в задуманому мобільному застосунку, проте весь функціонал відповідає темі базових математичних операцій.

В екрані налаштувань користувач може виконати наступні операції над додатком: вимкнути звук в ньому; обрати собі складність завдань; вибрати кількість завдань, які будуть відображатись; обрати мову додатку; змінити кольорову тему додатку при наявності про-версії; змінити параметри клавіатури для вводу необхідних значень. Відповідні екрани розглянутого мобільного додатку показано на рисунку 1.1

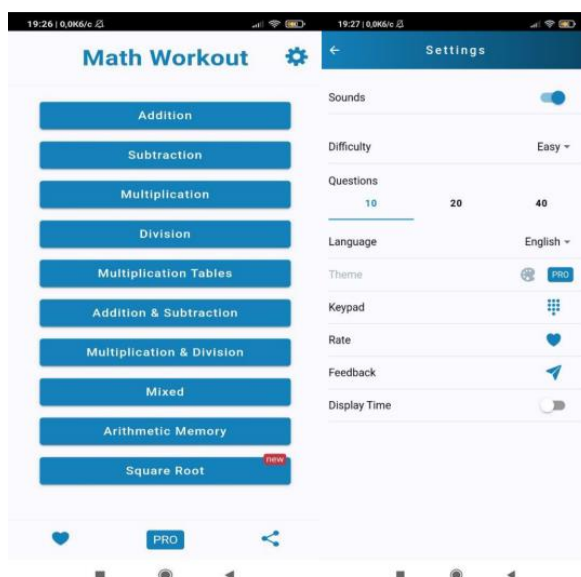


Рисунок 1.1 – Головне меню та екран налаштувань додатка «Math Workout»

					КВРІПЗ.200177.01.26.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Реалізація головного функціоналу даного застосунку, а саме виконання базових операцій з числами, реалізується шляхом вибору самого завдання з його подальшим виконанням в окремому вікні. На головному екрані користувач обирає завдання, яке він хоче виконати, натискає на нього і йому виводиться екран, на якому вказано: назву цього завдання; кількість запитань та їх складність, які можна додатково обрати в екрані налаштувань додатку, також під час роботи з даними завданнями відображається і результат їх виконання.

В нижній частині екрану з кожним завданням розміщено кнопку, яка розпочинає його виконання. При натисканні на неї відкривається екран з таймером після завершення, якого відображається саме завдання. Екран з самим завданням представлений таким чином, що у верхній його частині містяться два поля, в одному з яких, виводиться згенерований випадковим чином приклад, а в іншому відображається відповідь, яку вводить користувач. Введення відповіді відбувається за допомогою спеціальної клавіатури в нижній частині екрану. Відповідні екрани мобільного додатку показано на рисунку 1.2

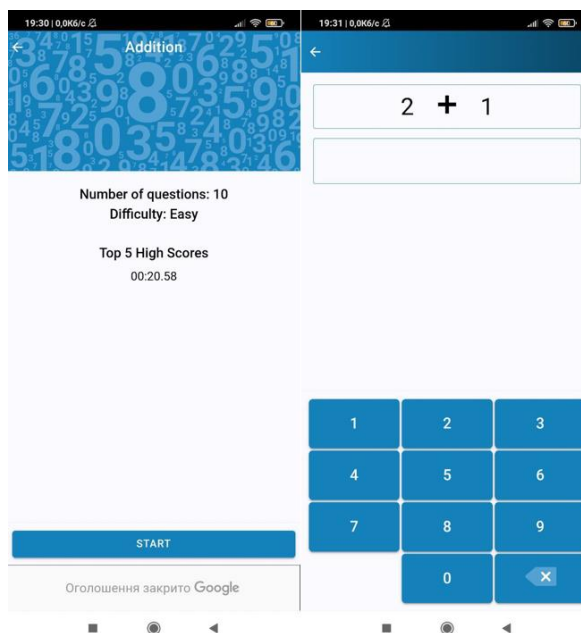


Рисунок 1.2 – Екран результатів та виконання завдання додатку «Math Workout»

Після дослідження даного додатку можна сформулювати те, які основні переваги та недоліки існують в ньому, для того, щоб виявити, що можна

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

використати при розробці свого додатку і чого додавати не варто. До переваг даного мобільного застосунку можна віднести:

- приємний і не перевантажений дизайн, та можливість його зміни;
- простий та зрозумілий інтерфейс;
- здатність змінити мову додатку на більш зручну для користувача;
- різноманітні завдання з числами, які відповідають темі базових математичних операцій, а також охоплює додаткові теми;
- відображення найкращого часу проходження завдань;
- можливість налаштовувати складність завдань та їх кількість.

До основних недоліків даного додатку можна віднести:

- наявність рекламних банерів, які частково закривають вміст, які можна відключити лише за наявності у користувача про-версії;
- доступ до деякого функціоналу заблокований до придбання про-версії, але, оскільки заблокована лише можливість зміни зовнішнього вигляду, то ця проблема не є критичною.

Наступним розглядався такий мобільний застосунок, як «Math Brain». Як і раніше описаний застосунок, він містить досить схожий з задуманим у розроблюваному мобільному додатку функціонал.

При вході, користувача зустрічає екран з певним мотивуючим текстом та кнопкою «Продовжити». Після натискання якої, користувачу виводиться екран, де він може обрати в чому саме він хоче попрактикуватись, а також обрати рівень складності самих завдань, також присутня можливість встановити так зване щоденне нагадування, яке буде відправляти користувачеві повідомлення про те що потрібно пройти тренування.

Окрім цього на даному екрані є кнопка, при натисканні на яку користувачеві буде показаний екран, що представляє собою варіант головного меню. Дані екрану показано на рисунку 1.3

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

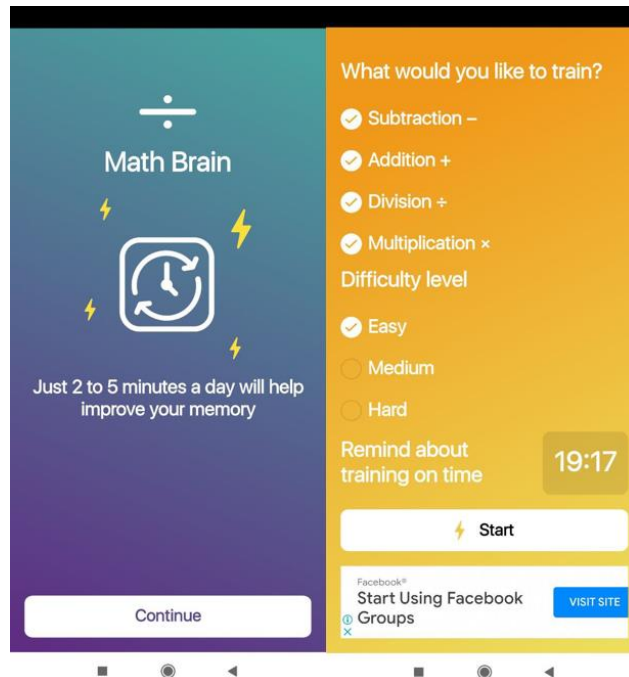


Рисунок 1.3 – Початкові екрани додатку «Math Brain»

Екран головного меню додатку має вигляд спливаючого вікна з трьома кнопками: «Start», «Results», «Settings». Перша кнопка надає доступ до виконання завдань користувачем, друга кнопка дає змогу переглянути індивідуальні результати, третя дає можливість змінювати налаштування.

Екран з самим завданням, яке повинен виконати користувач містить в собі такі елементи як: таймер, приклад, який відображається в центрі екрану і на який користувач повинен дати відповідь, а також під ним показується наступний приклад, на який потрібно буде дати відповідь, також на екрані присутня кнопка, яка дозволить поставити гру на паузу.

Принцип роботи з завданнями можна описати так, що користувачу виводиться сам приклад і замість того, щоб вводити відповідь самостійно, застосунком надається можливість обрати відповідь із запропонованих чисел. Варіанти вибору можна бачити в нижній частині екрану, над ними ж дрібними шрифтом вказується прогрес виконання завдань. Самі приклади, як і відповіді до них, можуть генеруватись випадковим чином, проте враховуючи той факт, що в цих створених випадковим чином відповідях має бути хоча б одна правильна, що практично неможливо, якщо використовувати такий метод.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

І щоб цього уникнути можливі декілька варіантів: правильна відповідь на завдання обчислюється одразу ж, як воно видається і просто вноситься до списку варіантів цієї відповіді, або ж сам приклад, як і відповідь до нього, отримуються з бази даних. Екрани з меню та завданням можна побачити на рисунку 1.4

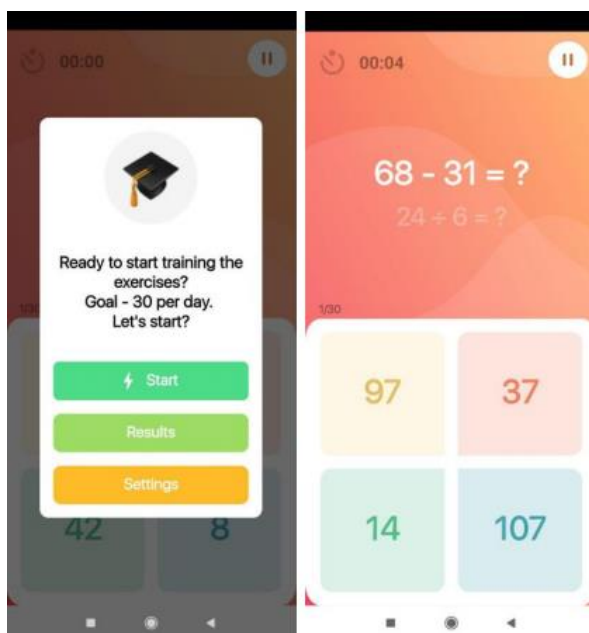


Рисунок 1.4 – Екран меню та завдання в додатку «Math Brain»

Після виконання всіх завдань, користувач має можливість переглянути статистику на екрані, де відображаються результати. Тут він може побачити рівень своїх навичок, графік з кількістю правильних відповідей в різні дні, коли він користувався додатком, що дозволяє слідкувати за своїм прогресом у навчанні. Також з даного екрану можна перейти в таблицю лідерів, в якій користувач може переглянути свою позицію в списку. Для того, щоб визначити порядок користувачів у цьому списку використовується загальна кількість правильних відповідей, які надав користувач при використанні додатку на протязі певного періоду часу.

Після дослідження даного додатку можна виділити його основні переваги та недоліки. До основних його переваг можна віднести:

- досить привабливий дизайн без різких кольорів;
- інтуїтивно зрозумілий інтерфейс;

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- здатність обирати правильну відповідь з уже запропонованих, що може частково полегшити роботу;
- можливість обирати тип завдань, які будуть відображатись користувачу, а також їх складність;
- можливість слідкувати за своїми результатами на протязі всього періоду використання додатку;
- рейтинг лідерів, в якому користувач може поборотись за перше місце.

До недоліків даного додатку можна віднести:

- одне безкоштовне використання на день, щоб отримати необмежений доступ потрібно купити VIP-підписку, або дивитись велику кількість реклами;
- рекламні банери, які частково перекривають вміст екранів, для вимкнення яких, також потрібна VIP-підписка;

Для того щоб підсумувати дослідження існуючих рішень можна створити таблицю, яка міститиме порівняння цих мобільних додатків. Дане порівняння показано у таблиці 1.1

Таблиця 1.1 – Порівняльна таблиця знайдених рішень

Критерій	Math Workout	Math Brain
Інтерфейс	Простий, інтуїтивно зрозумілий	Середній, інтуїтивно зрозумілий
Дизайн	Простий без використання різких кольорів	Хороший з використанням різних кольорів
Досягнення цілі	Добре досягає основної цілі	Добре досягає основної цілі
Переваги	Відповідь можна ввести самому	Вибір відповіді дається з певного набору
Недоліки	Рекламні банери	Необхідність VIP-підписки

1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення

Перш ніж розпочати розробку програмного забезпечення, важливо чітко визначити та описати всі його необхідні функції. Це робиться з тією метою, щоб краще зрозуміти майбутні вимоги до технічного завдання, які в подальшому необхідно буде сформулювати.

Для початку необхідно відобразити, як необхідні для роботи дані будуть переміщуватись в розроблюваному застосунку. Зробити це можна використовуючи DFD-діаграму, на якій буде показано, які основні функції може виконувати користувач в мобільному застосунку, як ці функції пов'язані між собою, а також те, як вони взаємодіють з базою даних. DFD-діаграма подана на рисунку 1.5

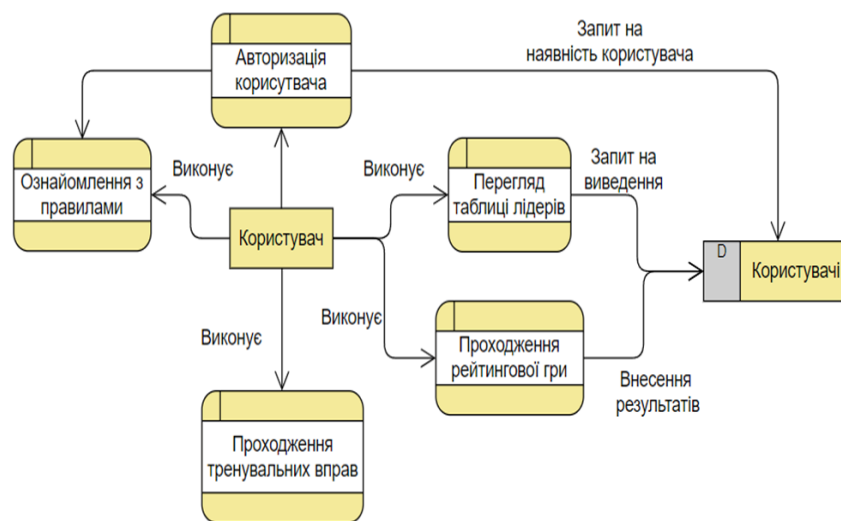


Рисунок 1.5 – DFD Діаграма

Для більш хорошого розуміння того, які ролі та процеси виконуються, можна використовувати різні діаграми для представлення їх у візуальному вигляді. Щоб визначити вимоги до розроблюваного програмного продукту, було створено діаграму варіантів використання. Для цієї діаграми було описано всіх

Закінчення таблиці 1.3

Актор	Варіант використання	Опис
	Отримання завдання	Користувач отримує завдання для розв'язку
Зареєстрований користувач	Введення відповіді	Користувач вводить відповідь на отримане завдання
	Натиснення кнопки для перевірки	Користувач перевіряє чи відповідь яку він ввів є правильною

Тепер, маючи більш детальний опис того, які актори існують для розроблюваного мобільного застосунку, які для них є варіанти використання, тобто те, що вони можуть виконувати в даній системі та дано трішки детальніший їх опис, необхідно відобразити це все на відповідних діаграмах варіантів використання. Дані діаграми показано на рисунку 1.6 та рисунку 1.7.

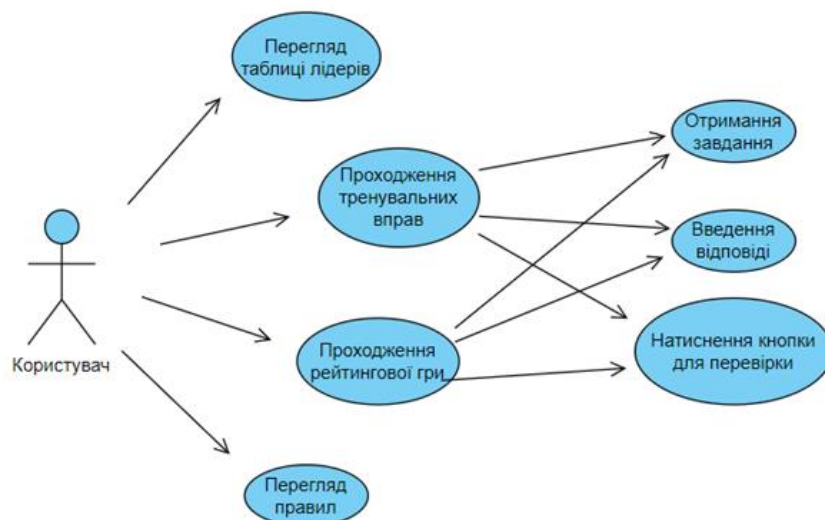


Рисунок 1.6 – Діаграма варіантів використання

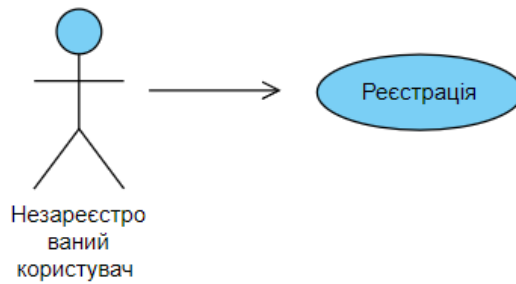


Рисунок 1.7 – Діаграма варіантів використання

Основний функціонал, який повинен містити розроблюваний мобільний застосунок поданий в даних варіантах використання та діаграмах до них. В подальшому можливе його доповнення додатковими функціями, наприклад додання таймера для виконання завдань, або повідомлення про різні події. Проте основний функціонал повинен залишитись незмінним.

Ще одним важливим фактором є не тільки те, що повинна робити система, а й те, яким чином вона повинна працювати. Для цього необхідним є визначити основні нефункціональні вимоги до розроблюваного програмного забезпечення. В результаті проведених досліджень можна зробити висновок про існування таких вимог:

- застосунок повинен працювати швидко та безперебійно, навіть при великій кількості користувачів та високих навантаженнях на сервер;
- інтерфейс застосунку повинен бути легким та інтуїтивно зрозумілим, щоб користувачі могли легко зорієнтуватися та користуватися додатком без підготовки;
- мобільний застосунок повинен бути доступним для користувачів з різними рівнями здібностей та можливостей, включаючи людей з обмеженими можливостями;
- програмне забезпечення повинно бути локалізований українською;
- застосунок повинен бути придатним для використання в різних контекстах та для різних цільових груп користувачів, наприклад, в навчанні, розважальних цілях або для розвитку когнітивних здібностей;

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

– застосунок повинен бути стійким до помилок та збоїв, щоб забезпечити коректну роботу та надійність для користувачів.

З додатковими відомостями можна ознайомитись в додатку А.

1.4 Висновки. Постановка задачі

По результатам проведених досліджень можна зробити висновки про те, що з себе повинен представляти розроблюваний програмний застосунок, тобто що він повинен робити, яким чином він повинен це робити, а також те який вигляд він повинен мати. Розроблюване програмне забезпечення повинно надавати користувачам весь необхідний функціонал для того щоб вони могли виконувати поставлені перед ними задачі. В даному застосунку за ці задачі взято вдосконалення навичок в математиці, розвиток певних когнітивних функцій, наприклад моторику пальців, а також уважність.

Враховуючи основну цільову аудиторію для якої повинен розроблятися програмний застосунок, а в даній кваліфікаційній роботі визначено, що даною цільовою аудиторією є учні початкових класів які повинні мати знання, необхідні для того, щоб зрозуміти для чого призначений застосунок.

Відповідно з урахуванням цільової аудиторії даного застосунку та їх потреб, визначені його функціональні та нефункціональні вимоги, на основі яких в подальшому потрібно буде розробити архітектуру програмного забезпечення та прийняти ряд інших рішень, які допоможуть досягти поставлених цілей.

Метою даної кваліфікаційної роботи відповідно є розробити програмне забезпечення, яке представлятиме собою мобільний застосунок, який зможе надати учням початкових класів можливість вдосконалити свої математичні навички, розвинути певні когнітивні функції (зокрема, моторику пальців) та увагу. Він повинен бути розроблений з урахуванням всіх визначених на етапі аналізу функціональних та нефункціональних вимог. На їх основі потрібно обрати найкраще архітектурне рішення для розробки логіки програмного забезпечення, розробити зручний інтерфейс та обрати інструменти для реалізації.

					КвРІПЗ.200177.01.26.ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Архітектура та функціональна структура застосунку

На даному етапі необхідним є детальний розгляд архітектури та функціональної структури мобільного застосунку, що представляє собою роду тренажер, який використовує свого роду інтерактивні ігрові елементи. Робота спрямована на розробку зручного та ефективного інструменту для вивчення математики, який стимулює користувачів до активного навчання шляхом використання ігрових елементів.

Оскільки, у розроблюваному, в даній кваліфікаційній роботі, мобільному застосунку не планується якась складна структура, а також не планується значним чином розширяти його функціонал в майбутньому, то вся архітектура даного застосунку буде розроблюватись на основі такого шаблону проектування для мобільних застосунків як MVC^[4].

Шаблон MVC (Model-View-Controller) є одним з популярніших підходів до проектування програмного забезпечення, включаючи мобільні застосунки. Він надає змогу відділити логічну частину застосунку від його візуального представлення та управління взаємодією з користувачем. Основними компонентами даного шаблону є:

Модель (Model):

- представляє собою бізнес-логіку або дані застосунку;
- відповідає за обробку даних, виконання операцій та зберігання інформації;
- не залежить від інтерфейсу користувача і може використовуватися у різних частинах застосунку.

Представлення (View):

- відповідає за візуальне представлення даних користувачу;
- може бути екраном застосунку, інтерфейсом для взаємодії з користувачем, або будь-яким іншим візуальним вікном;

					КвРІПЗ.200177.01.26.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

– отримує дані від моделі та відображає їх користувачеві.

