

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки

Кафедра Інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка інформаційної системи інтерактивних засобів навчання»

Виконала студентка 2 курсу групи МІС-19

спеціальності 122 Комп'ютерні науки

Федорова Антоніна Юріївна

Керівник к.ф.-м.н., доцент

Ткач Тетяна Борисівна

Рецензент к.т.н., доцент

Перелигін Борис Вікторович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки
Кафедра Інформаційних технологій
Рівень вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

“ 26 ” жовтня 2020 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Федоровій Антоніні Юріївні

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка інформаційної системи інтерактивних засобів навчання»

керівник роботи к.ф.-м.н., доцент Ткач Тетяна Борисівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 16 ” жовтня 2020р. №194 «С»

2. Строк подання студентом роботи 7 грудня 2020р.

3. Вихідні дані до роботи алгоритм генерації кросворду

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

1 Аналіз предметної області і постановка завдання

2 Аналіз існуючих засобів розробки

3 Моделювання інформаційної системи

4 Програмна реалізація інформаційної системи інтерактивних методів навчання

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 26 жовтня 2020р.

КАЛЕНДАРНИЙ ПЛАН

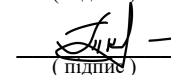
№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1.	Аналіз предметної області і постановка завдання	26.10	100	відмінно
2.	Аналіз інтерактивних методів навчання	29.10	100	відмінно
3.	Практика використання ігор у навчанні	2.11	100	відмінно
4.	Аналіз аналогів	7.11	100	відмінно
5.	Аналіз засобів розробки	9.11	100	відмінно
6.	Рубіжна атестація	19.11	100	відмінно
7.	Модельовання інформаційної системи	20.11	100	відмінно
8.	Створення діаграм	22.11	100	відмінно
9.	Створення бази даних	25.11	100	відмінно
10.	Розробка алгоритму генерації кросворду	28.11	100	відмінно
11.	Програмна реалізація інформаційної системи інтерактивних методів навчання	30.11	100	відмінно
12.	Розробка застосунку перевірки знань	1.12	100	відмінно
13.	Розробки програми генерації кросворду	5.12	100	відмінно
14.	Подання роботи на кафедрі	07.12		
15.	Перевірка на плагіат	08.12		
16.	Рецензування	16.12		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)		100	відмінно

Студентка


(підпис)

Федорова А.Ю.
(прізвище та ініціали)

Керівник роботи


(підпис)

Ткач Т.Б.
(прізвище та ініціали)

АНОТАЦІЯ

на магістерську кваліфікаційну роботу
«Розробка інформаційної системи інтерактивних засобів навчання»
студентки Федорової Антоніни Юріївни

Сучасну освітню систему характеризують: стислі терміни навчання, великий обсяг одержуваної інформації, серйозні вимоги до рівня знань, навичок і умінь учня або студента. Одна з головних задач для нинішнього викладача – зробити процес навчання цікавим, динамічним і сучасним. І в цьому допомагають інтерактивні технології.

Метою магістерської роботи є розробка програмних застосунків, що допомагають в освоєні матеріалу.

Об'єктом дослідження є процес інтерактивного навчання.

Предмет дослідження – інформаційна система для навчання.

Задачі дослідження: провести аналіз предметної області, проаналізувати сучасні засоби розробки систем, провести моделювання системи, реалізувати систему за допомогою вибраних засобів розробки.

Структура магістерської роботи складається з вступу, чотирьох розділів, висновків, переліку посилань. Повний обсяг роботи становить 80 сторінок, містить 28 рисунків, 6 таблиць, 3 формули, 16 джерел.

Ключові слова: інтерактивні засоби, інтерактивне навчання, генерація кросворду, вікторина.

SUMMARY

for a master's degree

«Development of an Information System for Interactive Learning Tools»

students of Fedorova Antonina

Every day, the system is characterized by the following: terms of knowledge, a great deal of acquired information, a great deal of knowledge to the level of knowledge, a tool and a student's wisdom. One of the main tasks for a small business is to make the process of creating tsikavim, dynamic and modern. In addition, there are interactive technologies.

By means of master's robots € the distribution of software supplements, which will help you to master the material.

The object of research – the process of interactive learning.

The subject of research is the informational system for learning.