Контролер (Controller):

– відповідає за обробку вхідних подій від користувача, а також за взаємодію між моделлю та представленням;

– приймає вхідні дані від користувача, обробляє їх та виконує відповідні дії, які можуть включати звернення до моделі або оновлення представлення.

Шаблон MVC спрощує створення додатків, які працюють на різних платформах і мають просту систему навігації. Він стандартизує дизайн інтерфейсу та забезпечує прямолінійний користувацький досвід.

Однак, шаблон MVC може мати деякі недоліки в порівнянні з іншими шаблонами, серед яких:

– у великих застосунках, контролери можуть стати дуже великими та складними, оскільки вони слугують мостом між моделлю та представленням.

– тестування може бути досить проблемним, оскільки контролери тісно пов'язані з представленнями.

– даний шаблон може бути менш гнучким для деяких сценаріїв, особливо коли потрібно багато взаємодії між представленнями.

Тому для вирішення деяких з цих проблем і були розроблені такі шаблони проектування, як MVVM або MVP. Але як і зазначалось раніше, оскільки для розроблюваного мобільного застосунку не планується використовувати складну структуру, а також, не планується значне розширення в майбутньому, то шаблону MVC має бути більш ніж достатньо для реалізації поставлених задач.

Слуга (Servant) – шаблон проектування, який дозволяє визначити спільний функціонал для різних ієрархій класів. Робиться це для того, щоб не дублювати код, а винести його у функцію чи спільний клас^[5]. В застосунку «MathMania» даний шаблон можна використати для того, щоб зменшити кількість коду у більшості класів. Оскільки всі завдання матимуть схожу суть, а саме генерацію випадкових цілих чисел і різні математичні операції над ними, а також перевірку того чи ввів користувач вірні дані, щоб не прописувати один і той самий код декілька разів доцільно винести його в окремий клас.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Для показу функціональної структури мобільного застосунку, яка буде вказувати на основні його елементи, які використовують прийняті архітектурні рішення. Можна використати схему, яка буде показувати, як компоненти даного мобільного застосунку, які були описані в архітектурних рішеннях взаємодіятимуть між собою так і які функції вони будуть виконувати. Дану схему можна побачити на рисунку 2.1

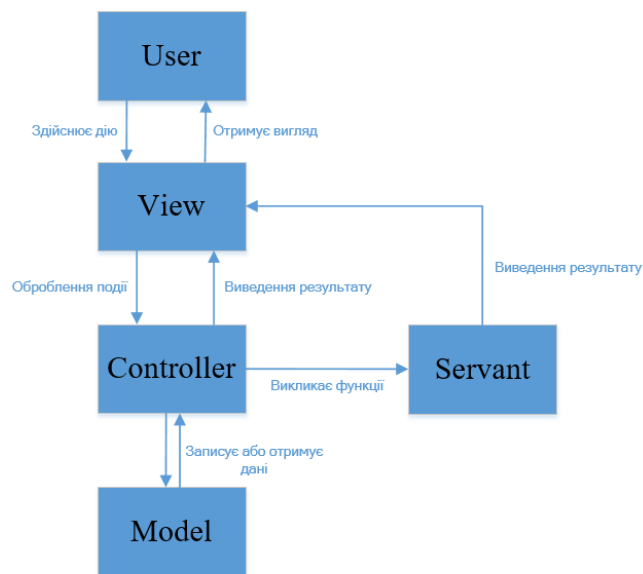


Рисунок 2.1 – Функціональна структура застосунку

На даній схемі можна відповідно побачити, як повинні взаємодіяти основні компоненти мобільного застосунку. Спершу користувач повинен виконати будь-яку дію над View на своїй стороні, в свою чергу Controller буде реагувати на цю дію і здійснювати запит до моделі на те, щоб записати в неї якісь дані, або отримати їх. Також в залежності від дії Controller буде викликати певні функції, які знаходяться в Servant, які будуть обробляти дані з моделі і виводити результат їх виконання у View, який буде відображати їх користувачу. Якщо ж Controller не викликати функцій Servant, то він сам буде виводити результат виконання дій для користувача.

Відповідно до всього цього View представлятиме собою графічний

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

інтерфейс з яким зможе взаємодіяти користувач, і в якому буде отримувати результати виконання своїх дій.

Controller в свою чергу представлятиме собою функції, які будуть реагувати на різні дії, які виконує користувач, наприклад натискає на кнопку чи запускає сам застосунок. І буде виводити результати, які він отримає від Model, а також в залежності від дії записувати дані в неї. Model буде представляти собою, як моделі для зв'язку з базою даних, так і всі атрибути, в яких будуть зберігатися дані, які потім повинен обробити Controller.

Servant повинен містити в собі функціонал, який спростить роботу Controller, зменшить кількість коду в застосунку, шляхом винесення деяких його функцій в нього. Також він, як і компонент, який відповідає за обробку подій, буде виводити результати виконання свої функцій в компонент, який відповідає за виведення даних користувачу.

2.2 Проектування структури бази даних

Оскільки, як було описано в попередніх розділах, де визначались вимоги до того, яким повинен бути застосунок, та що він повинен робити, однією з вимог була можливість змагання між користувачами. Вона мала бути забезпечена шляхом додавання своєї таблиці лідерів, в якій би відображалась кількість набраних балів для кожного користувача. Для того, щоб забезпечити таку можливість, необхідно розробити для мобільного застосунку базу даних, яка б зберігала в собі дані всіх авторизованих користувачів, а також зберігала дані про їх досягнення під час проходження певних елементів даного застосунку.

Для розроблення таблиць в даній базі даних спершу необхідно виділити, які саме дані повинні в них зберігатись. Враховуючи попередні вимоги, цими даними будуть відповідно ім'я користувача, пароль та рахунок, який представлятиме кількість правильно введених відповідей. Враховуючи той факт, що дані про ім'я

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача планується взяти, як унікальне для ідентифікації користувача, то між вказаними даними виникнуть наступні залежності: ім'я користувача → пароль, ім'я користувача → рахунок. Як можна побачити з цього, всі ці дані можна записати в одну таблицю бази даних, але для того щоб відокремити пароль користувача від його рахунку, було прийнято рішення розділити дану таблицю на дві окремі^[6].

В таблиці «Користувачі» поле ім'я користувача буде ключовим, і саме за ним буде ідентифікуватись та авторизуватиметься користувач, тоді як, поле пароль в даній таблиці, слугуватиме лише для авторизації користувача в додатку. Дані поля з їх типами та обмеженнями можна побачити в таблиці 2.1

Таблиця 2.1 – Користувачі

Назва поля	Тип поля	Обмеження	Коментар
Name	VARCHAR(20)	NOT NULL	Ім'я користувача, та його ідентифікатор
Password	VARCHAR(20)	NOT NULL	Пароль

У таблиці "Лідери" ім'я користувача та рахунок, обидва є допоміжними полями, які використовуються для знаходження записів та відображення досягнень кожного гравця. Ця таблиця представляє рейтинг користувачів у грі, де рахунок відображає набрані бали або досягнення. Вона сприяє конкуренції між гравцями за місця в рейтингу, що стимулює їх активну участь у грі та взаємодію з розроблюваним застосунком.

Таблиця 2.2 – Лідери

Назва поля	Тип поля	Обмеження	Коментар
Name	VARCHAR(20)	NOT NULL	Ім'я користувача
Score	INT	NOT NULL	Рахунок користувача

2.3 Проектування інтерфейсу користувача

На даному етапі необхідним є визначити, який вигляд повинен мати розроблюваний мобільний застосунок. Тобто те, як в ньому повинні бути розміщені різні елементи, такі як: кнопки, поля для введення та виведення даних. Також потрібно визначити, який дизайн слід використати при його створенні. Для цього використовуються раніше описані вимоги, як функціональні, так і не функціональні, а також те, для якої цільової аудиторії розробляється даний мобільний застосунок.

Оскільки цільовою аудиторією даного застосунку в основному виступають учні початкової школи, то його дизайн та інтерфейс повинні бути відповідними. За результатами попередніх досліджень, інтерфейс розроблюваного мобільного застосунку також повинен надавати можливість користувачу виконувати всі перелічені в функціональних вимогах задачі.

Першою такою задачею виступає можливість для користувача зареєструватися або авторизуватися в мобільному застосунку для подальшої роботи з ним. Оскільки цільова аудиторія – це діти початкової школи, то екран, який буде відповідати за виконання даної задачі не повинен містити багато різних елементів, тобто не бути перевантаженим^[7].

Для цього буде достатньо додати на екран два поля для введення, в які користувач буде вводити своє ім'я користувача та пароль для авторизації чи реєстрації в застосунку, а також відповідно дві кнопки, які б дозволяли запустити процедури авторизації та реєстрації з необхідними їм перевітками та які відповідно відкриватимуть йому доступ до виконання наступних задач.

Також можна додати на екран назву даного мобільного застосунку. Макет даного екрана з приблизним розташуванням на ньому вказаних раніше елементів можна побачити на рисунку 2.2.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

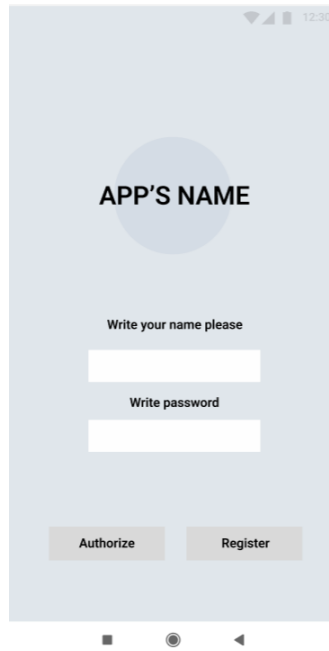


Рисунок 2.2 – Макет екрану авторизації

Для авторизованого користувача відкривається можливість виконувати завдання, які доступні тільки для нього, тобто проходити тренувальні вправи, рейтингову гру, переглядати правила використання мобільного застосунку та таблицю лідерів. Відповідно для того, щоб мати можливість перейти до виконання даних завдань необхідно розробити екран, на який потраплятиме користувач після успішної авторизації в застосунку. Тобто екран своєрідного головного меню. На цьому екрані потрібно буде розмістити, як і на попередньому назву мобільного застосунку, а також кнопки, які відповідатимуть за те, щоб переправити користувача на інші екрани, де він зможе виконати решту завдань. Кнопки для переправлення користувача на екрани з таблицею лідерів та правилами можна розмістити у верхній частині екрана з меню для того, щоб відділити їх від інших.

Під ними, відповідно, потрібно розмістити групу кнопок, кожна з яких буде відправляти користувача на відповідну тренувальну вправу, або ж в рейтингову гру. Між цією групою кнопок та попередніми кнопками можна розмістити текстове поле, в яке потім буде виводитись певний текст. Макет даного екрана можна побачити на рисунку 2.3.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

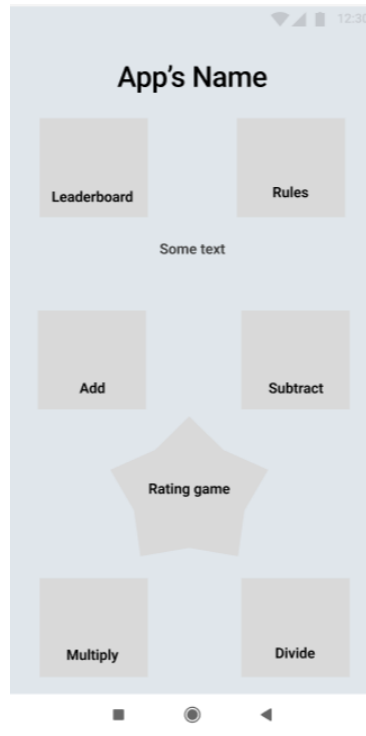


Рисунок 2.3 – Екран головного меню

Далі необхідно розробити інтерфейс екранів, які відповідатимуть за те, щоб користувач міг виконати завдання з отриманням задач, які йому необхідно розв'язати, ввести відповідь на нього та отримати результат перевірки. Для цього розроблятимуться екрани, які будуть відповідати за виконання даних задач. Загалом це буде 4 екрани, які будуть відповідати за виконання тренувальних вправ і один екран, який відповідатиме за проходження рейтингової гри. Ці екрани по своєму інтерфейсу будуть схожі, але міститимуть деякі відмінності. Для чотирьох екранів тренувальних вправ, необхідно розмістити певне текстове поле, в яке буде виводитись отримане користувачем завдання для розв'язку, а також поле для введення відповіді, яке потрібно розмістити біля нього. Під цими полями потрібно розмістити дві кнопки, які будуть відповідати за перевірку на правильність введеної відповіді, та можливість повернутись на екран з меню. Над текстовими полями можна розмістити поля, які відповідатимуть за відображення таймера, опису тренувальної вправи та її назву. Екран з рейтинговою грою буде схожим на них з відмінністю лише в відсутності кнопки, яка повертає користувача в меню. Макети даних екранів можна побачити на рисунку 2.4

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

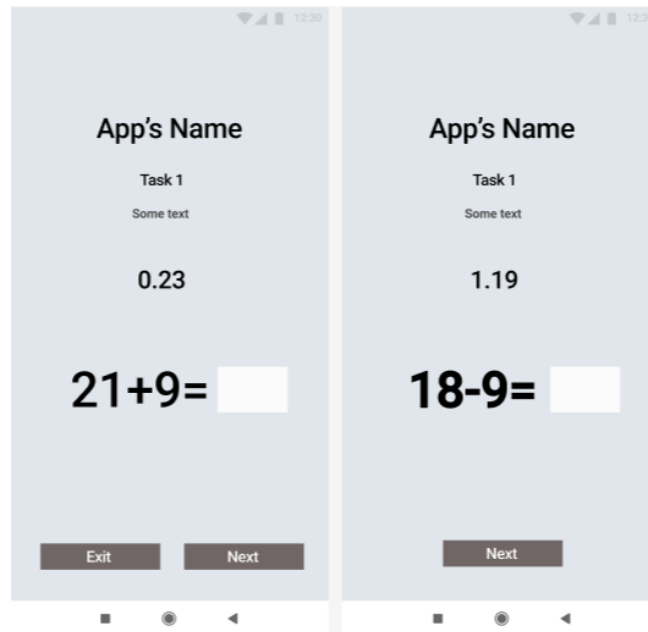


Рисунок 2.4 – Макети екранів для виконання математичних задач

Останніми екранами, інтерфейс, яких необхідно розробити – це екрани, які відповідатимуть за відображення користувачеві таблиці лідерів, де він зможе переглянути своє місце в ній, а також відображення правил використання мобільного застосунку. На екрані, який відображатиме таблицю лідерів потрібно розмістити місце, в якому буде розміщатись список з гравцями, та їх місцями в загальному рейтингу. Над цим місцем можна розмістити назву екрана, а також те, скільки саме користувачів буде виводитись у відведеному місці. Під ним також необхідно розмістити кнопку, при натисканні на яку користувач зможе повернутись на екран з головним меню.

Екран з правилами, як і екран з таблицею лідерів повинен містити в собі певне місце, в якому будуть виводитись відповідно правила користування застосунком. Як і в попередньому екрані, на ньому потрібно розмістити кнопку для переходу назад до головного меню. Оскільки дані екрани відповідають лише за виведення певної інформації, то їх макети хоч і необхідно створити, проте на даному етапі потреби в їх відображенні немає, тому те, як будуть виглядати дані екрани буде показано дещо пізніше.

2.4 Розроблення алгоритму роботи мобільного застосунку

Після визначення того, що розроблюваний мобільний застосунок повинен робити, а також того, який інтерфейс він повинен мати, щоб користувач міг взаємодіяти з ним для виконання поставлених перед ним задач, наступним стає крок, на якому потрібно описати те, яким чином це повинно відбуватись, тобто необхідно створити алгоритми роботи мобільного застосунку.

Для початку необхідно описати алгоритм того, як користувач буде авторизуватись в програмі, для чого необхідно використати таку UML-діаграму, як діаграма діяльності^[8]. Вона дозволить вказати порядок дій, які будуть виконуватись при певних умовах. Дану діаграму можна побачити на рисунку 2.5

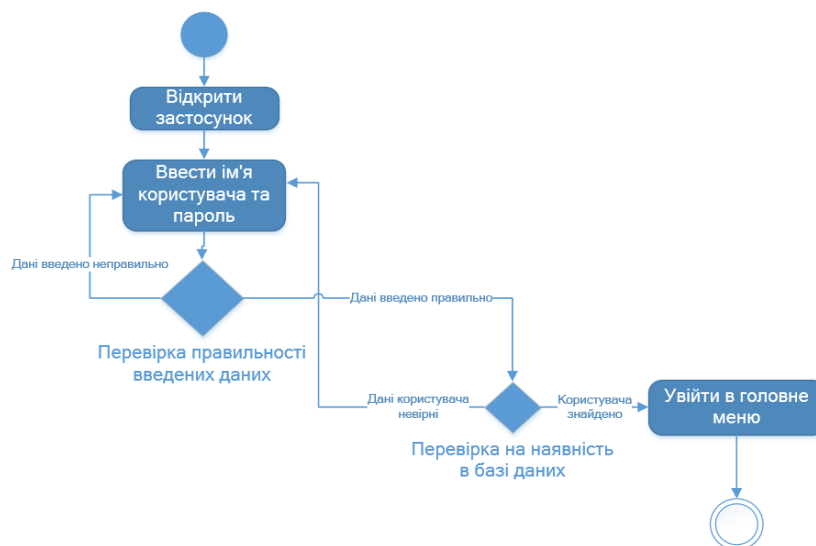


Рисунок 2.5 – Діаграма діяльності «Авторизація користувача»

Як видно з даної діаграми користувачеві спершу необхідно відкрити сам мобільний застосунок, після чого йому буде показано екран з який відповідає за авторизацію користувача. Далі йому необхідно ввести необхідні для авторизації дані та натиснути відповідну кнопку. Далі відбуватиметься перевірка того, чи введені дані є коректними, та в залежності від результатів даної перевірки будуть виконані подальші дії, відбудеться наступна перевірка, або ж користувачеві

потрібно буде ввести правильні дані і повторити процес знову. Якщо перша перевірка пройшла успішно, то наступна буде перевіряти те, чи користувач, який ввів свої дані, записаний в базі даних. Якщо він не буде знайдений, то можливі два випадки, користувач ввів свої дані неправильно або ж його взагалі немає в базі даних. В першому випадку його потрібно буде попросити ввести правильні дані, а от в іншому, потрібно попросити його зареєструватись натиснувши відповідну кнопку на екрані. Якщо ж користувач був знайдений в базі даних, то йому потрібно вивести екран з головним меню звідки він зможе обрати з чим працювати далі.

Реєстрація в застосунку також може проходити по даному алгоритму, тільки перевірка на наявність в базі даних буде іншою, тобто, якщо в базі даних буде знайдено користувача з введеними даними, то потрібно буде ввести інші, якщо ж ні його буде додано в неї і переведено на екран головного меню.

Наступним буде доцільно описати алгоритм виконання однієї з математичних завдань в даному застосунку. Нехай буде ситуація, коли користувач хоче потренуватись в додаванні чисел. Дії які йому необхідно виконати показано на рисунку 2.6.

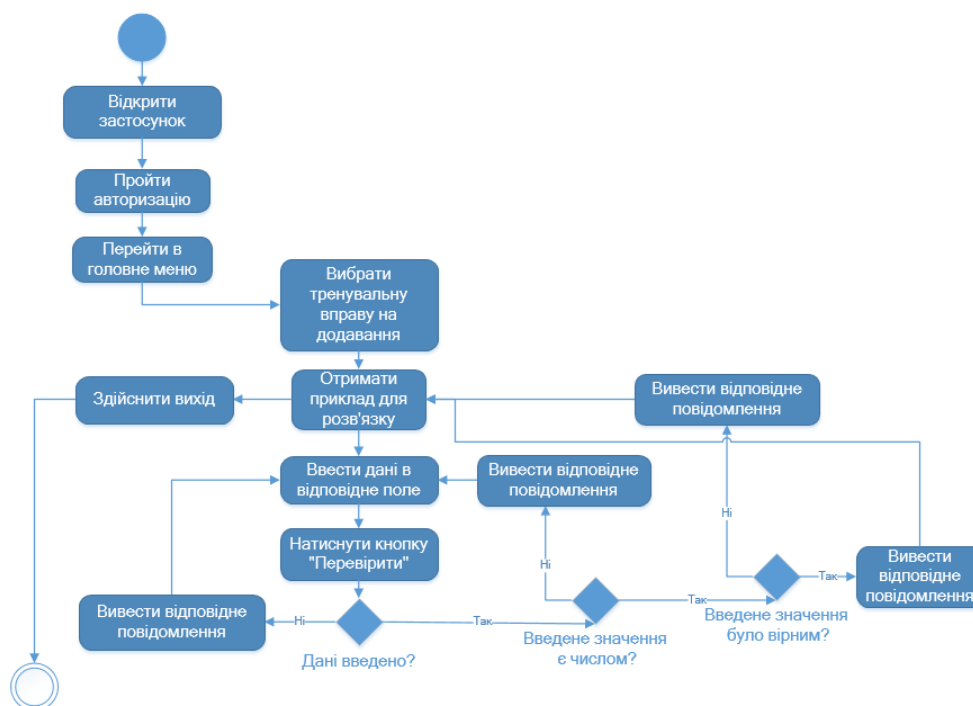


Рисунок 2.6 – Діаграма діяльності «Проходження тренувальної вправи на додавання чисел»

Для того, щоб виконати проходження тренувальної вправи на додавання чисел користувачу спершу потрібно відкрити сам застосунок і відповідно пройти авторизацію або реєстрацію по раніше поданому алгоритму, та після того, як йому буде виведений екран з головним меню потрібно буде натиснути кнопку, яка відповідатиме за те, щоб відобразити екран з необхідною для нього задачею.

Далі користувач має отримати завдання, яке йому необхідно розв'язати за певний відведений таймером час. Для цього йому потрібно ввести свою відповідь у спеціально призначене для цього поле та натиснути кнопку для перевірки правильності введеної їм відповіді. При натисканні на дану кнопку відбуватиметься ряд перевірок, перша з яких буде перевіряти те, чи ввів користувач відповідь, якщо він цього не зробив, то йому потрібно про це повідомити і попросити ввести відповідь. Якщо ж він все-таки ввів відповідь, то виконується наступна перевірка на те, чи є введене значення числом, якщо ні, то вивести повідомлення з відповідним вмістом і попросити його ввести числове значення, якщо значення, яке він ввів є числом, то відбувається остання перевірка. Вона якраз і перевіряє чи є введена користувачем відповідь правильна використовуючи певний алгоритм перевірки. Якщо його відповідь буде неправильною, то потрібно вивести повідомлення про це, в якому також необхідно вказати, яка відповідь була правильною, після чого видати користувачеві новий приклад для розв'язку. Якщо відповідь введена користувачем була правильною йому також буде виведено повідомлення з відповідним вмістом, після якого він отримає нове завдання для розв'язку. Даний алгоритм буде повторюватись стільки разів скільки захоче сам користувач і для того, щоб завершити виконання завдання йому потрібно натиснути на кнопку, яка переведе його назад на екран головного меню.

Даний алгоритм можна використати до всіх інших тренувальних вправ, так як вони є досить схожими по принципу своєї роботи. Алгоритм для завдання проходження рейтингової гри буде схожим на алгоритм проходження тренувальних вправ з відмінністю в тому, що не буде кнопки для повернення на екран головного меню, та не буде виведення повідомлень про правильність чи

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

неправильність введеної відповіді.

2.5 Створення прототипу мобільного застосунку

На основі створених раніше макетів мобільного застосунку, а також алгоритмів його роботи можна створити прототипи того, як він повинен виглядати, а також схему його роботи використовуючи схему навігації по екранам. Для розробки прототипів екранів як і для створення макетів використовуватиметься такий інструмент як Figma^[9].

Figma – це потужний інструмент для дизайну та створення прототипів, який дозволяє розробникам та дизайнерам створювати інтерфейс користувача для веб-сайтів, мобільних застосунків та графічного дизайну. Однією з його головних переваг є хмарна база, що дозволяє користувачам працювати над проектами з будь-якого пристрою та спільно редагувати їх у реальному часі. Його інтерфейс є інтуїтивно зрозумілим, але водночас він може запропонувати досить потужні можливості. Даний інструмент містить досить широкий набір інструментів для створення векторних графічних елементів, макетів і прототипів.

Крім того, Figma підтримує відкриті стандарти, що робить його сумісним з іншими програмами та сервісами, такими як Sketch або Adobe XD. Як було зазначено раніше, однією з ключових особливостей даного інструменту є можливість спільної роботи над проектами в реальному часі, під час якої користувачі можуть одночасно переглядати, редагувати та коментувати дизайн, що дозволяє ефективно співпрацювати над проектами у великих командах або навіть вести дистанційну роботу. Завдяки своїй гнучкості та функціональності, Figma став важливим інструментом для дизайнерів та розробників по всьому світу, які шукають зручний і спільний підхід до створення високоякісних інтерфейсів користувача та прототипів.

Самі прототипи для мобільного застосунку будуть розроблятися на основі раніше створених макетів графічного інтерфейсу користувача, з використанням

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

приємного дизайну в різних відтінках синього та блакитного, а також жовтого кольору. Прототипи екранів для раніше розроблених макетів мобільного застосунку подано на рисунку 2.7.

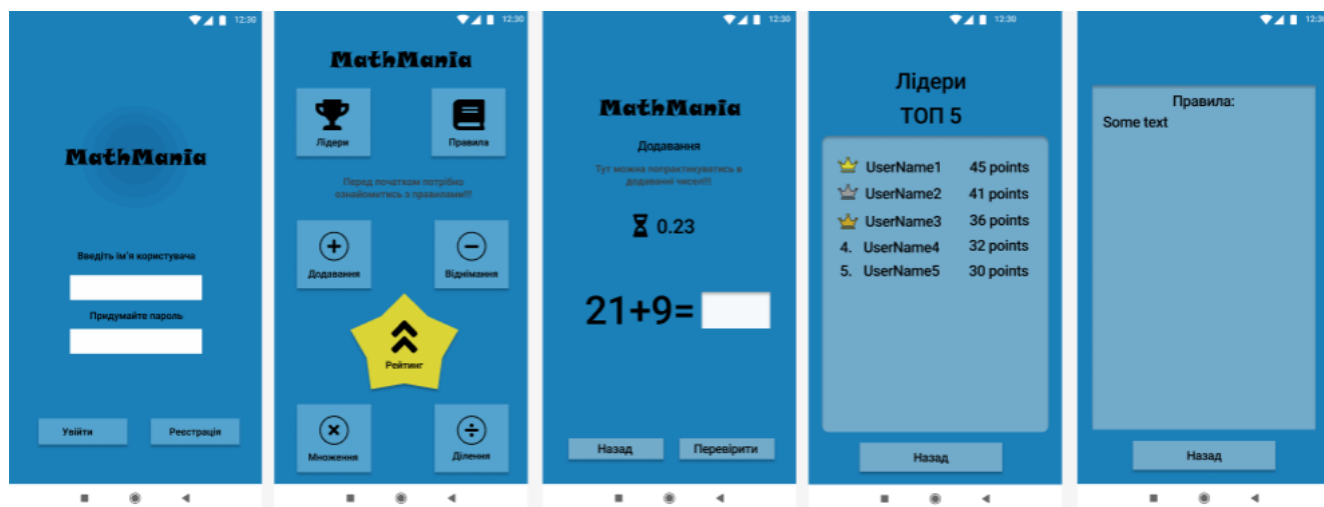


Рисунок 2.7 – Прототипи екранів мобільного застосунку

Як можна побачити на рисунку, вище всі прототипи основних екранів розроблюваного мобільного додатку відповідають розробленим раніше макетам графічного інтерфейсу користувача. Всі елементи знаходяться у спеціально відведених їм місцях, всім текстам задані шрифти, які повинні приваблювати цільову аудиторію, кольори кнопок відрізняються від кольору самих екранів для акцентування уваги користувача на них.

Прототип, який представляє собою екран головного меню мобільного застосунку, містить у своїх кнопках певні піктограми, які разом з текстом під ними вказують на те, які саме дії вони виконують. Кнопка, яка відповідає за відображення користувачу екрана з рейтингової грою відрізняється від інших кнопок, як кольором так і своєю формою, що робиться для акцентування уваги користувача на ній.

На екрані для виконання завдань як і було описано на етапі створення макетів розміщено текстові поля які вказують відповідно на назву завдання, його короткий опис та таймер, який містить поблизу спеціальну піктограму для того щоб вказати за що відповідає дане поле.

Також тут з'явилися прототипи інших розроблених макетів які не

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

показувались раніше, проте як і було описано для них на цих прототипах створені місця, в яких будуть відображатись необхідні дані, а також кнопка для того, щоб повернутись на екран з головним меню.

Після створення даних прототипів, використовуючи описані раніше алгоритми роботи мобільного додатку, необхідним є показати, як дані екрани взаємодіють між собою, для чого можна скористатись навігаційною схемою між екранами застосунку. В середовищі Figma дана схема навігації матиме вигляд який подано на рисунку 2.8.

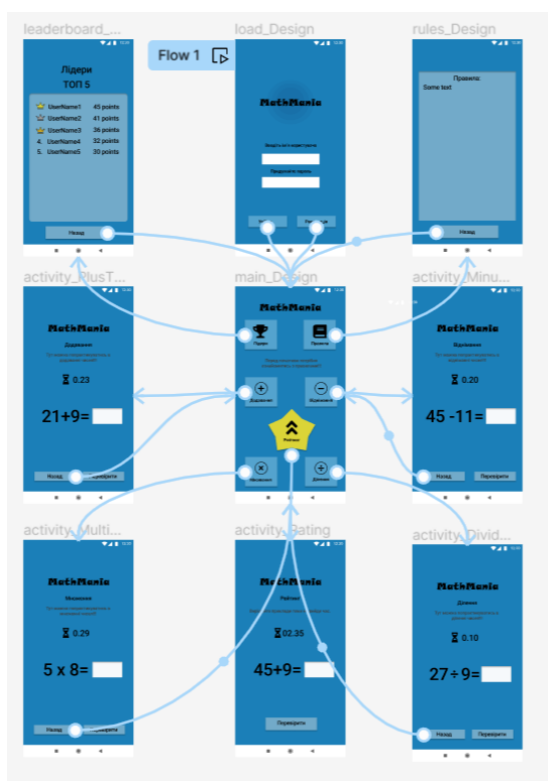


Рисунок 2.8 – Схема навігації застосунку

Як можна побачити з даної схеми навігації всі дії відповідають описаним раніше алгоритмам. Користувач спершу бачить екран з полями для введення даних для реєстрації чи авторизації, далі йому виводиться екран з головним меню звідки він може переміститись на будь-який з решти екранів в залежності від натиснутої ним кнопки і, відповідно, майже з кожного з них повернутись назад в екран з головним меню. При натисканні ж іншої кнопки на деяких з них відбуватиметься виклик самих цих екранів, але вже з іншими значеннями.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

2.6 Аналіз та вибір технологій і методів реалізації застосунку

Після визначення функціональних та нефункціональних вимог та після створення на їх основі всіх необхідних макетів та прототипів, визначення того, яким чином розроблюваний мобільний застосунок повинен працювати, необхідним є визначити технології та методи, які можна використати для його подальшої реалізації.

Спершу необхідно визначитись, для якої операційної системи буде розроблятися застосунок. Вибір здійснювався між двома операційними системами iOS (від Apple) та Android (від Google)^[10].

В iOS основними перевагами для розробки мобільних застосунків є її оптимізація, оскільки Apple виробляє як апаратне, так і програмне забезпечення, iOS може бути більш оптимізованим, що призводить до кращої продуктивності та ефективності роботи застосунків. Також через невелику кількість моделей пристроїв, iOS дозволяє розробникам швидше оновлювати та тестувати застосунки для різних пристроїв^[11].

До недоліків даної операційної системи можна віднести обмежену свободу в розробці через жорстку політику контролю якості та затвердження застосунків у App Store. Також до суттєвих недоліків можна віднести відсутність значної кількості налаштувань для користувачів та розробників^[12].

В Android основними перевагами є те, що дана операційна система відома своєю відкритістю та гнучкістю, що дозволяє розробникам мати більше контролю над функціональністю та виглядом своїх додатків, а також те, що вона надає користувачам більше можливостей для налаштування та персоналізації пристроїв та застосунків. До основних недоліків даної операційної системи можна віднести її фрагментацію, що може призвести до різних складнощів у тестуванні та оптимізації застосунків^[13].

Для розробки мобільного застосунку було обрано операційну систему Android саме через її відкритість та гнучкість, а також через те, що iOS не для всіх своїх версій дає можливість встановлювати сторонні додатки, які не пройшли

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

верифікацію в AppStore, або не було отримано спеціальних дозволів для завантаження його в інший спосіб.

Оскільки за операційну систему, для якої розроблятиметься застосунок було обрано Android, то серед мов програмування, які використовуються для створення такого типу програмного забезпечення, було розглянуто різні варіанти, серед яких були C#, JavaScript, C++, Python, Java та Kotlin. Більшість з цих мов працюють з мобільними застосунками тільки через свої, спеціально призначені для цього фреймворки, через що може бути проблемно налаштувати їх для зручної розробки. Тому вибір в основному робився між двома мовами, які є офіційними мовами програмування для Android, а саме між Java та Kotlin.

До основних переваг Java можна віднести те, що дана мова програмування має досить широке поширення і велика кількість ресурсів, оскільки, це одна з найпопулярніших мов програмування з великою кількістю документації, курсів і інструментів, доступних для розробників. Ще однією великою перевагою є те, що вона має багату екосистему та стабільну підтримку з боку Google^[14].

До недоліків ж можна віднести те, що в порівнянні з деякими сучасними мовами, такими як: Kotlin, Java може мати більшу кількість шаблонного коду для досягнення тих самих результатів. Також до значних недоліків можна віднести те, що використовується зайвий код для простих функцій, та меншу продуктивність порівняно з її конкурентами^[15].

Переваги Kotlin ж полягають в тому, що вона більш універсальна, оскільки, вона може бути використана як для розробки мобільних додатків, так і для розробки серверної частини додатків, що дозволяє розробникам використовувати одну мову для обох частин додатку, та її компактний синтаксис оскільки вона дозволяє писати менше коду для досягнення тих самих результатів порівняно з Java, що робить його більш зручним для розробників^[16].

До недоліків ж можна віднести нестабільність мови через оновлення через які можуть виникнути проблеми з сумісністю або стабільності коду, а також можливі проблеми з продуктивністю.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Для розробки мобільного застосунку в даній кваліфікаційній роботі було обрано мову програмування Java. Основними причинами вибору даної мови стали більший досвід роботи з нею ніж з Kotlin, а також її широка підтримка та доступність ресурсів та документації.

Для того щоб створити мобільний застосунок для операційної системи Android з використанням мови програмування Java, необхідно обрати середовище розробки, яке зможе надати достатній та зручний інструментарій, який дозволить виконати всі поставлені раніше задачі. Серед найбільш відомих середовищ розробки для розгляду було обрано Android Studio та NetBeans.

NetBeans має ряд переваг таких як: його безкоштовність та відкритість, а також широкі можливості для розширення, так як дане середовище має велику кількість плагінів та розширень, які дозволяють збільшити його функціональність для відповідності специфічним потребам розробника. Також дане середовище має добру підтримку мови програмування Java, що робить його придатним для розробки мобільних додатків для Android. Проте недоліки даного середовища розробки наступні, а саме: його обмежена підтримка Android незважаючи на те, що його можна використати для розробки застосунків для даної операційної системи, а також те, що він має невелику спільноту користувачів, що може призвести до меншої кількості доступних ресурсів та допомоги в мережі^[17].

В свою чергу Android Studio є офіційним інструментом для розробки мобільних застосунків для операційної для платформи Android, який розроблений і підтримується Google, що означає його постійну підтримку та інтеграцію з різними оновленнями та новими функціями операційної системи. Також він має широкий набір інтегрованих інструментів, таких як: редактор коду, графічний дизайнер, симулятори пристроїв, що робить розробку більш зручною і продуктивною. До недоліків даного інструменту можна віднести його високі вимоги до ресурсів, що може бути недоцільним для деяких розробників зі слабшими комп'ютерами, а також його складність для новачків^[18].

Враховуючи всі переваги та недоліки даних середовищ розробки, було прийняте рішення використовувати Android Studio. Оскільки дане середовище є

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

офіційним для розробки мобільних застосунків на Android, та містить весь необхідний інструментарій для комфортної роботи.

Оскільки для розроблюваного застосунку планується використовувати базу даних, то є необхідним також визначити, яка з них найкраще підходить для даного програмного забезпечення. Серед всіх баз, які можна використати при умові, що розробка ведеться для операційної системи Android, з використанням мови програмування Java та середовища розробки Android Studio можна виділити тільки дві – це SQLite та Firebase.

SQLite є частиною стандартного комплекту розробки Android, тому вона легко інтегрується з Android Studio і не потребує додаткових налаштувань, також вона є легкою вбудованою базою даних, яка не вимагає окремого сервера для роботи. Має простий синтаксис SQL для взаємодії з нею, є досить ефективною для мобільних пристроїв і може легко обробляти великі обсяги даних. До недоліків даної бази даних можна віднести її обмежені можливості порівняно з більш потужними серверними базами даних, а також її нездатність підтримувати великі об'єми одночасних запитів^[19].

В свою чергу Firebase є хмарною базою даних, що надає можливість зберігання даних у хмарі і легко синхронізувати її на різних пристроях та платформах. Дана база даних має ширший спектр функціональних можливостей, які включають в себе власний механізм аутентифікації користувачів, зберігання файлів та базу даних в реальному часі. Окрім цього, вона має дружній інтерфейс та просту документацію, що може бути зручним для її використання розробниками. Недоліками її є обмежена можливість розширення, що може бути проблемним у застосунку зі складними вимогами до бази даних, а також те, що деякі з її функцій можуть бути платними або мати обмежений функціонал^[20].

Оскільки для розроблюваного застосунку потрібно зберігати дані багатьох користувачів, які б могли змінюватись в реальному часі, то було прийнято рішення використати базу даних Firebase, оскільки її функціоналу цілком достатньо для того, щоб зберігати ті дані, які будуть використовуватись в програмному забезпеченні.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

2.7 Висновки

Після того, як було описано яка архітектура буде розроблятися в мобільному застосунку, а також вказано чому обрані саме такі архітектурні рішення та детально описано їх, була також визначена функціональна структура застосунку, яка показує те, як різні елементи обраного архітектурного рішення взаємодіють між собою.

Оскільки в програмному застосунку планується використати базу даних було відповідно спроектовано її структуру. В результаті даного проектування було встановлено, які таблиці повинні існувати в базі даних, які дані вони повинні містити та який формат повинні мати ці дані.

Далі з використанням відповідних інструментів було розроблено макети інтерфейсу користувача, який враховував потреби цільової аудиторії і створювався таким чином, щоб не викликати проблем у користувачів при використанні мобільного застосунку. В подальшому на основі даних макетів було розроблено також прототипи екранів, які в подальшому необхідно використати при реалізації розмітки відображень екранів мобільного застосунку.

З урахуванням функціональних вимог було розроблено основні алгоритми роботи мобільного застосунку, тобто описано, які дії повинен виконувати користувач, щоб виконати те чи інше завдання. Подані дані алгоритми були з використанням діаграм діяльності.

В кінці проводився вибір платформи, для якої розроблятиметься програмне забезпечення, а також визначалися інструменти, за допомогою яких можна найкраще реалізувати його для цієї платформи. Спершу обиралась мова програмування, на основі цього вибору обиралось середовище розробки а потім і база даних яка найкраще б підходила для застосунку.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Реалізація логіки мобільного застосунку

Після виконання попередніх етапів з проектуванням мобільного застосунку, під час яких було визначено, який функціонал він повинен містити, а також визначенням деяких алгоритмів його роботи, що вказували на те, які дії необхідно виконати для досягнення певної цілі, та було обрано платформу, для якої розробляється застосунок, мову програмування та середовище розробки, наступним кроком є реалізації логіки додатку з використанням зібраних даних та обраних інструментів.

Вся логіка розроблюваного програмного забезпечення буде використовувати такий клас, як Activity, який будуть наслідувати всі інші класи. Всі вони будуть представляти собою логіку при роботі користувача з відповідним їм екраном^[21].

Першим таким класом виступає LoginActivity, який містить в собі логіку роботи екрану з авторизацією та реєстрацією користувача в застосунку. Для цього в даному класі створюються змінні, які повинні приймати в себе дані введені користувачем для її подальшої роботи та методи, які будуть обробляти різні дії, що буде виконувати користувач при роботі з представленням.

Першими змінним виступають userNameText та passwordText, в які будуть записуватись відповідні дані, глобальна змінна userName, що буде використовуватись в подальшому, змінні mDataBase, KEY та екземпляр newUser моделі User, які відповідатимуть за зв'язок з базою даних.

Основними методами даного класу, які реагуватимуть на дії користувача, оброблятимуть їх використовуючи раніше описані змінні, та виводитимуть відповідний результат є onCreate(), що буде виводити представлення екрану, а також буде створювати посилання на поля даного екрану і присвоювати їх userNameText та passwordText для того, щоб введені користувачем дані в полях

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

екрану присвоювались цим змінним, також даний метод буде ініціювати підключення до певної таблиці бази даних.

`authorize_click()` та `register_click()` відповідають відповідно за процедури авторизації та реєстрації користувача, при натисканні відповідних кнопок на екрані. Вони спершу перевіряють чи є підключення до Інтернету за допомогою метода `isNetworkAvailable()`, так як для коректної роботи бази даних Firebase потребується дане підключення, далі кожен з них перевіряє те, чи ввів користувач якісь дані, та в результаті даної перевірки або виводять спливаюче повідомлення з текстом про відсутність даних, або продовжують свою роботу і здійснюють перевірку на наявність користувача в базі даних. Процес авторизації користувача можна виконати за допомогою даного фрагмента коду:

```
Query checkUser = mDataBase.orderByChild("name").equalTo(name);
checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if (snapshot.exists()) {
            for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                String passwordFromDB =
userSnapshot.child("password").getValue(String.class);
                if (passwordFromDB.equals(password)) {
                    Intent intent = new Intent(LoadActivity.this,
MenuActivity.class);
                    startActivity(intent);
                } else {
                    Toast.makeText(LoadActivity.this, "Невірний пароль",
Toast.LENGTH_SHORT).show();
                    return;
                }
            }
        } else {
            Toast.makeText(LoadActivity.this, "Користувача не знайдено",
Toast.LENGTH_SHORT).show();
            return;
        }
    }
});
```

В даному фрагменті коду, спершу створюється запит та бази таблиці бази даних на те, чи існує в ній такий користувач, далі створюється спеціальний обробник подій, методи якого потрібно перевизначити в методі `onDataChange()` використовується клас `DataSnapshot`, екземпляр якого міститиме дані з бази даних і дозволить працювати з ними в подальшому. Далі здійснюється перевірка того, чи були знайдені дані користувач і виводить спливаюче повідомлення про його

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

відсутність або починає перебирати всі знайдені записи з даним ім'ям користувача і порівнювати паролі там з введеним.