Tasks of the advancement: to analyze the subject area, analyze the current development of systems, conduct a model of the system, implement the system behind the additional vibration of the distribution.

The structure of the master's robot is built up from the entry, the four sections, the conclusions, the change to the power. A new exchange of robots should be 80 pages, 28 sources, 6 tables, 3 formulas, 16 sources.

Key words: interactive, interactive learning, crossword generation, quiz.

ЗМІСТ

Вступ.....	4
1. Аналіз предметної області і постановка завдання.....	6
1.1 Аналіз інтерактивних методів, форм і засобів навчання.....	6
1.2 Практика використання ігор у навчанні.....	11
1.3 Кросворд у навчальному процесі	17
1.4 Аналіз існуючих інтерактивних засобів навчання	19
1.5 Постановка завдання	24
2. Аналіз існуючих засобів розробки	26
2.1 Операційна система Android.....	26
2.2 Характеристика програмних засобів	32
2.2.1 Інтегроване середовище Android Studio	32
2.2.2 Інтегроване середовище Microsoft Visual Studio	37
2.2.3 Інтегроване середовище Eclipse Java IDE	38
2.2.4 Порівняльний аналіз представлених IDE	40
3. Моделювання інформаційної системи	43
3.1 Проектування інформаційної системи додатку за допомогою діаграм опису бізнес – процесів IDEF0	43
3.2 Розроблення діаграми варіанту використання.....	49
3.3 Проектування бази даних мобільного додатку	52
3.3.1 Опис таблиць бази даних і зв'язків між ними.....	52
3.4 Алгоритм генерації кросворду	56
3.4.1 Метод повного перебору	56
3.4.2 Оптимізація алгоритму повного перебору	58
3.4.3 Оптимізація алгоритму генерації кросворду	59
4. Програмна реалізація інформаційної системи інтерактивних методів навчання	62
4.1 Реалізація мобільного застосунку «Перевір себе»	62
4.2 Реалізація проекту «Кросворд»	67

	7
4.2.1 Опис структури проекту.....	67
4.2.2 Опис роботи програми	68
Висновки.....	71
Перелік джерел посилання	73

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

ВНЗ – вищий навчальний заклад.

ОС – операційна система.

ПЗ – програмне забезпечення.

ADT – Android Development Tools.

ALM – Application Lifecycle Management.

API – Application Programming Interfaces.

APK – Android Package.

ARM – Advanced RISC Machine.

ASP – Active Server Pages.

CPU – Central Processing Unit.

DSL – Domain-Specific Language.

IDE – Integrated Development Environment.

JDT – Java Development Tools.

MIPS – Microprocessor without Interlocked Pipeline Stages.

MVC – Model-View Controller.

OHA – Open Handset Alliance.

PDE – Plug-in Development Environment.

RCP – Rich Client Platform.

RISC – Reduced Instruction Set Computing.

SCM – Supply Chain Management.

SDK – Software Development Kit.

SE – Standard Edition.

SOA – Service-Oriented Architecture.

SSDT – SQL Server Data Tools.

SQL – Structured Query Language.

TFVC – Team Foundation Version Control.

XML – Extensible Markup Language.

Wi-Fi – Wireless Fidelity.

ВСТУП

Перший мобільний пристрій з'явився у 1973 році, і швидко завоював популярність у всьому світі. Він був засобом для телефонування та розмов. В наш час мобільні пристрої – невід'ємна частина життя сучасної людини. Вони є багатофункціональними: дозволяють комунікувати між людьми, виконують функції фото- та відеоапарату, дають можливість прослуховувати музику, використовувати Інтернет – ресурси, грати в ігри та багато іншого.

У наш час найрозповсюдженою та найдоступною ОС є Android. Головними плюсами ОС Android – є те, що вона підтримує більшість смартфонів, та дає безкоштовні засоби розробки (у той час, як ОС IOS потребує чималих затрат).

Друга частина розробки дипломної роботи – створення кросворду. Кросворд з'явився на початку ХХ століття.

Слово кросворд утворилося з двох англійських слів cross – перетин і word – слово. Рішення кросвордів допомагає тренувати пам'ять, розширює кругозір, а також деякі типи кросвордів сприяють розвитку кмітливості та асоціативного мислення.