В результаті перевірки або буде викликано екран головного меню застосунку, або буде виведено повідомлення про те, що введений пароль є неправильним. Для реєстрації виконується схожий код з відмінністю в тому, що якщо користувача буде знайдено виведеться відповідне спливаюче повідомлення, в іншому випадку користувача буде додано до бази даних і буде викликано екран головного меню.

Наступною активністю, що розробляється є MenuActivity, яка відповідатиме за те, щоб користувач зміг переходити до інших екранів мобільного застосунку. Для цього клас даної активності міститиме тільки методи, які будуть викликати різні екрани з завданнями та інформацією при натисканні на відповідну кнопку. Першим методом є onCreate(), який відповідає за відображення представлення з усіма кнопками для навігації по застосунку. plusClick(), minusClick(), multiplyClick(), divideClick(), ratingClick(), які відповідатимуть відповідно за відображення користувачу екранів з проходженням тренувальних вправ на додавання, віднімання, множення та ділення чисел, а також на екран проходження рейтингової гри. Останні два методи leadersClick() та rulesClick() при виконанні будуть виводити користувачеві екрани з інформацією про таблиці лідерів та правила. Дані методи мають схожий вигляд, оскільки всі вони створюють екземпляр класу Intent, в якому вказують, який клас необхідно викликати і після чого відповідно використовуючи метод startActivity, та передавши в нього екземпляр класу Intent буде викликати відповідний екран.

Далі розробляються активності для роботи з екранами тренувальних вправ та оскільки їх функціонал є досить схожим, то і структура класів активностей для них теж буде подібною. Всі вони міститимуть п'ять змінних, одна для збереження і відображення користувачеві завдання для розв'язування у відповідне представлення. Змінну, яка буде містити дані для симуляції роботи таймера для користувача, змінну, яка буде зберігати введену користувачем відповідь для її подальшої перевірки, та екземпляр класу, в який було винесено більшу частину

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

функцій цих активностей, для цілей які було вказано в обраному архітектурному рішенні. Також вони всі містять змінну яка представлятиме собою таймер.

Методи, які вони містять в собі – це onCreate(), backClick(), check_click(). Метод onCreate(), як і у всіх попередніх випадках відповідатиме за відображення користувачеві відповідного представлення, а також першого завдання на виконання користувачем. Метод backClick() буде здійснювати повернення користувач на екран головного меню, а метод check_click() буде здійснювати перевірку на те, чи ввів користувач дані, а також визивати методи класу, в якому знаходяться функції для перевірки правильності введеної відповіді та решта необхідного функціоналу який був винесений в нього.

Наступним необхідно розробити клас для екрану рейтингової гри. Незважаючи, що вона має схожий функціонал до тренувальних вправ, реалізація таких же ж методів в даному класі дещо відрізняється, через що весь його функціонал було винесено в окремий клас. Як і тренувальні вправи даний клас містить змінні для відображення користувачеві самого завдання, таймера, а також змінну, яка буде зберігати в собі введену відповідь. Окрім них даний клас також містить змінні temporary, first_numb та second_numb, які зберігатимуть в собі числа, які будуть використовуватись для формування прикладів для розв'язку. Також він буде містити екземпляр допоміжного класу та масив знаків математичних операцій, змінну для роботи з таймером, та змінні для роботи з таблицею бази даних.

Методи, які містить даний клас будуть наступними: onCreate(), Lower(), check_click(), Check(), correct(), Wr_answer(), Time_out(). Метод onCreate() виводить екран з рейтингової грою користувачеві, Lower() генерує приклад на віднімання чисел, check_click() зчитує введені користувачем дані і використовуючи різні перевірки викликає методи допоміжного класу або метод Check(), який в свою чергу здійснює перевірку того, чи користувач ввів дані правильно. В залежності від результату викликає ті ж самі методи допоміжного класу або метод correct(), який збільшує показник правильних відповідей і виводить користувачеві новий приклад для розв'язку, або метод Wr_answer(),

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

який не збільшує кількість правильних відповідей і просто видає наступне завдання. Метод `Time_out()` виконується тоді, коли час відведений у таймері завершується, даний метод здійснює запис даних користувача, а саме: його ім'я та рахунок у відповідну базу даних^[22].

Далі розробляються класи, які відповідають за роботу з таблицею лідерів та правилами. Клас для роботи з таблицею лідерів містить змінні необхідні для роботи з базою даних, серед яких список, в якому зберігатимуться дані отримані з бази, змінні для відображення цих записів користувачу та відповідно змінні для роботи з базою даних. Також даний клас містить такі методи як `onCreate()`, який і в попередніх випадках просто виводить користувачеві екран зі списком лідерів, який він отримує використовуючи метод `getData()` який і здійснює запит до бази даних, отримає з неї дані, які потім записує в список. Далі він сортує даний список і виводить певну кількість користувачів з нього. Метод `backClick()` просто повертає користувача на екран з головним меню. Клас для відображення правил містить лише два методи: `onCreate()`, який виводить екран з текстом правил та `backClick()`, який повертає користувача назад у головне меню.

Останнім класом, який потрібно розробити – буде клас `GameSupport`, який являється допоміжним класом для тренувальних вправ і який містить в собі деякі методи, які були винесені з них і зроблені більш універсальними. Даний клас містить в собі лише дві змінні, які будуть містити інформацію про два числа, що будуть використовуватись для створення завдання, та для обчислення і перевірки. Також даний клас міститиме такі методи як: `RandomNumber()`, `isNumber()`, `Wr_answer()`, `no_value()`, `Time_out()`, `no_number()`, `correct()`, `Check()`, `generate()`, `Lower()`. Метод `RandomNumber()` призначений для того, щоб видати випадком чином обране число з певного діапазону, він приймає два числа, які представляють собою початок та кінець діапазону. В методі `Check()` здійснюються всі основні перевірки, спершу завдяки методу `isNumber()` здійснюється перевірка того, чи введене користувачем значення є числом, далі в залежності від знаку виконується певна математична операція і здійснюється порівняння отриманого значення з тим, що ввів користувач, та в залежності від результату буде

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

викликано метод correct(), який виведе повідомлення про правильність відповіді, або Wr_answer(), який виведе повідомлення про її хибність. Методи no_number() та no_value() просто повертають повідомлення про те, що значення не є числом або воно взагалі не було введено^[23].

Код даних класів та їх методів подано у додатку Б.

3.2 Реалізація розмітки мобільного застосунку

Після того, як було розроблено логіку роботи мобільного застосунку, необхідним є реалізувати його розмітку, для цього необхідно скористатися раніше розробленими макетами інтерфейсу користувача та створеними на їх основі прототипами, тобто на даному етапі необхідно перенести ці прототипи в розроблюване програмне забезпечення використовуючи наявні в обраному середовищі інструменти для цього. Вся розмітка розроблюваних екранів буде створюватись з використанням мови XML.

XML (Extensible Markup Language) – є розширюваною мовою розмітки, що використовується для представлення та обміну даними у вигляді текстових документів. Вона часто використовується для створення структурованих даних, таких як: конфігураційні файли, документи звітності, та інші текстові файли, які будуть читабельними і зможуть оброблятися як людьми, так і комп'ютерами. XML дозволяє визначати власні теги та структури даних, що робить його досить гнучким і універсальним для використання у різних областях та мовах програмування.

У розробці мобільних застосунків, XML використовується для опису інтерфейсу користувача. В XML файлі можна визначити розміщення та вигляд різних елементів, таких як: кнопки, тексти, зображення на екрані мобільного пристрою. Це дозволяє розробникам створювати структуровані та легко читабельні інтерфейси користувача, розділяючи логіку програми від її представлення. Крім того, XML також використовується для зберігання

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

конфігураційних даних, наприклад, параметрів застосунку, URL-адрес, шляхів до ресурсів тощо^[24].

Оскільки створювати розмітку екранів мобільного застосунку, та налаштовувати її може бути дещо складним, якщо використовувати сам XML. Проте в даній проблемі є рішення, яке надає середовище розробки, а саме зручний графічний дизайнер з достатнім для розроблюваного мобільного застосунку інструментарієм. Також даний дизайн дозволяє розділити екран, одночасно представляючи як графічний інтерфейс для роботи з графічним дизайнером, так і з XML представленням.

Першим екраном, розмітку якого необхідно створити є `load_screen.xml`, який представлятиме собою екран для авторизації та реєстрації користувача. Відповідно до прототипу на ньому потрібно розмістити поля, в які користувач зможе вводити свої дані для подальшої авторизації чи реєстрації в мобільному застосунку. Для цього можна використати такий елемент як `PlainText`, який відповідає за текстове представлення даних. Проте для того, щоб привернути увагу користувача до даних полів можна додати елементи `PlainText` в якесь визначене місце, для створення якого можна використати простий прямокутник білого кольору, в який потрібно помістити необхідні елементи. Так як таких полів має бути два, створюється два таких прямокутника, в які поміщаються такі елементи як `UserName` та `Password`, які є стандартними елементами `PlainText`, але яким даний інший ідентифікатор для того, щоб їх потім можна було легко знайти у коді. Потім створені поля для введення розміщуються так, як було показано на прототипі даного екрану. Далі необхідно додати текст, який би вказував на те, які дані необхідно ввести в дані поля. Для цього також використовуються ті ж елементи, що і для полів для введення, одному з них встановлюється текст «Введіть ім'я користувача», тоді як для іншого, встановлюється текст «Введіть ваш пароль» і дані елементи розміщуються над відповідними полями для введення. Оскільки, з цими полями в класі, який працює з даним розроблюваним представленням не відбуватиметься ніяких дій, їх імена можна не змінювати.

Також необхідно створити кнопки, що будуть відповідати за запуск методів,

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

які здійснюватимуть авторизацію та реєстрацію користувача у мобільному застосунку. Для цього потрібно використати такий елемент як Button, що представляє собою кнопку, при натисканні на яку буде викликатись певна дія. Відповідно потрібно створити дві таких кнопки, дати їм унікальні імена, а також вписати в них текст, який б показував за яку дію вона відповідає. Після створення дані кнопки потрібно розмістити так, як показано на розробленому для даного екрана прототипі. Щоб розмітка повністю відповідала прототипу, потрібно також додати назву додатку і розмістити її у верхній частині екрану, для чого також можна використати TextView.

Наступним екраном, розмітку якого необхідно створити, є activity_menu.xml, яка представляє собою головне меню мобільного застосунку. З нього користувач може перейти до інших екранів і виконувати інші завдання. Відповідно до розробленого для даного екрана прототипу, розмітка повинна включати в себе кнопки, при натисканні на які будуть запускатися обробники, які і будуть переміщати користувача між екранами. Для створення даних кнопок потрібно використати такий елемент як Button.

Спершу потрібно створити кнопки, які будуть відповідати за відображення користувачу екранів з таблицею лідерів та правилами. Відповідно з використанням вище вказаного елемента створюються дані кнопки, але для того, щоб дати їм такий вигляд як на макеті, необхідно створити власний вигляд кнопки і присвоїти його створеній кнопці. Для цього потрібно використати XML в якому задати всі необхідні параметри для кнопок. Також згідно прототипу на кожній кнопці повинен бути текст, який вказує на те, що вона показуватиме при натисканні. Також на даній кнопці потрібно розмістити певну картинку для того, щоб привернути увагу користувача до неї. Для даних кнопок – це картинки трофею та книжки. І для того, щоб їх додати до кнопки можна скористатись двома способами: завантажити піктограму і потім додати її до кнопки через XML-код, або взяти готовий XML-код піктограми на сайті, який надає таку можливість, потім створити XML-файл в папці drawable і підключити його до кнопки через тег android:drawableTop, для того щоб відобразити піктограму над текстом. Створенні

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

кнопки згідно прототипу потрібно розмістити у верхній частині екрану одна біля одної.

Далі потрібно по такому ж принципу створити кнопки для тренувальних вправ і розмістити їх згідно прототипу. Оскільки кнопка для переходу на екран рейтингової гри відрізняється по дизайну від інших кнопок, то для її створення використовується дещо інших підхід. В даному випадку дана кнопка створюється, як зображення за допомогою елемента `ImageView`, для якої створюється унікальний вигляд, як на прототипі. Також в коді даного елемента додається `android:onClick="ratingClick"` для того, щоб він виступав як кнопка. Всі кнопки потім розміщуються згідно їх розміщення в прототипі.

Далі створюються розмітки для екранів тренувальних вправ та рейтингової гри, які взаємодіятимуть з відповідними їм класами. Ці розмітки відповідно наступні: `activity_divide_training.xml` представляє собою екран для виконання тренувальних вправ з діленням чисел; `activity_minus_training.xml` є розміткою для завдання з відніманням чисел; `activity_multiply_training.xml` відповідає за вигляд екран для проходження тренувальної вправи на множення чисел; `activity_plus_training.xml` представляє собою екран, на якому користувач зможе попрактикуватись в додаванні чисел, `activity_rating_game.xml` що представляє екран, який об'єднує в собі всі попередні завдання і відображає користувачу екран, на якому він зможе прийняти участь в рейтинговій грі.

Всі ці екрани судячи з їх прототипів будуть схожими, тому як і для екрану з авторизацією та реєстрацією користувача створюється поле для введення, яке використовує для своєї побудови білий прямокутник і елемент `PlainText`, якому надається унікальне ім'я. Оскільки, з нього потім потрібно буде отримати значення, цей елемент поміщається у прямокутник і потім розміщується так як на прототипах. Також за допомогою `TextView` створюється елемент, що буде приймати в себе математичний приклад, який потрібно буде вирішити користувачу, та для якого потрібно зробити унікальний ідентифікатор в кожній розмітці. За допомогою того ж `TextView` створюється також елемент, який буде симулювати роботу таймера, та якому задається певне ім'я для того, щоб потім

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

його можна було оновлювати з коду мобільного застосунку. Біля цього таймера також потрібно розмістити якусь піктограму таким же ж чином, як було описано раніше. Згідно розроблених макетів для даних екранів у всіх розмітках крім екрану рейтингової гри потрібно розмістити дві кнопки. Одна з них слугуватиме для виклику методів перевірки і видачі нового завдання, тоді як інша буде просто переправляти користувача на головний екран. Далі згідно прототипів в розмітці розміщуються елементи, які міститимуть певну довідкову інформацію.

Останніми екранами розмітки для яких необхідно розробити – це екрани таблиці лідерів та правил. Згідно прототипу таблиці лідерів, розмітка повинна містити текст, який б вказував на те що це за екран.

Даний текст можна створити використовуючи TextView, після чого задати створеному елементу відповідне значення. Також створюється ще один такий елемент, який буде вказувати на те, скільки саме користувачів відобразатиметься в даній таблиці. Окрім цього потрібно виділити певне місце, в якому будуть відобразатись користувачі, які знаходяться в таблиці. Для цього можна створити два елемента TextView і налаштувати їх так, щоб можна було відобразити список користувачів, а також те скільки балів вони набрали під час проходження рейтингової гри. Також необхідно створити кнопку для можливості повернення у головне меню і розмістити всі ці елементи згідно їх положення на прототипі.

Для екрану правил розмітка робиться схожим чином, тобто створюється текстова мітка, яка вказує на те, що це за екран. Під цією міткою повинно виділятися місце, в якому будуть відобразатись основні правила користування застосунком. І під всім цим розміщується кнопка, для того щоб можна було повернутись на екран головного меню.

3.3 Розроблення бази даних

Для того, щоб зберігати дані користувачів, а саме їх імена, паролі та рахунок, який вони набрали під час проходження рейтингової гри, та мати

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

можливість порівняти деякі з цих даних для авторизації користувача або визначення йому місця у таблиці лідерів було спроектовано базу даних, яка повинна містити лише дві таблиці: одна відповідатиме за збереження імені користувача та його пароля і буде використовуватись при авторизації чи реєстрації в мобільному застосунку; тоді як інша, буде зберігати дані про ім'я авторизованого користувача та його рахунок у рейтинговій грі, та буде використовуватись для запису цих даних в список, який потім буде впорядковуватись за певним полем і виводитись у відповідному представленні з деякими додатковими особливостями. Поля даних таблиць разом з типами їх даних та іншими параметрами були подані у відповідних таблицях, ще на етапі проектування. Також на етапі вибору технологій за базу даних, в якій і будуть створюватись дані таблиці було обрано Firebase, чому був здійснений даний вибір описано на етапі вибору технологій^[25].

Для того, щоб користуватись даною базою даних у розроблюваному мобільному застосунку середовище розробки Android Studio вже надає достатній інструментарій для цього. Щоб почати роботу необхідно обрати меню Tools, в якому обрати Firebase для того, щоб викликати асистента, який допоможе створити базу даних і пізніше підключитись і використовувати її. У відкритому діалоговому вікні потрібно обрати пункт з базою даних в реальному часі і пройти всі кроки. Спочатку необхідно підключити програмне забезпечення до Firebase. Для цього потрібно зареєструватись на їх офіційному сайті^[26], а також у самому середовищі для того, щоб синхронізувати їх між собою.

Наступним кроком, потрібно створити на сайті новий проект і підключити в його налаштуваннях розроблюваний Android застосунок слідуючи покроковій інструкції. Після того, як все буде підключено потрібно додати Realtime Database до проекту використовуючи асистента Android Studio, який все зробить сам.

Наступним кроком стає налаштування прав доступу до даної бази даних, на самому сайті обравши базу даних в реальному часі можна в верхній частині сторінки побачити меню з чотирма пунктами: Data, Rules, Backups, Usage. В Data буде відображатись структура бази даних з записаними в таблицях значеннями, в

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Backups будуть знаходитись резервні копії баз даних, але щоб отримати доступ до даної функції безкоштовного плану недостатньо. Пункт Usage показує те, як використовується база даних, тоді як Rules буде містити права доступу до бази даних, для даного мобільного застосунку достатньо надати права на читання та запис даних.

Для подальшої роботи з даною базою даних використовуються офіційні бібліотеки від Google для Firebase. Наприклад, для того щоб додати в базу певні дані можна використати даний код:

```
FirebaseDatabase database = FirebaseDatabase.getInstance();  
DatabaseReference myRef = database.getReference("message");  
myRef.setValue("Hello, World!");
```

Порівняно з іншими базами даних Realtime Database, яку надає Firebase значно відрізняється. Так дані, які в ній зберігаються у вигляді деревоподібної структури, схожої на JSON-об'єкт. Кожен вузол в дереві має свій унікальний ідентифікатор, який називається ключем. Цей ключ визначає місце вузла в структурі дерева та є важливим для доступу до конкретних даних, кожен з таких вузлів може містити в собі різні типи даних такі як: рядки, числа, об'єкти або списки. Всі вузли в даній структурі можуть мати як батьківський вузол, який безпосередньо передує вузлу так і дочірні вузли, які безпосередньо йому підпорядковані. Ще одним важливим елементом даної структури є корінь дерева, який є вищим рівнем і є початковим елементом у дереві, він не має батьківського вузла і є базовим елементом з якого починається саме дерево^[27].

Для того, щоб в такій структурі створити таблиці потрібно звернутись до кореня дерева і додати два вузли, які будуть представляти собою таблиці. Далі в програмній частині мобільного застосунку використовуючи моделі і вказуючи ключі, або ж назви даних таблиць буде отримуватись доступ до відповідних вузлів даного дерева. В них потім будуть створюватись вузли, які будуть представляти собою записи в даній таблиці, кожен з яких буде містити вузли, які представлятимуть собою поля з даними. Структуру такого дерева подано на рисунку 3.1

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

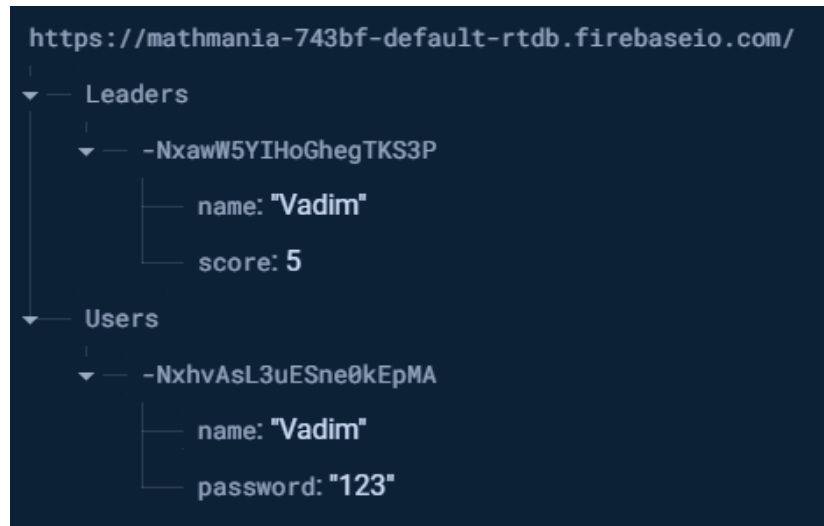


Рисунок 3.1 – Деревоподібна структура бази даних

3.4 Керівництво користувача

На даному етапі необхідним є вказати те, як правильно користуватись розробленим мобільним застосунком, з якими елементами екрану повинен взаємодіяти користувач для того, щоб виконати дії, які були описані на етапі реалізації логіки програмного забезпечення. Тобто потрібно створити керівництво користувача, яке і буде вказувати, що потрібно робити для досягнення тієї чи іншої цілі, а також буде опис певних особливостей застосунку.

Для початку роботи користувачу необхідно відкрити встановлений мобільний застосунок, після чого йому буде показано екран, де йому потрібно буде авторизуватись в ньому, або ж зареєструватись, якщо застосунок використовується вперше. Для обох дій користувачеві необхідно у визначені для цього поля ввести своє ім'я та пароль, щоб була змога зрозуміти в яке поле, та які дані вводити над кожним з них розміщений певний текст, який вказує на те, що в нього потрібно ввести. Після чого користувачу необхідно натиснути кнопку «Увійти», для того щоб запустити процес авторизації. На даному моменті впливає одна з особливостей мобільного застосунку, а саме те, що він потребує доступ до Інтернету. Це необхідно для того, щоб перевірити наявність користувача в базі даних, яка знаходиться на віддаленому сервері і в залежності

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

від результату даної перевірки виконати певну дію. Для того, щоб користувач не вважав відсутність реакції на його дії, як за помилку, йому буде виводитись повідомлення про відсутність з'єднання з Інтернетом. Такі ж повідомлення зі своїм вмістом, будуть виводитись, якщо користувач не введе якісь зі своїх даних. При умові того, що користувач ввів усі необхідні дані, в нього є з'єднання з Інтернетом і він натиснув кнопку «Увійти» можуть відбутись наступні дії: виведеться повідомлення про те, що користувача з введеним іменем не знайдено, що може означити його відсутність в базі даних або неправильно введене ім'я, або буде показано повідомлення про те, що пароль невірний, що означає наявність користувача в базі, але дані для входу некоректні. Якщо ніяких повідомлень не буде показано, то користувачу буде показаний екран головного меню.

При реєстрації будуть виводитись повідомлення про те, що введене ім'я кимось зайняте, якщо ж дане повідомлення не буде показано, то користувачу буде показано екран з головним меню, а його дані будуть записані в відповідну таблицю в базі даних.

На екрані головного меню користувач при натисканні на різні кнопки буде отримувати різні екрани мобільного застосунку, кожен з яких буде містити визначений для функціонал. Усього користувач зможе взаємодіяти з шістьма такими кнопками, для початку бажаним є ознайомлення з правилами застосунку. Для того щоб це зробити, потрібно натиснути на кнопку з відповідним текстом та картинкою книжки, після чого буде виведений екран з коротким описом того, що можна виконувати в даному застосунку, щоб повернутись назад у головне меню потрібно натиснути єдину кнопку яка є на цьому екрані.

Після ознайомлення з правилами користувач може як проходити тренування, так і одразу перейти до рейтингової гри. При натисканні на кнопки «Додавання», «Віднімання», «Множення» та «Ділення» з відповідними їм зображеннями математичних операцій, будуть виводитись екрани з відповідними прикладами для вирішення. Одразу при показі екрану, буде також показано перший математичний приклад, який необхідно розв'язати і вписати свою відповідь у поле, яке знаходиться біля прикладу. Після чого потрібно натиснути

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

кнопку «Перевірити» і відповідно запустити процедуру перевірки введеної відповіді. В залежності від результату буде виведено одне з наступних повідомлень: при умові що відповідь правильна, виведеться діалогове вікно, яке буде вказувати на про це і яке міститиме кнопку, при натисканні на яку користувач отримає новий приклад для розв'язку, якщо ж відповідь була неправильною буде виведено діалогове вікно, яке наголосить на цьому. А також вказуватиме, яка відповідь була правильною, як і в попередньому випадку при натисканні на кнопку в даному вікні буде отримано новий приклад для розв'язку. Також при натисканні кнопки для перевірки виводитимуться повідомлення про те, що не була введена відповідь або вона не є числом. Окрім цього, для кожного завдання доступний таймер, який вказує на час, за який потрібно дати відповідь, та по завершенню якого виведеться відповідне повідомлення. Користувач зможе виконувати тренувальні вправи доти, поки не натисне кнопку «Назад» для того, щоб повернутись у головне меню.

Проходження рейтингової гри в цілому схоже на тренувальні вправи. Користувач також отримує приклад для розв'язку і вводить відповідь на нього, після чого тисне на кнопку перевірки, тільки тепер йому не будуть показуватись повідомлення про те, чи була введена ним відповідь правильною, але повідомлення про те, що не було введено значення, або воно взагалі не являється числом залишаються.

Дана гра буде проходитись до тих пір, поки не закінчиться час, який відведений для неї, та який відображається на таймері. По завершенню таймера буде виведено діалогове вікно, яке вказуватиме на кількість правильно введених відповідей і при натисканні кнопки, в якій дані будуть записані в відповідну таблицю в базі даних і користувачеві буде показано екран головного меню.

Для того, щоб переглянути кількість набраних балів та своє місце серед інших користувачів у головному меню потрібно натиснути на кнопку «Лідери» з картинкою трофея, після чого буде виведено екран з першими п'ятьма відсортованими по рахунку користувачами.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

3.5 Технічні характеристики мобільного застосунку

Після того, як було розроблено мобільний застосунок і визначена логіка його роботи, наступним кроком стає визначення технічних характеристик пристрою, на якому цей застосунок буде запускатися та тестуватися. Це необхідно зробити для забезпечення оптимальної продуктивності та коректної роботи застосунку. Технічні характеристики пристрою повинні відповідати вимогам самого застосунку, забезпечуючи його стабільну та ефективну роботу, до них можна віднести: кількість оперативної пам'яті та обсяг вільного місця на диску чи характеристики процесора. Правильний вибір технічних характеристик має забезпечити оптимальну продуктивність, та дозволити користувачу без зайвих проблем взаємодіяти з додатком.

Відповідно характеристики мобільного пристрою для того, щоб на ньому можна було успішно запустити розроблений мобільний застосунок наступні:

– операційна система: оскільки за прийнятими на етапі вибору технологій рішеннями, даний мобільний застосунок розроблявся для такої операційної системи як Android, необхідним є вказати мінімальну та рекомендовані версії даної платформи для коректної роботи застосунку. Мінімальною версією Android для коректної роботи є 8.0, тоді як рекомендованою версією є 10 Q і вище;

– технічні характеристики: для коректної роботи розробленого мобільного застосунку, пристрій повинен мати процесор по характеристиками близький до Qualcomm Snapdragon 665 Octa-core (4x Kryo 260 at 2.0 GHz & 4x Kryo 260 at 1.8 GHz). Щодо обсягу оперативної пам'яті, то мінімальна вимога – це 2 GB RAM, тоді як рекомендована для роботи 4 GB RAM. Вільного місця на диску для того, щоб можна було встановити застосунок потрібно в районі 50 MB;

– параметри екрану: для того, щоб правильно відображати створену для екранів мобільного застосунку розмітку, параметри екрану мобільного пристрою користувача повинні бути такими як: діагональ екрана має становити 6.53 дюйма з розширенням 2340x1080 пікселів, з частотою оновлення екрану в 60 Гц та типом екрану IPS.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Також для того, щоб дані користувача можна було записувати кудись для мобільного застосунку була розроблена база даних з використанням певних інструментів. Як результат для застосунку існує віддалена база даних, для доступу до якої потрібне стабільне з'єднання з мережею. Відповідно пристрій повинен підтримувати мережеві стандарти, такі як 4G/LTE, 5G або Wi-Fi. Для мобільного Інтернету важливо мати підтримку мережі 4G або 5G; тоді як для Wi-Fi рекомендується підтримка стандартів Wi-Fi 802.11 a/b/g/n/ac/ax.

До переліку пристроїв, які мають подібні характеристики, та на яких можна успішно запустити мобільний застосунок можна віднести Xiaomi Redmi Note 8 Pro, Samsung Galaxy A51, Motorola Moto G Power (2020), Nokia 7.2, Huawei Y9s.

3.6 Тестування мобільного застосунку

3.6.1 Аналіз методів тестування мобільного застосунку

Останнім, проте не менш важливим етапом, при розробці будь-якого програмного забезпечення є його тестування. Тестування програмного забезпечення є процесом дослідження та оцінки програмного продукту з ціллю виявлення помилок в його роботі, недоліків та інших проблем. Виконується воно з метою переконатися в тому, що розроблене програмне забезпечення відповідає всім визначеним перед його розробкою вимогам, воно виправдовує очікування користувачів та працює належним чином^[28].

Мета та важливість тестування програмного забезпечення полягає в багатьох факторах серед яких можна виділити наступні:

– виявлення помилок: тестування допомагає виявити помилки та дефекти в програмному забезпеченні до того, як воно буде введене в експлуатацію. До цих дефектів можна віднести помилки логіки, або проблеми з продуктивністю;

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

– підвищення якості: після проведення тестування можна значно покращити якість програмного забезпечення. Тобто можна зробити його більш надійним, зручним у використанні та стійким до помилок;

– задоволення користувачів: воно дозволяє надати гарантії того, що програмне забезпечення відповідає потребам та очікуванням користувачів. Це відповідно робить його більш привабливим для користувачів, та сприяє кращому досвіду роботи з ним;

– зниження витрат: ще один важливий фактор, через який потрібно проводити тестування це те, що виявлення помилок на ранніх стадіях розробки може значно заощадити час та гроші, оскільки якщо серйозні помилки будуть знайдені на стадії експлуатації програмного забезпечення можуть виникнути непоправні наслідки, які буде досить складно усунути в майбутньому, та на що може піти значна кількість коштів.

Для того щоб в повній мірі перевірити розроблене програмне забезпечення існує багато різних типів тестування, кожен з яких можна використовувати для різних цілей. Деякі з найпоширеніших типів включають:

– функціональне тестування, яке перевіряє, чи відповідає програмне забезпечення очікуваній функціональності;

– навантажувальне тестування, що перевіряє, як програмне забезпечення поводить себе в умовах високого навантаження;

– тестування безпеки, яке має ціль перевірити, чи програмне забезпечення безпечно від кібератак та інших вразливостей.

Всі вище перераховані аспекти тестування програмного забезпечення повністю підходять і для тестування мобільних застосунків, яке проводиться з урахуванням особливостей мобільної платформи такі як: різні розміри екранів, операційні системи та типи введення^[29].

Для розробленого в даній кваліфікаційній роботі мобільного застосунку, тестування буде проводитись з використанням всіх перелічених раніше особливостей. Тобто необхідно протестувати чи функцій в ньому відповідають тим, що були зазначені у вимогах до функціоналу, а також чи працюють вони

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

правильно. Необхідно перевірити чи є застосунок зручним у користуванні та інтуїтивно зрозумілим, а також те, чи його зовнішній вигляд є привабливим. Оскільки в програмному забезпеченні реалізована база даних, то потрібно також перевірити чи дані, які вводить користувач разом з даними, які можуть бути внесені в неї під час його роботи з застосунком, зберігаються в ній, та чи зберігаються вони правильно і їх можна отримати та відобразити за потреби^[30].

Тестування буде проводитись з використанням двох пристроїв, один з них буде представляти з себе емулятор, який симулюватиме роботу реального мобільного пристрою, тоді як інший буде реальним мобільним пристроєм, характеристики якого будуть схожими з тими, що вказані у вимогах до характеристик мобільного пристрою. На обидва пристрої буде встановлюватись застосунок, перевірятиметься те, як відображаються його екрани, та чи виконується основний закладений в нього функціонал.

3.6.2 Тестування мобільного застосунку за допомогою емулятора

Як було зазначено раніше тестування розробленого мобільного застосунку буде проводитись на двох пристроях, перший знаходиться в емуляторі тоді, як другий є реальним фізичним пристроєм. Для того, щоб протестувати застосунок з використанням емулятора необхідно його створити. Оскільки для розробки було обрано таке середовище розробки як Android Studio, яке надає можливість створити емулятор мобільного пристрою. Для цього в ньому потрібно відкрити пункт «Tools» в навігаційній панелі і в ньому обрати пункт «Device Manager», після чого з'явиться вікно, яке запропонує два варіанти: створити віртуальний мобільний пристрій або підключитись до реального.

Для того, щоб створити емулятор, потрібно натиснути відповідну кнопку для того, щоб викликати вікно, в якому потрібно обрати з готових варіантів розмірів пристрою той, що підходить по характеристикам. Серед доступних варіантів було обрано варіант, який має розмір екрану в 6.4 дюйма та розширення

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

екрану 1080x2400 пікселів, що найбільш близьке до тих параметрів, що вказані в технічних вимогах. Після чого необхідно обрати версію операційної системи. Для тестування було обрано 8 версію, яка вказана, як мінімальна для того, що на ній можна було запустити застосунок.

Після створення, даний емулятор необхідно запустити і встановити на нього розроблений застосунок, після чого відкрити його. Після успішного відкриття необхідно перевірити чи працює закладений в програмне забезпечення функціонал правильно. Для початку необхідно перевірити чи працює авторизація та реєстрація користувача, тому у відповідні поля екрану вводяться необхідні дані та натискаються відповідні кнопки. Якщо все працює, як задумувалось, то при правильно введених даних користувачу буде відображено екран з головним меню. На цьому екрані потрібно перевірити чи відбуваються дії при натисканні на кнопки, і чи виводяться відповідні їм екрани. Щоб перевірити роботу функціоналу з тренувальними вправами потрібно натиснути на кнопку «Додавання» і виконати всі дії описані в керівництві користувача для виконання даної тренувальної вправи. Результати тестування з виглядом застосунку в емуляторі подано на рисунку 3.2

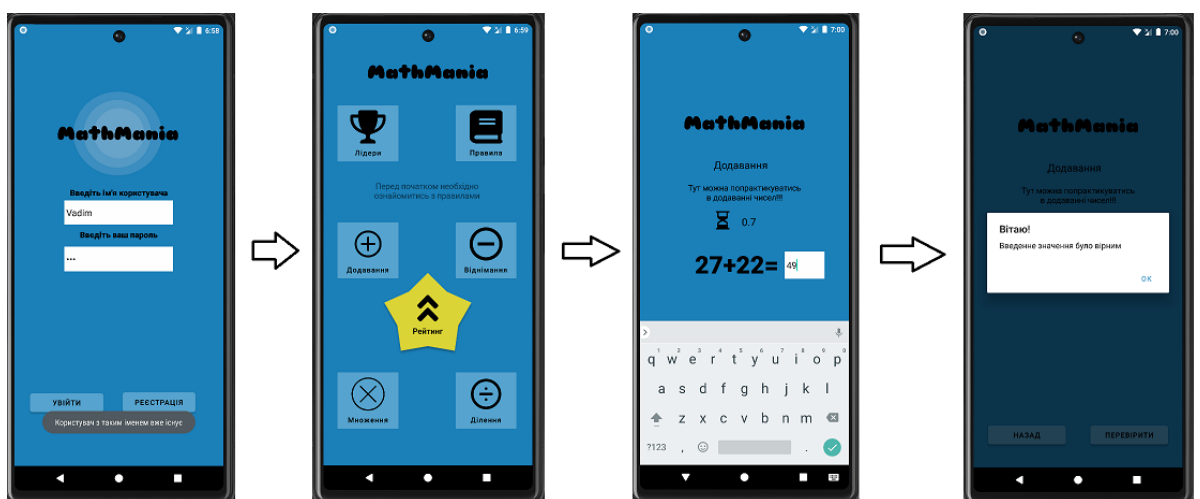


Рисунок 3.2 – Тестування функцій застосунку в емуляторі

Як можна побачити з рисунка функціонал, який необхідно було перевірити працює, а створені екрани відображаються в необхідній мірі. Але тестування

					КВРІПЗ.200177.01.26.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

мобільного застосунку на емуляторі не може дати повної картини того, чи він працює правильно оскільки емулятор використовує ресурси пристрою на якому він запущений, враховуючи лише вручну задані параметри. Тому необхідно протестувати мобільний застосунок також на реальному мобільному пристрої. Для того, щоб це зробити Android Studio надає різні можливості. Одна з них дає можливість створити .apk-файл через, який потім можна встановити застосунок на пристрій, тоді як інша дозволяє через USB підключити пристрій до середовища і відкрити застосунок напряму. Це також може бути зручним так, як дозволить одразу вносити нові зміни і застосовувати їх, без створення нового .apk-файлу, проте для цього потрібно надати певні дозволи у самому пристрої.

Процес тестування на реальному пристрої не відрізняється від такого ж у емуляторі. В даному випадку теж потрібно перевірити чи виконується функціонал та чи правильно відображаються екрани застосунку, але тепер необхідно перевірити чи вносяться дані користувача в базу даних. Для цього необхідно зареєструватись, як новий користувач і в головному меню обрати рейтингову гру натиснувши на відповідну кнопку, пройти її, та після проходження перевірити чи були дані внесені до бази. Результат тестування показано на рисунку 3.3

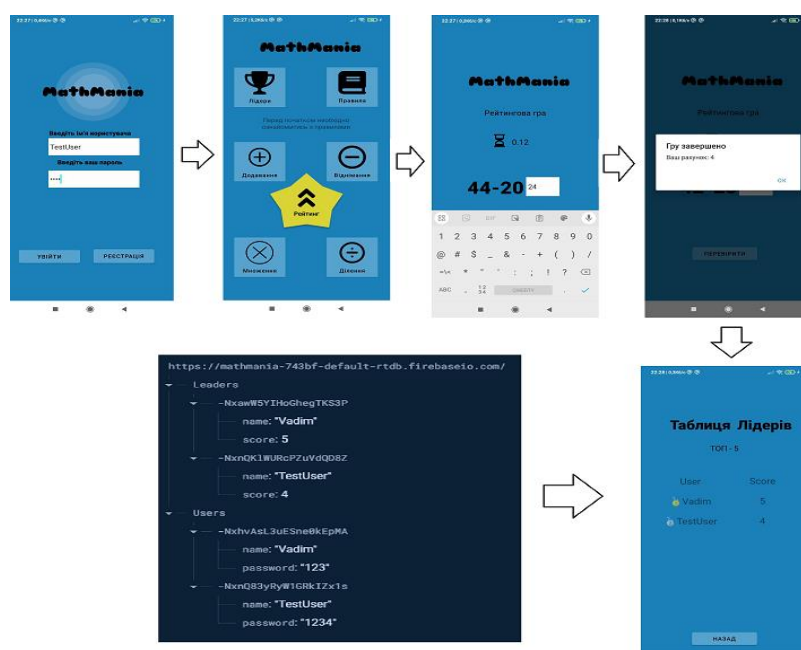


Рисунок 3.3 – Тестування функцій застосунку на реальному пристрої

3.6.3 Аналіз результатів тестування мобільного застосунку

Після проведення тестування було встановлено, що більша частина функціоналу мобільного застосунку працює, як задумувалось, а його екрани відображаються так, як вказано на прототипах. Дані, які вводить користувач також заносяться в базу даних і при потребі їх можна звідти вивести.

Проте, було виявлено певні проблеми та незручності під час тестування додатку. Для початку було виявлено, що при натисненні певних кнопок на самому пристрої можна повернутись до екранів застосунку коли це не є доречним, через що можуть виникати різного роду помилки, тому бажано їх не натискати. Ця ситуація може виникати через те, що застосунок працює використовуючи різні Acivity і коли в коді створюються нові такі елементи попередні не видаляються, тому в майбутньому потрібно вирішити дану проблему. Ще однією незручністю, більше ніж проблемою, можна назвати те, як вводяться дані з клавіатури, спершу з'являється клавіатура з літерами і кожен раз потрібно перемикатись на числа. Оскільки часу на виконання дається достатньо, це не сильно заважає проте при його зменшені дана проблема може зіпсувати досвід користувача.

3.7 Висновки

Після визначення вимог до розроблюваного у кваліфікаційній роботі програмного застосунку, та розробці на їх основі архітектурних рішень, які б дозволили в достатній мірі реалізувати визначені вимоги, створенні інтерфейсу користувача з використанням макетів і подальшого створення прототипів екранів, визначенні логіки роботи застосунку та виборі технологій, настав етап реалізації мобільного застосунку з використанням усіх перерахованих даних.

Спершу було проведено реалізацію логіки мобільного застосунку з використанням обраних архітектурних рішень на технологій. В результаті розроблено класи з їх атрибутами та методами які будуть використовуватись при

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

різних реакціях на дії користувача, які він буде проводити на своїй стороні і отримувати відповідні результати.

Далі створено розмітку для екранів мобільного застосунку, з якими буде взаємодіяти користувач, тобто різні кнопки та поля для введення. Вся розмітка була виконана в залежності з раніше розробленими прототипами кожного екрану. Також була розроблена база даних, структура якої дещо відрізняється від звичної структури інших баз даних. Вказано яким чином здійснюється зв'язок зі створеною базою даних, та як в неї вносяться дані.

На основі раніше розроблених алгоритмів роботи застосунку та з використанням реалізованої логіки та розмітки, створено керівництво користувача. Воно пояснює, що можна робити у застосунку та яким чином це можна зробити, а також те, що буде отримувати користувач у відповідь на свої дії.

В кінці проведено тестування готового застосунку на емуляторі та реальному пристрої з метою знаходження потенційних помилок, та перевірки працездатності застосунку в цілому.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У рамках кваліфікаційної роботи було необхідно провести дослідження щодо потреб учнів початкових класів та вчителів у математичному навчанні, оцінити існуючі рішення та визначити їх переваги та недоліки. На основі даного дослідження відповідно необхідно було розробити концепцію та визначити вимоги до мобільного застосунку.