Кросворд використовують у різних сферах. Наприклад, викладачі теж застосовують кросворд в своїй роботі, це дозволяє учням засвоювати знання в ігровій формі. Сфера застосування кросвордів постійно розширюється, кросворди застосовуються для тестування професійних якостей.

Складання кросворду – заняття досить трудомістке і кропітке, вимагає уважності й посидючості. Для складання кросвордів необхідно мати доступ до словників із списками слів і тлумачень.

Темою мого дипломного проекту є «Розробка інформаційної системи інтерактивних засобів навчання».

Перше розроблене ПЗ – це інформаційна система на базі Android, гра – вікторина «Перевір себе», яка орієнтована на перевірку засвоєного матеріалу

після лекцій. Вікторина – це гра, яка направлена на когнітивний розвиток людини, перевіряє логіку, пам'ять та її мислення.

Друге розроблюване ПЗ – це ПЗ для операційної системи Windows. Має забезпечити можливість автоматично генерувати кросворди згідно користувальницького шаблону, заповнюючи його словами зі словнику.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз інтерактивних методів, форм і засобів навчання

Інтерактивні методи навчання на сьогоднішній день є актуальним способом роботи викладача в аудиторії, тренера в групі, педагога в будь-якому освітньому закладі. На відміну від традиційних, інтерактивні методи навчання базуються на тому, що учасники будуть активно взаємодіяти на навчальному процесі та між собою.

Залежно від характеру взаємодії вчителя та учнів розрізняють пасивні, активні та інтерактивні методи навчання.

Пасивні методи – це навчання, у процесі якого вчитель – дійова особа, яка керує тим, як йде заняття (рис. 1.1). Учні – виконують роль пасивних слухачів, та підпорядковуються указам вчителя [1]¹⁾.

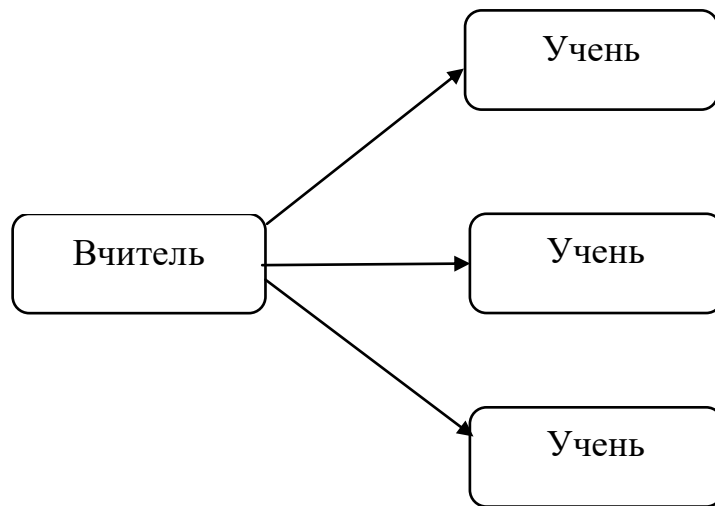


Рисунок 1.1 – Схема пасивного методу навчання

Зв'язок педагога з учнями здійснюється за допомогою опитувань, самостійних, контрольних робіт, тестів.

¹⁾ [1] DSpace JSPUI (загол. з екрану). URL: <http://repo.knmu.edu.ua/handle/123456789/8425>. (дата звернення – 01.12.2020).

З огляду на ефективність засвоєння навчального матеріалу, пасивні методи – мало ефективні. Але вони мають деякі плюси: вчителю відносно легко підготуватись до заняття, є можливість викласти великий обсяг навчального матеріалу за обмежений час, та можливість працювати з великою аудиторією.

Активні методи – це навчання, у процесі якого учні та вчитель взаємодіють один з одним (рис. 1.2). Тут учні – не пасивні, а активні учасники. У пасивних методах – авторитарний стиль. Активні методи роблять акцент на демократичний.