Спершу проводилось дослідження предметної області мобільного застосунку. В даній кваліфікаційній роботі за предметну область було обрано навчальну гру, яка мала б покращувати знання учнів початкових класів в області математики. Було досліджено те, що з себе представляють навчальні ігри та який їх стан на теперішній час, що вони роблять та які проблеми вони мають. Проведено аналіз існуючих рішень які відповідають предметній області кваліфікаційної роботи, виявлено їх переваги, які можна використати при розробці власного рішення, та недоліки, які бажано уникнути. На основі цих даних визначено ряд функціональних та нефункціональних вимог до розроблюваного мобільного застосунку, вимоги були подані з використанням UML-діаграм варіантів використання та таблиць з їх більш детальним описом.

На основі визначених вимог було обрано архітектурне рішення, яке найкраще підходить для реалізації поставлених цілей. В даній кваліфікаційній роботі – це архітектура, яка базується на шаблоні MVC, а також використовує такий шаблон проектування як Servant. Далі відбувалось проектування бази даних для мобільного застосунку, визначено які таблиці в неї входять, та які дані вони повинні зберігати, після чого спроектовані макети інтерфейсу користувача, на основі яких були розроблені прототипи екранів мобільного застосунку. Описано логіку роботи частини функціоналу, який повинен містити застосунок. Для цього використовувались діаграми діяльності, які представляли собою алгоритм того, як виконати певні дії та які результати будуть отримані як їх наслідок. В кінці обрано технології для розробки застосунку. За платформу для розробки було обрано Android, для розробки під дану платформу було обрано мову

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

програмування Java та середовище розробки Android Studio, яке надає весь необхідний інструментарій. За базу даних для застосунку було обрано Firebase.

Після того, як були визначення вимоги, а також спроектовані на їх основі архітектурні рішення, описана логіка роботи застосунку та розроблені прототипи екранів його інтерфейсу, завершальним етапом стала реалізація та тестування розробленого програмного забезпечення. Спершу з використанням обраної мови програмування та архітектурно рішення була реалізована логіка мобільного застосунку. В результаті чого створені класи з їх атрибутами та методами, які оброблятимуть та даватимуть певну відповідь на різні дії користувача над застосунком. На основі розроблених раніше прототипів екрану, створена розмітка екранів мобільного застосунку з використанням такої мови як XML, та описано за що відповідатиме кожен її елемент та що відбуватиметься при взаємодії з ним. Описано розробку бази даних застосунку, яка має нестандартний вигляд порівняно з іншими базами даних. Оскільки вона являється деревоподібною структурою, яка представляє собою JSON-файл, тому описано, як саме в даній базі записуються дані. Використовуючи розроблені раніше алгоритми, створено керівництво користувача, яке пояснює які дії потрібно виконувати для досягнення тих чи інших цілей. Описано технічні характеристики пристроїв необхідні для успішного запуску на них розробленого мобільного застосунку. В кінці проведено тестування, як на емуляторі, який містив мінімальні характеристики для запуску програмного забезпечення, так і на реальному пристрої з рекомендованими характеристиками. В результаті тестування були виявлені певні недоліки, які необхідно усунути.

Розроблений мобільний застосунок надає ряд переваг для користувачів, серед яких: ефективне навчання математиці, залучення до навчання, покращення математичної грамотності та вдосконаленні таких навичок як уважність, швидкість мислення та моторика пальців. Такий застосунок може бути використаний, як вдома для самостійного навчання, так і в школах як додатковий засіб навчання математики для учнів початкових класів.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Richard Garfield, George Skaff Elias, K. Robert Gutschera. Characteristics of Games. Cambridge : The MIT Press, 2020. 336p.
2. Jamie Madigan. Getting Gamers: The Psychology of Video Games and Their Impact on the People who Play Them. Lanham : Rowman & Littlefield Publishers, 2019. 320p.
3. Ялик В.І, Праворська Н.І. Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький. 2023. – 333-336 с.
4. Neal Ford, Mark Richards, Pramod Sadalage, Zhamak Dehghani. Software Architecture: The Hard Parts: Modern Trade-Off Analyses for Distributed Architectures[1st edition]. Sebastopol : O'Reilly Media, 2021. 459p.
5. Neal Ford, Mark Richards. Fundamentals of Software Architecture: An Engineering Approach[1st Edition]. Sebastopol : O'Reilly Media, 2020. 419p.
6. Joe Reis, Matt Housley. Fundamentals of Data Engineering: Plan and Build Robust Data Systems [1st edition]. Sebastopol : O'Reilly Media, 2022. 446p.
7. Jenifer Tidwell, Charles Brewer, Aynne Valencia. Designing Interfaces: Patterns for Effective Interaction Design [3rd edition]. Sebastopol : O'Reilly Media, 2020. 599p.
8. Alan Dennis, Barbara Wixom, David Tegarden. Systems Analysis and Design: An Object-Oriented Approach with UML. Hoboken : Wiley, 2020. 544p.
9. Fabio Staiano. Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop. Birmingham : Packt Publishing, 2022. 382p.
10. Abhinav Mishra. Mobile App Reverse Engineering: Get started with discovering, analyzing, and exploring the internals of Android and iOS apps. Birmingham : Packt Publishing, 2022. 166p.

					КВРІПЗ.200177.01.26.ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Ahmad Sahar. iOS 17 Programming for Beginners - Eighth Edition: Unlock the world of iOS Development with Swift 5.9, Xcode 15, and iOS 17 - Your Path to App Store Success [8th edition]. Birmingham : Packt Publishing, 2023. 604p.
12. Eric Vennaro. iOS Development at Scale: App Architecture and Design Patterns for Mobile Engineers [1st ed. edition]. New York : Apress, 2023. 676p.
13. Lawrence Schiefer, G. Blake Meike. Inside the Android OS: Building, Customizing, Managing and Operating Android System Services (Android Deep Dive)[1st edition]. Boston : Addison-Wesley Professional, 2021. 272p.
14. Cay Horstmann. Core Java: Fundamentals, Volume 1 [12th edition]. Austin : Oracle Press, 2021. 944p.
15. Herbert Schildt, Coward Danny. Java: The Complete Reference, Thirteenth Edition [13th edition]. New York : McGraw Hill, 2024. 1280p.
16. Mike Dunn, G. Meike, Pierre-Olivier Laurence, Amanda Hinchman-Dominguez. Programming Android with Kotlin: Achieving Structured Concurrency with Coroutines [1st edition]. Sebastopol : O'Reilly Media, 2022. 352p.
17. Ioannis Kostaras, Constantin Drabo, Josh Juneau, Sven Reimers, Mario Schröder, Geertjan Wielenga. Pro Apache NetBeans: Building Applications on the Rich Client Platform [1st edition]. New York : Apress, 2019. 490p.
18. Ted Hagos. Learn Android Studio 4: Efficient Java-Based Android Apps Development. New York : Apress, 2020. 324p.
19. Louis Davidson. Pro SQL Server Relational Database Design and Implementation: Best Practices for Scalability and Performance[6th ed. edition]. New York : Apress, 2020. 1183p.
20. Nabendu Biswas. Beginning React and Firebase: Create Four Beginner-Friendly Projects Using React and Firebase [1st edition]. New York : Apress, 2021. 196p.
21. Bryan Sills, Brian Gardner, Kristin Marsicano, Chris Stewart. Android Programming: The Big Nerd Ranch Guide [5th edition]. Boston : Addison-Wesley Professional, 2022. 688p.

					КВРІІІ3.200177.01.26.ІІЗ	Арк. 71
Змн.	Арк.	№ докум.	Підпис	Дата		

22. Ted Hagos, Mario Zechner, J.F. DiMarzio, Robert Green. *Beginning Android Games Development: From Beginner to Pro* [4th ed. edition]. New York : Apress. 384p.
23. Trisha Gee, Bert Bates, Kathy Sierra. *Head First Java: A Brain-Friendly Guide* [3rd edition]. Sebastopol : O'Reilly Media, 2022. 752p.
24. Jeff Friesen. *Java XML and JSON: Document Processing for Java SE* [2nd ed. edition]. New York : Apress, 2019. 546p.
25. Michael J Hernandez. *Database Design for Mere Mortals: 25th Anniversary Edition* [4th edition]. Boston : Addison-Wesley Professional, 2020. 640p.
26. Офіційний сайт Google по Firebase. URL : <https://firebase.google.com> (дата зверення: 13.05.2024)
27. Ron Itelman, Juan Cruz Viotti. *Unifying Business, Data, and Code: Designing Data Products with JSON Schema* [1st edition]. Sebastopol : O'Reilly Media, 2024. 354p.
28. Byron DeVries, Paul C. Jorgensen. *Software Testing: A Craftsman's Approach, Fifth Edition* [5th edition]. Boca Raton : Auerbach Publications, 2022. 528p.
29. Simon Amey. *Software Test Design: Write comprehensive test plans to uncover critical bugs in web, desktop, and mobile apps*. Birmingham : Packt Publishing, 2022. 426p.
30. Matthew Heusser, Michael Larsen. *Software Testing Strategies: A testing guide for the 2020s*. Birmingham : Packt Publishing, 2023. 378p.

					КВРІІЗ.200177.01.26.ІЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується згідно з процесом розробки програмного застосунку «MathMania».

1 Підстава для розробки

Підставою для розробки даного програмного застосунку є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри програмного забезпечення.

Найменування розробки: «Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор».

2 Призначення розробки

2.1 Функціональне призначення

Функціональне призначення – надати користувачам можливість покращити свої знання в області математики. Застосунок надасть користувачам можливість проходити різноманітні тренувальні вправи з базовими математичними операціями, а також перевірити набуті знання при цьому змагаючись з іншими користувачами.

2.2 Експлуатаційне призначення

Програмний застосунок має використовуватись школярами початкових класів без жодної професійної підготовки за допомогою мобільного пристрою з операційною системою Android.

3 Вимоги до програмного продукту

3.1 Вимоги до функціональних характеристик

Система має надавати користувачам такі можливості:

- можливість реєструватися та авторизуватися в застосунку, для того, щоб ці дані використовувались в подальшій роботі;
- можливість проходження тренувальних вправ, для покращення знань;

- можливість перевіряти свої знання шляхом проходження спеціально призначеної для цього своєрідної рейтингової гри;

- можливість перегляду своїх результатів, а також результатів інших користувачів застосунку.

3.2 Вимоги до надійності

Програмний застосунок повинен захищати дані користувачів і не надавати доступу для застосунку для тих користувачів які вводять не всі необхідні для авторизації дані.

Система повинна реагувати на різні дії користувачів, які можуть привести до некоректної роботи програмного застосунку – виводити різні помилки при введенні неправильних даних, а також даних які були введені у неправильному форматі.

3.3 Вимоги до складу та параметрів технічних засобів та програмної сумісності

Для того, щоб розроблений програмний застосунок міг бути запущений на мобільному пристрою користувача, він повинен відповідати таким параметрам:

- операційна система Android 8.0 і вище;
- процесор близький по характеристикам Qualcomm Snapdragon 665 Octa-core (4x Kryo 260 at 2.0 GHz & 4x Kryo 260 at 1.8 GHz);
- 2 GB RAM;
- 50 MB вільного місця на диску;
- діагональ екрана має становити 6.53 дюйма з розширенням 2340x1080 пікселів, з частотою оновлення екрану в 60 Гц та типом екрану IPS;
- підтримка мережевих стандартів 4G/LTE, 5G або Wi-Fi, для доступу в інтернет для виконання деяких функцій застосунку.

3.4 Вимоги до транспортування та зберігання

Розроблений програмний продукт може поставлятися через офіційні магазини для мобільних додатків, наприклад Google Play Store та фізичні носії.

Документація надається у двох варіантах – електронному та паперовому.
Умови експлуатації не виходять за межі умов експлуатації технічних засобів.

3.5 Спеціальні вимоги

Для коректної функціональності даного ПЗ спеціальні вимоги відсутні.

4 Вимоги до програмної документації

Замовнику має бути надана вся необхідна документація по програмному застосунку, а також технічне завдання.

Відповідно замовнику надається розроблений програмний застосунок з відповідною до нього інструкцією користувача.

5 Обґрунтування корисності розробки

Користь використання даного ПЗ полягає у виконанні процесів:

- перевірка знань учнів початкових класів в області математики;
- посилення бажання користувачів до самостійного навчання;
- можливість навчатися самостійно в будь-який час.

6 Стадії розробки

Строк виконання етапів роботи	Етап	Зміст
01.12 – 31.12.2023	Вибір тематики кваліфікаційної роботи	Розгляд тематики кваліфікаційних робіт, визначення та узгодження індивідуальної теми
01.01 – 20.02.2024	Збір необхідних матеріалів, дослідження предметної області	Дослідження предметної області, аналіз існуючих рішень та визначення вимог для розроблюваного продукту
21.02 – 20.03.2024	Проектування програмного забезпечення	Розроблення модулів програмного забезпечення, створення прототипів інтерфейсу користувача, вибір технологій необхідних для розробки.

21.03 – 30.04.2024	Програмна реалізація з використанням відповідних засобів розробки. Тестування ПЗ	Реалізація раніше розроблених модулів та інтерфейсу користувача з використанням обраних інструментів, тестування програмного застосунку.
01.05 – 25.05.2024	Створення необхідної документації	Оформлення пояснювальної записки до розробленого у КвР програмного застосунку.
Травень 2024	Попередній захист КвР	Попередній захист, для отримання рекомендацій з покращення пояснювальної записки
26.05 – 30.05.2024	Отримання документів від керівників	Отримання необхідних документів для затвердження кваліфікаційної роботи
з 11.06.2024	Захист КвР	Підготовка до захисту та захист КвР

7 Порядок контролю та приймання

Контроль і приймання здійснюється користувачами програмного застосунку, та керівниками проекту. Також перед цим повинні проводитись всі необхідні тестування розробленого продукту, для перевірки того чи всі його функції виконуються правильно та чи відповідають вони визначеним у технічному завданні вимогам.

ДОДАТОК Б (обов'язковий)

КОД(ЛІСТИНГ) ПРОГРАМИ

Код класу LoadActivity

```
package com.example.myapplication;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;

public class LoadActivity extends AppCompatActivity {
    EditText userNameText, passwordText;
    public static String userName;
    private DatabaseReference mDataBase;
    User newUser;
    private String KEY = "Users";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.load_screen);
        userNameText = (EditText) findViewById(R.id.UserName);
        passwordText = (EditText) findViewById(R.id.Password);
        mDataBase = FirebaseDatabase.getInstance().getReference(KEY);
    }
    public void authorize_click(View v)
    {
        if(!isNetworkAvailable()) {
            return;
        }
        String name = userNameText.getText().toString();
        String password = passwordText.getText().toString();
        if(name.isEmpty() || password.isEmpty()) {
            Toast.makeText(LoadActivity.this, "Будь ласка, введіть ім'я" +
                " користувача та пароль", Toast.LENGTH_SHORT).show();
            return;
        }
        userName = name;
        Query checkUser = mDataBase.orderByChild("name").equalTo(name);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    for (DataSnapshot userSnapshot : snapshot.getChildren())
```

```

        String passwordFromDB =
userSnapshot.child("password").getValue(String.class);
        if (passwordFromDB.equals(password)) {
            Intent intent = new Intent(LoadActivity.this,
MenuActivity.class);
            startActivity(intent);
        } else {
            Toast.makeText(LoadActivity.this, "Невірний
пароль", Toast.LENGTH_SHORT).show();
            return;
        }
    }
    } else {
        Toast.makeText(LoadActivity.this, "Користувача не
знайдено", Toast.LENGTH_SHORT).show();
        return;
    }
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
    return;
}
});
}
public void register_click(View v)
{
    if(!isNetworkAvailable()) {
        return;
    }
    String name = userNameText.getText().toString();
    userName = name;
    String password = passwordText.getText().toString();
    if(name.isEmpty() || password.isEmpty()) {
        Toast.makeText(LoadActivity.this, "Будь ласка, введіть ім'я" +
" користувача та пароль", Toast.LENGTH_SHORT).show();
        return;
    }
    Query checkUser = mDataBase.orderByChild("name").equalTo(name);

    checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.exists()) {
                Toast.makeText(LoadActivity.this, "Користувач з таким
іменем вже існує", Toast.LENGTH_SHORT).show();
                return;
            } else {
                newUser = new User(name, password);
                mDataBase.push().setValue(newUser);
                Intent intent = new Intent(LoadActivity.this,
MenuActivity.class);
                startActivity(intent);
            }
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        return;
    }
});
}
private boolean isNetworkAvailable() {
    ConnectivityManager connectivityManager

```

```

        = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetworkInfo =
connectivityManager.getActiveNetworkInfo();
        return activeNetworkInfo != null && activeNetworkInfo.isConnected();
    }
}

```

Код класу MenuActivity

```

package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MenuActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);
    }
    public void plusClick(View v)
    {
        Intent intent = new Intent(this, PlusTrainingActivity.class);
        startActivity(intent);
    }
    public void minusclick(View v)
    {
        Intent intent = new Intent(this, MinusTrainingActivity.class);
        startActivity(intent);
    }
    public void multiplyClick(View v)
    {
        Intent intent = new Intent(this, MultiplyTrainingActivity.class);
        startActivity(intent);
    }
    public void ratingClick(View v)
    {
        Intent intent = new Intent(this, RatingGameActivity.class);
        startActivity(intent);
    }
    public void divideClick(View v){
        Intent intent = new Intent(this, DivideTrainingActivity.class);
        startActivity(intent);
    }
    public void leadersClick(View view){
        Intent intent = new Intent(this, ScoreTableActivity.class);
        startActivity(intent);
    }
    public void rulesClick(View view){
        Intent intent = new Intent(this, RulesActivity.class);
        startActivity(intent);
    }
}

```

Код класу GameSupport

```

package com.example.myapplication;
import android.app.Activity;
import android.content.DialogInterface;
import android.os.CountDownTimer;

```

```

import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AlertDialog;

public class GameSupport {
    int first_num;
    int second_num;
    public int RandomNumber(int first, int second){
        int a = first;
        a = (int) (Math.random()*second + 1);
        return a;
    }
    public static boolean isNumber(String mes){
        try
        {
            Integer.parseInt(mes);
            return true;
        }
        catch (NumberFormatException e)
        {
            return false;
        }
    }
    public void Wr_answer(int value, Activity activity, TextView label,
        CountDownTimer timer, EditText answer, String symbol)
    {
        timer.cancel();
        AlertDialog.Builder builder = new AlertDialog.Builder(activity);
        builder.setTitle("Неправильна відповідь!")
            .setMessage("Відповідь була неправильною! Правильна
відповідь: " + value)
            .setCancelable(false)
            .setNegativeButton("OK",
                new DialogInterface.OnClickListener() {
                    id)
                    public void onClick(DialogInterface dialog, int
                        {
                            generate(label, symbol);
                            answer.setText("");
                            timer.start();
                            dialog.cancel();
                        }
                    });
        AlertDialog alert = builder.create();
        alert.show();
    }
    public void no_value(Activity activity)
    {
        AlertDialog.Builder builder = new AlertDialog.Builder(activity);
        builder.setTitle("Помилка!")
            .setMessage("Не було введено значення")
            .setCancelable(false)
            .setNegativeButton("OK",
                new DialogInterface.OnClickListener() {
                    id)
                    public void onClick(DialogInterface dialog, int
                        {
                            dialog.cancel();
                        }
                    });
        AlertDialog alert = builder.create();
        alert.show();
    }
    public void Time_out(Activity activity, TextView label, CountDownTimer

```

```

timer, String symbol)
{
    AlertDialog.Builder builder = new AlertDialog.Builder(activity);
    builder.setTitle("Час вийшов")
        .setMessage("Час відведений на виконання завдання
закінчився.")
        .setCancelable(false)
        .setNegativeButton("OK",
            new DialogInterface.OnClickListener() {
id)                public void onClick(DialogInterface dialog, int
                    {
                        generate(label, symbol);
                        timer.start();
                        dialog.cancel();
                    }
                });
    AlertDialog alert = builder.create();
    alert.show();
}
public void no_number(Activity activity, EditText answer)
{
    AlertDialog.Builder builder = new AlertDialog.Builder(activity);
    builder.setTitle("Помилка!")
        .setMessage("Введенне значення не є числом")
        .setCancelable(false)
        .setNegativeButton("OK",
id)                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
                        {
                            dialog.cancel();
                            answer.setText("");
                        }
                    });
    AlertDialog alert = builder.create();
    alert.show();
}

public void correct(Activity activity, CountdownTimer timer, TextView
label, EditText answer, String symbol)
{
    timer.cancel();
    AlertDialog.Builder builder = new AlertDialog.Builder(activity);
    builder.setTitle("Вітаю!")
        .setMessage("Введенне значення було вірним")
        .setCancelable(false)
        .setNegativeButton("OK",
id)                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
                        {
                            generate(label, symbol);
                            answer.setText("");
                            timer.start();
                            dialog.cancel();
                        }
                    });
    AlertDialog alert = builder.create();
    alert.show();
}
}

```

```

    public void Check(String answer, Activity activity, CountdownTimer timer,
TextView label, EditText w_answer, String symbol) {
    int user_answer = 0;
    int correct_answer = 0;

    if(isNumber(answer))
    {
        if (symbol == "+")
        {
            user_answer = Integer.parseInt(answer);
            correct_answer = first_numb + second_numb;
        }
        if (symbol == "-")
        {
            if(second_numb > first_numb)
            {
                user_answer = Integer.parseInt(answer);
                correct_answer = second_numb - first_numb;
            }
            else
            {
                user_answer = Integer.parseInt(answer);
                correct_answer = first_numb - second_numb;
            }
        }
        if (symbol == "x")
        {
            user_answer = Integer.parseInt(answer);
            correct_answer = first_numb * second_numb;
        }
        if (symbol == "÷")
        {
            user_answer = Integer.parseInt(answer);
            correct_answer = first_numb / second_numb;
        }

        if (user_answer == correct_answer)
        {
            correct(activity, timer, label, w_answer, symbol);
        } else Wr_answer(correct_answer, activity, label, timer,
w_answer, symbol);
        }
        else no_number(activity, w_answer);
    }
    public void generate(TextView label, String symbol)
    {
        if (symbol == "+") {
            first_numb = RandomNumber(0, 100);
            second_numb = RandomNumber(0, 100);
            label.setText(first_numb + "+" + second_numb + "=");
        }
        if (symbol == "-") {
            first_numb = RandomNumber(0, 100);
            second_numb = RandomNumber(0, 100);
            label.setText(Lower(first_numb, second_numb));
        }
        if (symbol == "x")
        {
            first_numb = RandomNumber(0,10);
            second_numb = RandomNumber(0,10);
            label.setText(first_numb + "x" + second_numb + "=");
        }
        if (symbol == "÷")
        {

```

```

        second_numb = RandomNumber(0, 10);
        first_numb = second_numb * RandomNumber(0,10);
        label.setText(first_numb + "÷" + second_numb + "=");
    }
}
public String Lower(int a, int b){
    int temporary;
    String text = "";
    if(b > a)
    {
        temporary = b;
        text = temporary + "-" + a + "=";

    }
    else
    {
        text = a + "-" + b + "=";
    }

    return text;
}
}

```

Код класу PlusTrainingActivity

```

package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class PlusTrainingActivity extends AppCompatActivity {
    TextView textView_plus;
    TextView timer_text;
    EditText user_answer;
    GameSupport gameSupport = new GameSupport();
    CountDownTimer new_timer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_plus_training);
        timer_text = (TextView) findViewById(R.id.TimerText);
        textView_plus = (TextView) findViewById(R.id.Plus_label);
        gameSupport.generate(textView_plus, "+");
        new_timer = new CountDownTimer(10*1500, 1000) {
            @Override
            public void onTick(long l) {
                timer_text.setText("0." + Long.toString(l/1000));
            }
            @Override
            public void onFinish()
            {
                gameSupport.Time_out(PlusTrainingActivity.this,
                textView_plus, new_timer, "+");
            }
        };
        new_timer.start();
    }
    public void backClick(View v)
    {

```

```

        Intent intent = new Intent(this, MenuActivity.class);
        startActivity(intent);
    }
    public void check_click(View v)
    {
        user_answer = (EditText) findViewById(R.id.PlAnswer);
        if(user_answer.getText().toString().length() < 1) {
            gameSupport.no_value(PlusTrainingActivity.this);
        } else gameSupport.Check(user_answer.getText().toString(),
            PlusTrainingActivity.this, new_timer, textView_plus,
            user_answer, "+");
        }
    }
}

```

Код класу RatingGameActivity

```

package com.example.myapplication;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;

public class RatingGameActivity extends AppCompatActivity {
    TextView textView_rating;
    TextView timer_text;
    EditText user_answer;
    int temporary;
    int first_num;
    int second_num;

    int counter = 0;
    GameSupport gameSupport = new GameSupport();
    char[] symbols = new char[] {'+', '-', 'x', '÷'};

    int sign;
    CountDownTimer new_timer;
    private DatabaseReference mDataBase1;
    private String KEY = "Leaders";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rating_game);
        mDataBase1 = FirebaseDatabase.getInstance().getReference(KEY);
        timer_text = (TextView) findViewById(R.id.ratingTimer);
        sign = (int) (Math.random() * 4);
        if(symbols[sign] == '+')
        {
            first_num = gameSupport.RandomNumber(0,100);
            second_num = gameSupport.RandomNumber(0,100);
            textView_rating = (TextView) findViewById(R.id.rating_label);
        } else if(symbols[sign] == '-')

```

```

    {
        first_num = gameSupport.RandomNumber(0,100);
        second_num = gameSupport.RandomNumber(0,100);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(Lower(first_num, second_num));
    } else if (symbols[sign] == 'x')
    {
        first_num = gameSupport.RandomNumber(0,10);
        second_num = gameSupport.RandomNumber(0,10);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "x" + second_num + "=");
    } else if (symbols[sign] == '/')
    {
        second_num = gameSupport.RandomNumber(0, 10);
        first_num = second_num * gameSupport.RandomNumber(0,10);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "/" + second_num + "=");
    }
    new_timer = new CountdownTimer(10 * 3000, 1000) {
        @Override
        public void onTick(long l) {
            timer_text.setText("0." + Long.toString(l / 1000));
        }

        @Override
        public void onFinish() {
            Time_out();
        }
    };
    new_timer.start();
}
public String Lower(int a, int b){
    String text = "";
    if(b > a)
    {
        temporary = b;
        text = temporary + "-" + a + "=";
    }
    else
    {
        text = a + "-" + b + "=";
    }
    return text;
}
public void check_click(View v)
{
    user_answer = (EditText) findViewById(R.id.rateAnswer);
    if(user_answer.getText().toString().length() < 1) {
        gameSupport.no_value(RatingGameActivity.this);
    } else Check(user_answer.getText().toString(), symbols[sign]);
}
public void Check(String answer, char value) {
    int ans;
    if(gameSupport.isNumber(answer))
    {
        ans = Integer.parseInt(answer);
        if(value == '+')
        {
            int t_ans = first_num * second_num;
            if (ans == t_ans)
            {
                correct();
            } else {
                Wr_answer();
            }
        }
    }
}

```

```

    }
    } else if (value == '-')
    {
        if(first_numb > second_numb) {
            int t_ans = first_numb - second_numb;
            if (ans == t_ans)
            {
                correct();
            } else {
                Wr_answer();
            }
        } else if (second_numb > first_numb) {
            int t_ans = second_numb - first_numb;
            if (ans == t_ans)
            {
                correct();
            } else {
                Wr_answer();
            }
        }
    }
    } else if (value == 'x')
    {
        int t_ans = first_numb * second_numb;
        if (ans == t_ans)
        {
            correct();
        } else {
            Wr_answer();
        }
    }
    } else if(value == '÷')
    {
        int t_ans = first_numb / second_numb;
        if (ans == t_ans)
        {
            correct();
        } else {
            Wr_answer();
        }
    }
    }
    else gameSupport.no_number(RatingGameActivity.this, user_answer);
}
public void correct()
{
    user_answer.setText("");
    counter++;
    sign = (int) (Math.random() * 4);
    if(symbols[sign] == '+')
    {
        first_numb = gameSupport.RandomNumber(0,100);
        second_numb = gameSupport.RandomNumber(0,100);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_numb + "+" + second_numb + "=");
    } else if(symbols[sign] == '-')
    {
        first_numb = gameSupport.RandomNumber(0,100);
        second_numb = gameSupport.RandomNumber(0,100);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(Lower(first_numb, second_numb));
    } else if(symbols[sign] == 'x')
    {
        first_numb = gameSupport.RandomNumber(0,10);
        second_numb = gameSupport.RandomNumber(0,10);
    }
}

```

```

        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "x" + second_num + "=");
    } else if (symbols[sign] == '÷')
    {
        second_num = gameSupport.RandomNumber(0, 10);
        first_num = second_num * gameSupport.RandomNumber(0, 10);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "÷" + second_num + "=");
    }
}
public void Wr_answer()
{
    user_answer.setText("");
    sign = (int) (Math.random() * 4);
    if (symbols[sign] == '+')
    {
        first_num = gameSupport.RandomNumber(0, 100);
        second_num = gameSupport.RandomNumber(0, 100);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "+" + second_num + "=");
    } else if (symbols[sign] == '-')
    {
        first_num = gameSupport.RandomNumber(0, 100);
        second_num = gameSupport.RandomNumber(0, 100);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(Lower(first_num, second_num));
    } else if (symbols[sign] == 'x')
    {
        first_num = gameSupport.RandomNumber(0, 10);
        second_num = gameSupport.RandomNumber(0, 10);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "x" + second_num + "=");
    } else if (symbols[sign] == '÷')
    {
        second_num = gameSupport.RandomNumber(0, 10);
        first_num = second_num * gameSupport.RandomNumber(0, 10);
        textView_rating = (TextView) findViewById(R.id.rating_label);
        textView_rating.setText(first_num + "÷" + second_num + "=");
    }
}
public void Time_out()
{
    AlertDialog.Builder builder = new
AlertDialog.Builder(RatingGameActivity.this);
    builder.setTitle("Гру завершено")
        .setMessage("Ваш пахунок: " + counter)
        .setCancelable(false)
        .setNegativeButton("OK",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int
id)
                {
                    int score = counter;
                    UserAchievement userAchievement = new
UserAchievement(LoadActivity.userName, score);
                    Query checkUser =
mDataBase1.orderByChild("name").equalTo(LoadActivity.userName);

                    checkUser.addListenerForSingleValueEvent(new
ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull
DataSnapshot snapshot) {
                            if (snapshot.exists()) {

```

```

                                for (DataSnapshot userSnapshot :
snapshot.getChildren()) {
                                int currentScore =
userSnapshot.child("score").getValue(Integer.class);
                                if (score > currentScore) {
userSnapshot.getRef().child("score").setValue(score);
                                }
                                } else {
mDataBasel.push().setValue(userAchievement);
                                }
                                dialog.cancel();
                                Intent intent = new
Intent(RatingGameActivity.this, MenuActivity.class);
                                startActivity(intent);
                                }
                                @Override
                                public void onCancelled(@NonNull
DatabaseError error) {
                                return;
                                }
                                });
                                });
                                AlertDialog alert = builder.create();
                                alert.show();
                                }
}

```

Код класу RulesActivity

```

package com.example.myapplication;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class RulesActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rules);
        TextView rules_TextView = (TextView)
findViewById(R.id.rules_TextView);
        String rules_text = "Тут наведено основні правила використання даного
мобільного застосунку.\n\n" +
        "Всього доступно 4 тренувальні вправи на: додавання,
віднімання, множення та ділення чисел.\n\n" +
        "Також можна пройти рейтингову гру на час де є змога
перевірити набуті знання.\n\n" +
        "По завершенню проходження даної гри кількість правильних
відповідей буде записана" +
        "та буде можливість переглянути її в таблиці лідерів.";
        rules_TextView.setText(rules_text);
    }
    public void backClick(View view)
    {
        Intent intent = new Intent(this, MenuActivity.class);
        startActivity(intent);
    }
}

```

```
    }  
}
```

Код класу ScoreTableActivity

```
package com.example.myapplication;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.List;  
  
public class ScoreTableActivity extends AppCompatActivity {  
    private List<UserAchivement> myList;  
    TextView userText;  
    TextView userScore;  
    private String user;  
    private String score;  
    private DatabaseReference mDataBase;  
    private String KEY = "Leaders";  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_score_table);  
        userText = (TextView) findViewById(R.id.Users);  
        userScore = (TextView) findViewById(R.id.Score);  
        myList = new ArrayList<>();  
        mDataBase = FirebaseDatabase.getInstance().getReference(KEY);  
        getData();  
    }  
    private void getData()  
    {  
        ValueEventListener valueEventListener = new ValueEventListener() {  
            @Override  
            public void onDataChange(@NonNull DataSnapshot snapshot) {  
                if (myList.size() > 0) myList.clear();  
                for (DataSnapshot ds : snapshot.getChildren())  
                {  
                    UserAchivement userAchivement =  
ds.getValue(UserAchivement.class);  
                    myList.add(userAchivement);  
                }  
                Collections.sort(myList, new Comparator<UserAchivement>() {  
                    @Override  
                    public int compare(UserAchivement u1, UserAchivement u2)  
{  
                        return Integer.compare(u2.score, u1.score);  
                    }  
                });  
                user = "User" + "\n\n";  
                score = "Score" + "\n\n";  
            }  
        }  
    }  
}
```

```

int limit = Math.min(myList.size(), 5);
for (int i = 0; i < limit; i++) {
    UserAchivement us = myList.get(i);
    String icon = "";
    switch (i) {
        case 0:
            icon = "🥇"; // Золота медаль для першого місця
            break;
        case 1:
            icon = "🥈"; // Срібна медаль для другого місця
            break;
        case 2:
            icon = "🥉"; // Бронзова медаль для третього місця
            break;
    }
    user += icon + us.name + "\n\n";
    score += us.score + "\n\n";
}
userText.setText(user);
userScore.setText(score);
}
@Override
public void onCancelled(@NonNull DatabaseError error) {

}
};
mDataBase.addValueEventListener(valueEventListener);
}
public void backClick(View view)
{
    Intent intent = new Intent(this, MenuActivity.class);
    startActivity(intent);
}
}
}

```

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ
КВАЛІФІКАЦІЙНА РОБОТА
НА ТЕМУ
ПРОГРАМНИЙ ЗАСТОСУНОК ДЛЯ НАВЧАННЯ
МАТЕМАТИЦІ УЧНІВ ПОЧАТКОВИХ КЛАСІВ З
ВИКОРИСТАННЯМ ІНТЕРАКТИВНИХ ІГОР

Розробка велась:

Здобувач: Ялик Вадим Ігорович

Керівник: Праворська Наталія Іванівна, канд. пед. наук, доцент

МЕТА ТА ЗАВДАННЯ

Метою кваліфікаційної роботи є розробити програмний застосунок, який зможе надати учням початкових класів можливість перевірити свої знання а також вдосконалити математичні навички.

Згідно мети поставлені такі завдання:

- дослідити та проаналізувати особливості обраної предметної області;
- розглянути існуючі рішення;
- обрати архітектурні рішення та технології для розробки застосунку;
- на основі обраних рішень, здійснити проектування та програмну реалізацію застосунку;
- провести тестування створеного продукту для виявлення проблем в його роботі, та перевірки того чи відповідає він заявленим вимогам.

АКТУАЛЬНІСТЬ

Актуальність теми даної кваліфікаційної роботи визначається наступними чинниками:

Покращення математичної грамотності – оскільки математика є ключовим предметом, і розробка програмних застосунків, спрямованих на підвищення знань та навичок в даній області, має велике значення для покращення якості освіти.

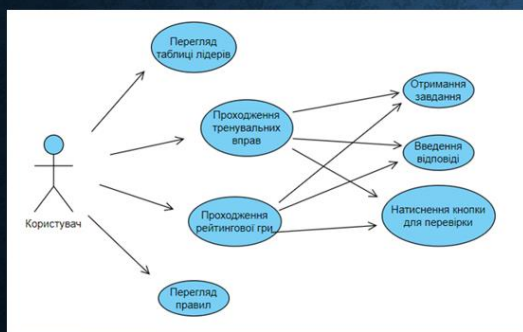
Зручний доступ до навчального матеріалу – програмні застосунки забезпечують можливість навчатися в будь-який час і місці, що підтримує самостійне навчання та адаптується до індивідуального темпу учня, особливо, якщо це застосунок для мобільних платформ, розробка якого буде вестись в даній кваліфікаційній роботі.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Серед існуючих рішень які відповідають тематиці даної кваліфікаційної роботи, розглядалися такі застосунки як «Math Workout» та «Math Brain». В результаті їх розгляду була сформована така таблиця, що порівнює основні їх елементи:

Критерій	Math Workout	Math Brain
Інтерфейс	Простий, інтуїтивно зрозумілий	Середній, інтуїтивно зрозумілий
Дизайн	Простий без використання різких кольорів	Хороший з використанням різних кольорів
Досягнення цілі	Добре досягає основної цілі	Добре досягає основної цілі
Переваги	Відповідь можна ввести самому	Вибір відповіді дається з певного набору
Недоліки	Рекламні банери	Необхідність VIP-підписки

ВАРІАНТИ ВИКОРИСТАННЯ



Після аналізу предметної області, а також розгляду існуючих рішень, було визначено які основні дії повинен виконувати користувач у розроблюваному програмному застосунку. Їх відповідно подано на даній діаграмі варіантів використання.

ВИМОГИ

Відповідно до раніше отриманих з дослідження даних та визначення того що користувач повинен робити у програмному застосунку можна визначити ряд наступних вимог:

Функціональні вимоги:

- застосунок повинен надавати можливість користувачеві попрактикуватись у своїх знаннях та надавати можливість отримати правильну відповідь якщо вона такою не була;
- застосунок повинен надавати користувачеві можливість перевірити свої знання шляхом визначення того наскільки швидко він може вирішувати математичні приклади;
- застосунок повинен зберігати дані користувача разом з його результатами і відображати їх при необхідності, як і результати інших користувачів для забезпечення певної мотивації досягати кращих результатів.

Нефункціональні вимоги:

- застосунок повинен працювати швидко та безперебійно, навіть при великій кількості користувачів та високих навантаженнях на сервер;
- інтерфейс застосунку повинен бути легким та інтуїтивно зрозумілим, щоб користувачі могли легко зорієнтуватися та користуватися додатком без підготовки.

АРХІТЕКТУРА

Для побудови архітектури розроблюваного програмного застосунку було обрано такий шаблон проектування як MVC.

Шаблон MVC (Model-View-Controller) є одним з популярніших підходів до проектування програмного забезпечення, включаючи мобільні застосунки. Він надає змогу відділити логічну частину застосунку від його візуального представлення та управління взаємодією з користувачем

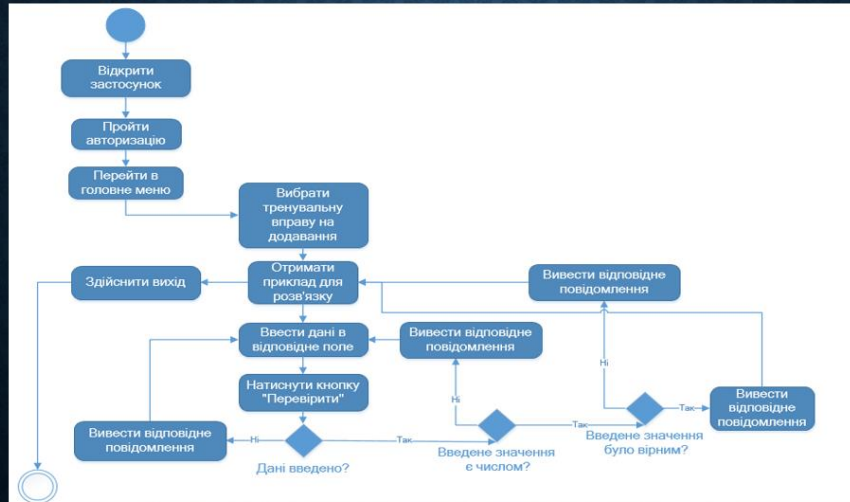
Також для застосунку було обрано такий шаблон проектування як **Servant** суть якого полягатиме в зменшенні коду в застосунку та реалізації деяких його функцій.

ПРОЄКТУВАННЯ БАЗИ ДАНИХ

Для збереження певних даних користувачів, щоб забезпечити більшу зручність користуванням, а також для того щоб могла виконуватись одна з його функцій, необхідним було розробити базу даних яка б зберігала необхідні дані. В результаті було розроблено такі таблиці бази даних:

Назва поля	Тип поля	Обмеження	Коментар	Назва поля	Тип поля	Обмеження	Коментар
Name	VARCHAR(20)	NOT NULL	Ім'я користувача, та його ідентифікатор	Name	VARCHAR(20)	NOT NULL	Ім'я користувача
Password	VARCHAR(20)	NOT NULL	Пароль	Score	INT	NOT NULL	Рахунок користувача

ЛОГІКА РОБОТИ



ВИБІР ТЕХНОЛОГІЙ

Розробка мобільного застосунку в даній кваліфікаційній роботі ведеться для операційної системи **Android**.