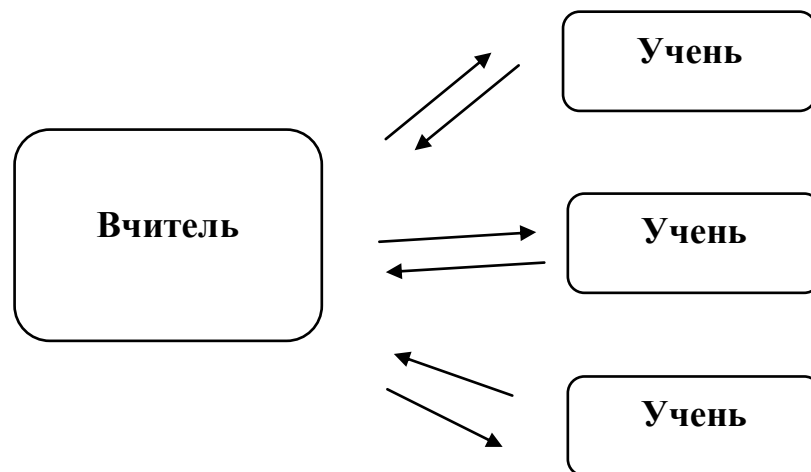


Рисунок 1.2 – Схема активного методу навчання

Інтерактивні методи – це форма навчання, у процесі якої учні і вчителя взаємодіють один з одним, ведуть діалог між собою (рис. 1.3). Це – взаємонавчання: вчитель – учень, учень – учень. При цьому, вчитель і учень – рівноправні. В цьому методі виключається домінування одного учасника над іншим.

Інтерактивні методи, на відміну від активних, орієнтовані на взаємодію учнів не тільки з вчителем, а й один з одним, на активність учнів у процесі

навчання. Слово «інтерактивний» прийшло з англійської мови і виникло від слова «interact». «Inter» – це «взаємний», а «act» – «діяти».

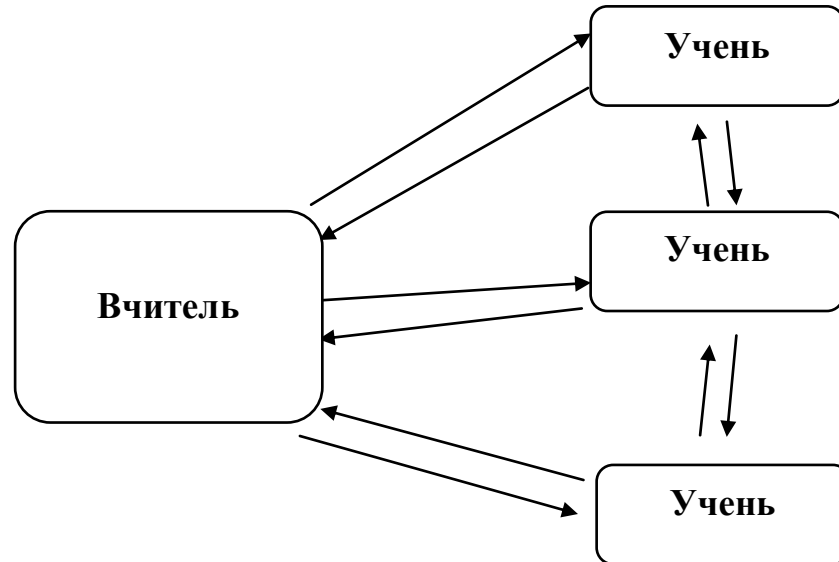


Рисунок 1.3 – Схема інтерактивного методу навчання

За допомогою інтерактивних технологій учні вчаться:

- робити аналіз навчальної інформації і творчо підходити до засвоєння матеріалу;
- формулювати власну думку, висловлювати її, доводити свою позицію та аргументувати;
- поважати думку інших;
- будувати конструктивні стосунки у групі, уникати конфліктів та розв'язувати їх, шукати компроміси;
- самостійно працювати, виконувати творчу роботу.

У 1980-х роках Національним тренінговим центром(США, штат Меріленд) були проведені дослідження, які показують, що завдяки інтерактивному навчанню, різко збільшується відсоток засвоєння матеріалу, так як воно впливає не лише на свідомість учнів, а й на його почуття.

Результати цих досліджень відображено у схемі, яка дістала назву «Піраміда навчання» (рис.1.4).



Рисунок 1.4 – Піраміда навчання

Згідно з пірамідою, видно, що найменші результати досягаються за умов пасивного навчання: лекція – 5%, читання – 10%, зорове і слухове сприймання – 20%, демонстрування – 30%.[2]¹⁾. А найбільші результати – за умов інтерактивного навчання: групова дискусія – 50%, практичні вправи – 75%, навчання інших (практичне) – 90%. Ці дані – середньостатистичні, тому можуть різнитися.