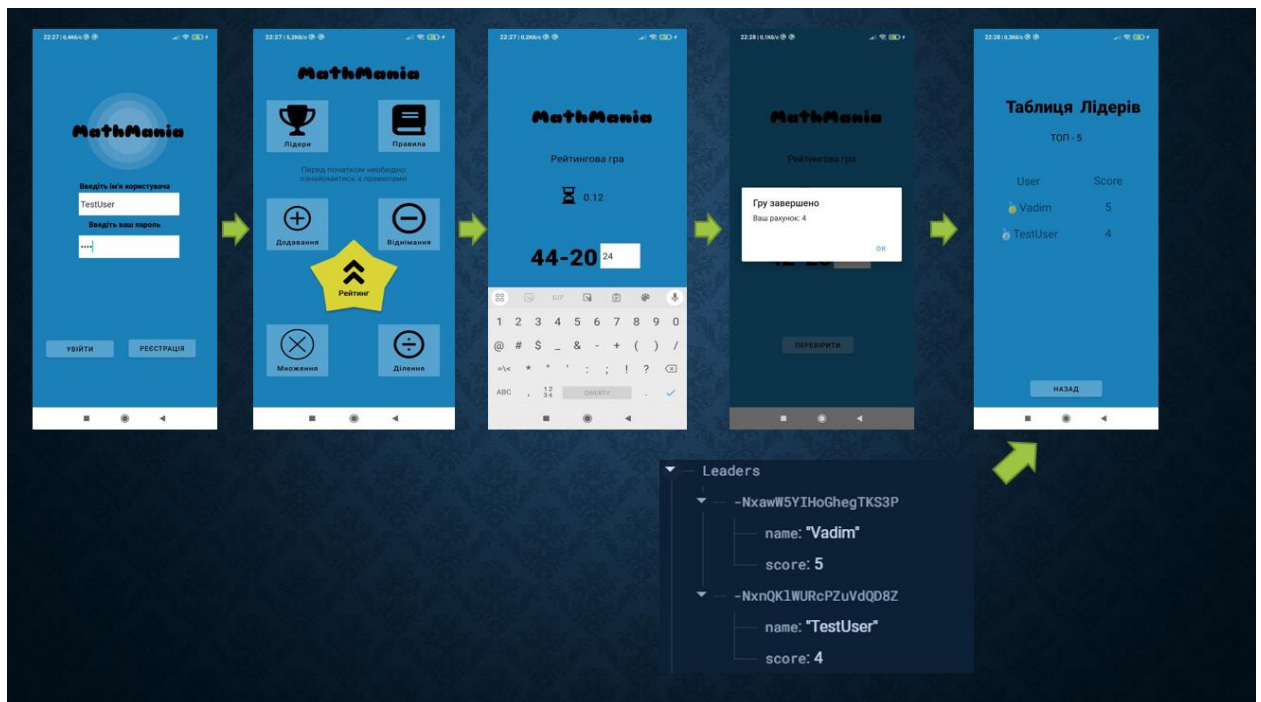
Серед мов програмування, на яких можна розробляти мобільні застосунки для даної операційної системи, було обрано мову програмування **Java**.

Серед середовищ розробки які використовують обрану мову та на яких можна розробити застосунок для обраної операційної системи, обрано **Android Studio**.

Для реалізації спроектованої бази даних було обрано такий інструмент як **Firebase**.

РОЗРОБКА ТА ТЕСТУВАННЯ

Після розробки архітектури мобільного застосунку та вказання логіки його роботи, на основі цих даних виконується розробка самого застосунку. Поетапно відбуваються розробка модулів застосунку, створення графічного інтерфейсу користувача з яким будуть працювати розроблені модулі та створення таблиць в базі даних, де будуть зберігатись відомості користувачів та їх результати. Після розробки, застосунок тестується, для перевірки того чи все працює правильно, та чи відповідає зазначеним вимогам. Тестування здійснюється як на емуляторі який надає обране середовище розробки, так і на реальному фізичному пристрої.



Розробка також була висвітлена в тезах Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023»

Знайти її можна за таким посиланням: Ялик В.І, Праворська Н.І. Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький. 2023. – 333-336 с.

ВИСНОВКИ

При виконанні кваліфікаційної роботи на тему «Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор» було спроектовано та реалізовано функціонал програмного забезпечення що є застосунком для мобільних платформ і є своєрідною грою.

Використано шаблон проектування MVC та **Servlet**, мову програмування **Java**, середовище **Android Studio** та базу даних **Firebase**.

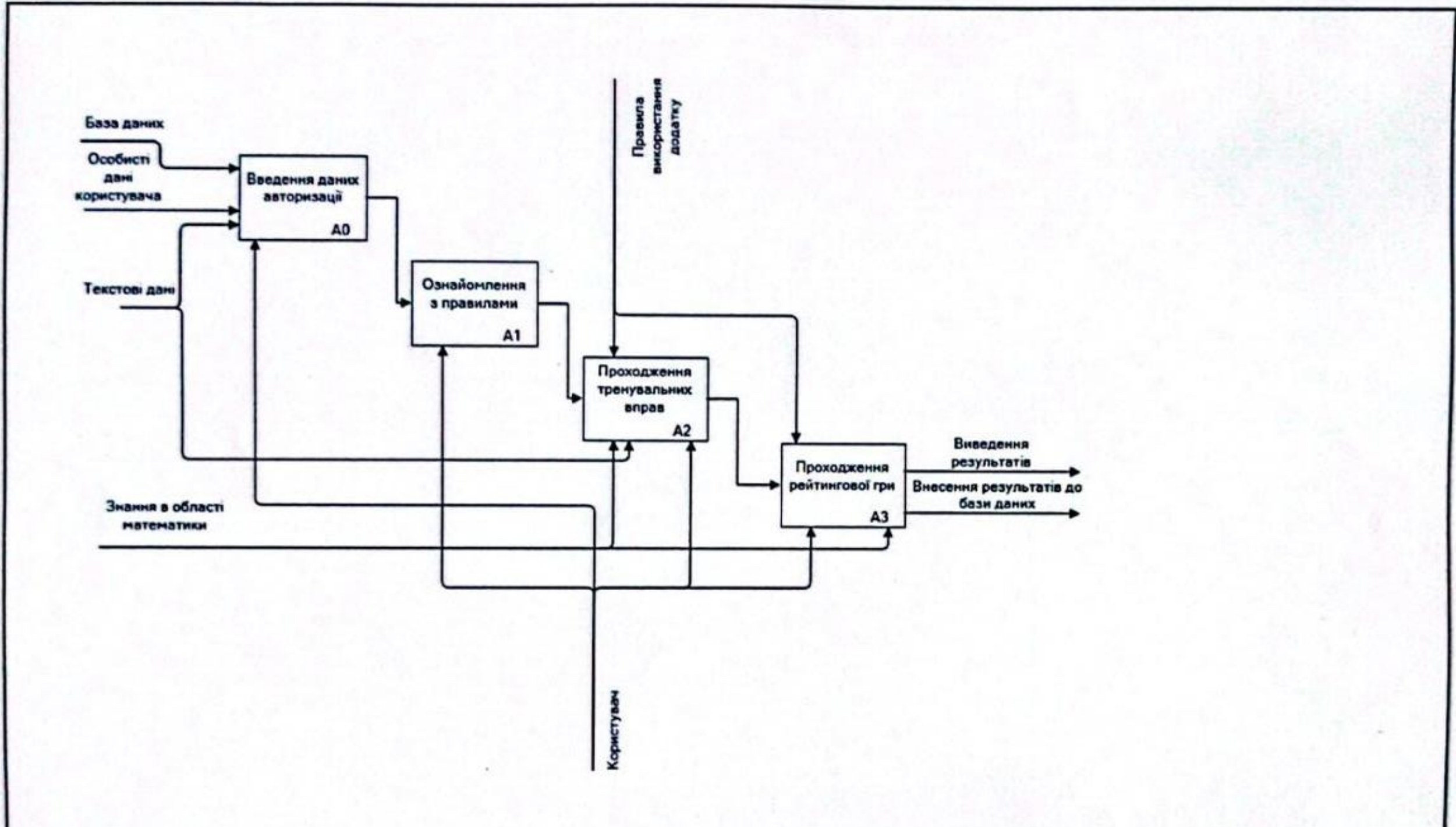
Корисність даної роботи полягає в тому, що завдяки розробленому в ній програмному застосунку користувачі зможуть перевірити рівень свої знань в області математики та покращити їх.

При виконанні роботи було створено програмний застосунок, пояснювальну записку та три креслення.

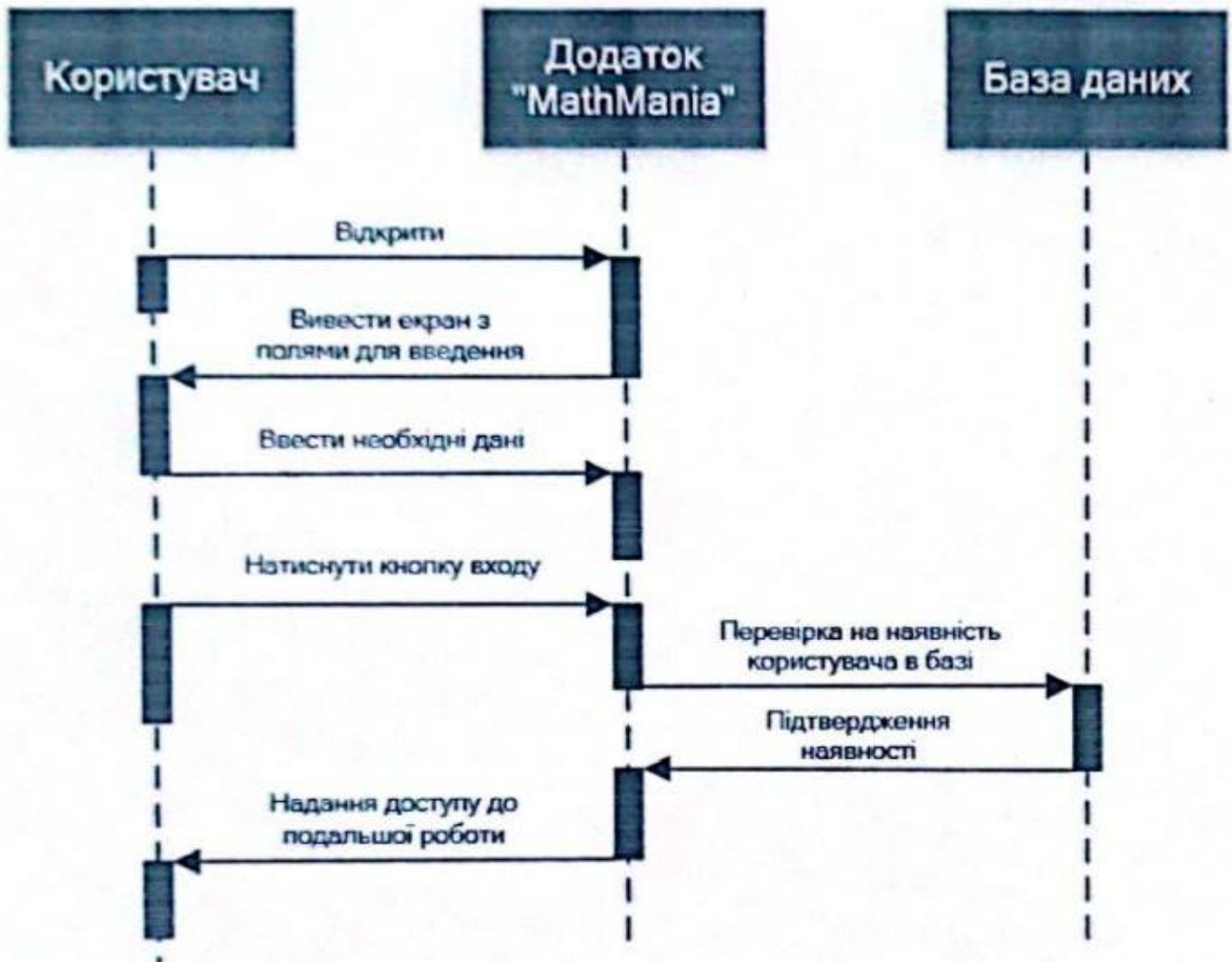
Розроблений програмний застосунок відповідає всім вимогам та виконує усі необхідні функції, проте потребує подальшого вдосконалення.

ДЯКУЮ ЗА УВАГУ

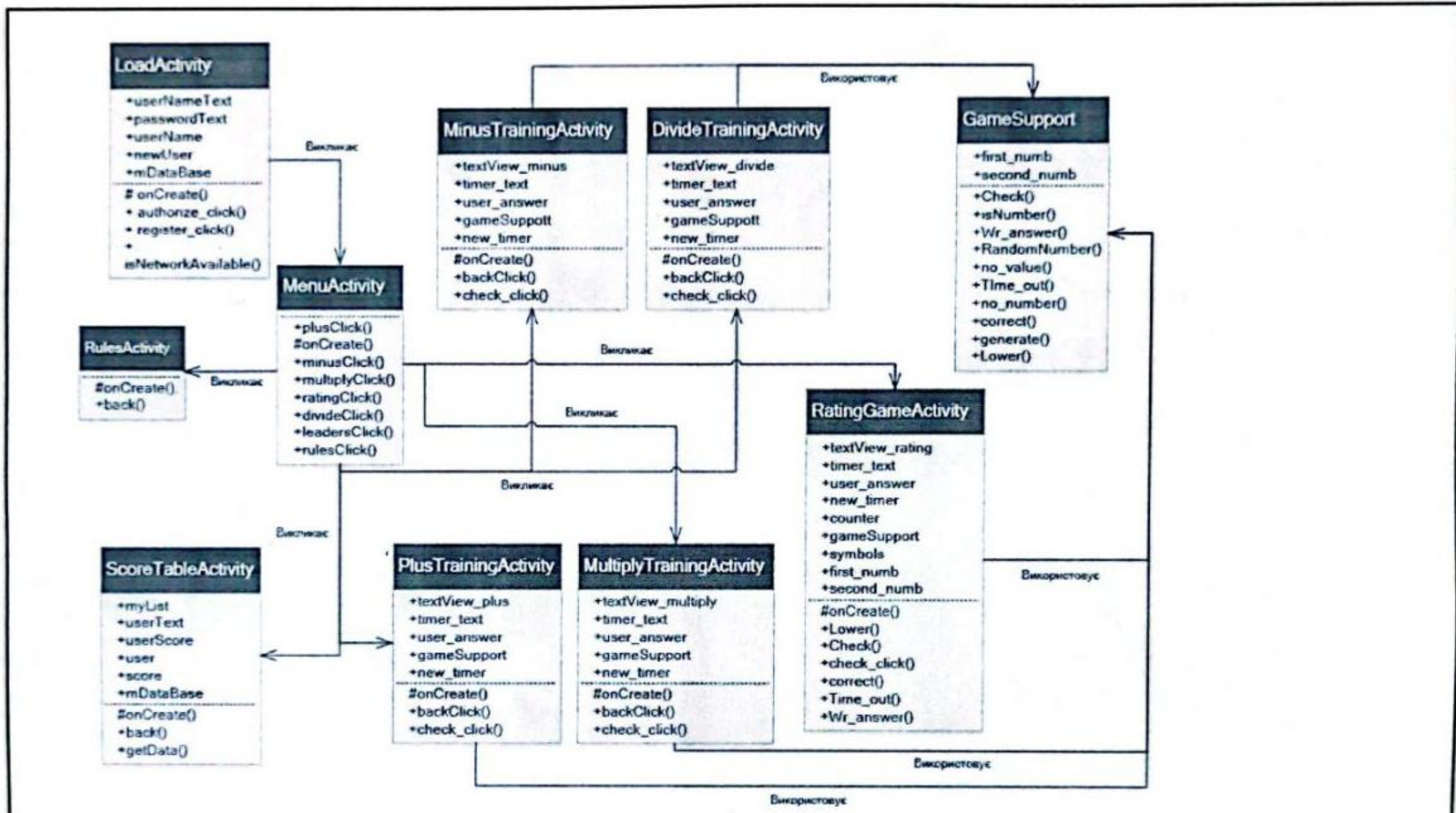
ГРАФІЧНА ЧАСТИНА



					КвРІПЗ.200177.01.26.E8		
Зм.	Арк.	№ докум.	Підпис	Дата	Літера	Маса	Месцяб
Розробив		Ялик В.І	<i>[Signature]</i>	26.02			
Керівник		Праворська Н.	<i>[Signature]</i>				
Консульт.					Аркуш 1	Аркуш 3	
Н. Контр.		Яшина О.М	<i>[Signature]</i>	26.02			
Зав. каф.		Бедратюк Л.В	<i>[Signature]</i>	26.02			



					КвРІПЗ.200177.01.26.Е8			
Зм	Арк.	№ докум.	Підпис	Дата	Програмний застосунок для навчання математики учнів початкових класів з використанням інтерактивних ігор	Літера	Маса	Масштаб
Розробив		Ялик В.І.	<i>[Signature]</i>	16.01				
Керівник		Праворська Н.	<i>[Signature]</i>		UML-діаграма послідовності авторизації	Аркуш 2	Аркушів 3	
Н. Контр.		Яшина О.М.	<i>[Signature]</i>					
Зав. каф.		Бедратюк Л.Л.	<i>[Signature]</i>					



					КвРІПЗ.200177.01.26.Е8					
Зм.	Арк.	№ докум.	Підпис	Дата	Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор			Літера	Маса	Масштаб
Розробив		Ялик В.І	<i>[Signature]</i>	<i>[Date]</i>						
Керівник		Праворська І.І	<i>[Signature]</i>							
Консульт.					UML-діаграма класів застосунку			Аркуш 3	Аркушів 3	
Н. Контр.		Яшина О.М	<i>[Signature]</i>	<i>[Date]</i>						
Зав. каф.		Бедратюк Л.Я	<i>[Signature]</i>	<i>[Date]</i>						

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Ялика В.І

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-20-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті», згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

27.05.2024
Дата


Підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 8%

ID: 128466 Назва: БКР Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор Додано в БД: 2024-06-05 Автора: Ялик Вадим Керівники: ПРАВОРСЬКА Н.І., Консультанти: канд. пед. наук, доцент Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	107653	739	5534 (5%)	63 (9%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Ім'я користувача:
ІПЗ

Дата перевірки:
06.06.2024 00:22:09 EEST

Дата звіту:
06.06.2024 11:30:45 EEST

ID перевірки:
1016322698

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012953

Назва документа: БКР_Програмний застосунок для навчання математиці учнів початкових класів з використа...

Кількість сторінок: 68 Кількість слів: 16138 Кількість символів: 118678 Розмір файлу: 1.75 MB ID файлу: 1016121133

7.05% Схожість

Найбільша схожість: 2.4% з джерелом з Бібліотеки (ID файлу: 1016121087)

5.26% Джерела з Інтернету 742 Сторінка 70

3.4% Джерела з Бібліотеки 213 Сторінка 74

0.14% Цитат

Цитати 2 Сторінка 75

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Ялик Вадим Ігорович

Тема Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 ; кількість сторінок записки 62

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі здійснено дослідження та аналіз предметної області, розглянуто існуючі на ринку рішення, та відповідно встановлено їх переваги та недоліки, на основі яких було визначено вимоги до програмного забезпечення. Також було розглянуто перелік технологій необхідних для розробки, розроблено структуру програмного продукту та прототип інтерфейсу користувача, а також розроблено таблиці бази даних для збереження інформації, що у подальшому було використано при розробці програми. По завершенню розробки було проведено ряд тестувань для перевірки відповідності вимог та коректності роботи, за результатами якого можна стверджувати, програмне забезпечення готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до усіх поставлених вимог та завдань.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі було доведено актуальність теми роботи, встановлено мету та основні завдання. У першому розділі проведено розгляд та аналіз предметної області, розглянуто існуючі на ринку рішення з наведенням їх переваг та недоліків, на основі отриманих з досліджень даних здійснено постановку задач та вимог. У другому розділі було проведено вибір архітектурних рішень, а також проведено аналіз та порівняння різних технологій для їх реалізації, на основі чого було обрано такі архітектурні шаблони як MVC та Servant, які реалізуються з використанням мови програмування Java та всіх інструментів які працюють з нею. Також було розроблено прототип інтерфейсу користувача та здійснено розробку структури проекту з розробкою відповідних діаграм. У третьому розділі було виконано реалізацію модулів програмного забезпечення з наведенням коду певного функціоналу, розроблено інтерфейс користувача на основі створених раніше прототипів, та розроблено таблиці бази даних. Також було проведено ряд тестувань для перевірки на коректність роботи та відповідність вимогам.

4. Позитивні сторони роботи Актуальність теми даної кваліфікаційної роботи було описано в достатній мірі. При виборі засобів розробки було виконано опис технологій в необхідній мірі з наведенням їх переваг та недоліків. Для розробки бази даних було використано нестандартні інструменти, які в чомусь є кращими за стандартні рішення, а в чомусь гірші за них. Інтерфейс користувача в розробленому програмному застосунку є інтуїтивно зрозумілим для його цільової аудиторії.

5. Негативні сторони роботи Реалізований програмний застосунок не до кінця відповідає темі кваліфікаційної роботи, так як наявність інтерактивних ігор в ньому досить мала, як і самих інтерактивних елементів, тому його можна покращити шляхом додавання таких елементів. Також необхідно вдосконалити способи мотивації того щоб користувачі прагнули покращувати свої знання, наприклад використовуючи різні досягнення, оскільки наявного в роботі методу з використанням таблиці лідерів може бути недостатньо

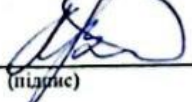
6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення пояснювальної записки виконано у вигляді різноманітних рисунків та діаграм, що відповідають темі кваліфікаційної роботи. Структура та наповнення пояснювальної записки виконана відповідно до вимог та чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує на позитивну оцінку. Матеріал пояснювальної сторінки структурований, послідовний та детально описаний, що дозволяє зрозуміти увесь викладений матеріал відповідної тематики. Наведеного графічного матеріалу достатньо для того щоб наочно вобразити усі аспекти виконаної роботи.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі проте не до кінця відповідає своїй темі через відсутність деяких елементів в ній, або ж їх не повне представлення, тому вона заслуговує на оцінку «добре»

РЕЦЕНЗЕНТ Мартинюк Валерій Володимирович, доктор технічних наук, професор, зав. кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки ХНУ

" 6 " 06 2024 р. 
(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Програмний застосунок для навчання математиці учнів початкових класів з використанням інтерактивних ігор»

Автор: Ялик Вадим Ігорович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Праворська Наталія Іванівна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноновживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) запозичення, виявлені в тексті роботи, є фрагментарними.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 3.0%. Обсяг запозичень, визначений системою Unicheck виявлення збігів ідентичності/схожості, складає 7.05% і адресується до 742 джерел з Інтернету і 213 джерела з бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 5.06.2024

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Наталія ПРАВОРСЬКА