Виділяють такі принципи інтерактивного навчання:

- принцип активності (кожен учень повинен активно брати участь у процесі обговорення та активно взаємодіяти з іншими);

¹⁾[2] 3.3 Огляд інтерактивних методів (загол. з екрану). URL: lt.multycourse.com.ua/ua/page/19/67#2. (дата звернення – 01.12.2020).

- принцип відкритого зворотного зв'язку (учасники групи висловлюють свої думки, ідеї або заперечення), саме завдяки цьому вони пізнають особливості поведінки, стиль мислення інших;
- принцип експериментування (учні активно шукають нові ідеї та шляхи вирішення поставлених завдань);
- принцип довіри у спілкуванні (зазвичай – прийом розташування учня і вчителя по колу обличчям одне до одного);
- принцип рівності позицій (вчитель не нав'язує учням своєї думки, а навпаки, діє разом нарівні з ними).

Види інтерактивних методів навчання: [3]¹⁾

- метод мозгової атаки (мозговий штурм) – учні спонтанно висувають свої ідеї на задану тему. На першому етапі йде запис ідей, далі – їх класифікують і використовують для подальшого аналізу;
- «круглий стіл» – слухачі пробують поставити питання по темі, аргументують методи їх вирішення;
- дискусія – активний метод проведення занять, який мобілізує практичні і теоретичні знання учасників, а також погляди на проблему, яка розглядується. Зазвичай в складі вивчаємої теми шукають питання, на які слухачі мають різні точки зору;
- дебати – організований процес формулювання і захисту своїх позицій щодо конкретної проблеми двома чи більше учасниками;
- ситуаційний аналіз – слухачі після ознайомлення з проблемою, самостійно аналізують ситуацію та надають ідеї рішення іншим слухачам.

Виділяють:

- ситуації – ілюстрації (містить приклад з управлінської практики, і спосіб рішення ситуації), ситуації – оцінки (опис ситуації і її можливе рі-

¹⁾[3]Інтерактивні методи навчання (загол. з екрану).

URL: <https://sites.google.com/site/nmcmky/naukova-dialnist/interaktivni-metodi-navcanna>. (дата звернення - 01.12.2020)

- шення в готовому виді, потрібна оцінка ефективності), ситуації – вправи (підготовлений епізод конкретної вправи підготовлений так, що його рішення вимагає стандартних дій);
- аналіз конкретних ситуацій – аналіз конкретних ситуацій, що включає глибоке і детальне дослідження реальної або імітованої ситуації.

1.2 Практика використання ігор у навчанні

Психологи вважають, що гра є одним із основних видів діяльності людини, та охоплює всі періоди її життя. Ігрова форма навчання – це спосіб взаємодії вчителя і учнів, зумовлений грою, що веде до визначення та реалізації дидактичних цілей та завдань навчання.

Зазвичай наближення дій студента в навчальній аудиторії до реальної професійної діяльності має характер певної умовності, що є характерною ознакою ігрових дій. Таке моделювання, спрямоване на орієнтацію професійної поведінки, в умовах навчання, відбувається завдяки педагогу, який розробляє комплекс різноманітних ігрових ситуацій. Роль викладача в цьому – запропонувати студенту навчальні ситуації, які мають професійне спрямування для студента.

Саме в таких симуляційно-ігрових ситуаціях студенти можуть асоціювати себе з роллю тієї особи, яку вони виконують. Також можуть поставити себе на місце цієї особи в певних обставинах, які допоможуть збагатити їхній життєвий досвід. Це стає можливим саме при моделюванні віртуально-професійних ситуацій у фаховій підготовці. Деякі дослідники (У. Ріверс) називають таку діяльність імітаційно-моделюючою грою. Практично відмінності між рольовими іграми та симуляціями є здебільшого умовними, оскільки мета як симуляцій, так і рольових ігор полягає у формуванні професійних умінь студентів під час вирішення різноманітних професійних проблем. Та-

ким чином, реалізується рольова поведінка майбутніх соціальних працівників у заданих чи створених викладачем умовах.

В рольових іграх погляди або переконання учасників можуть не збігатися з поглядами їх персонажів, тоді як симуляції надають змогу учасникам бути самим собою.

Саме з цієї причини багато дослідників розмежовують поняття рольових і ділових ігор та симуляцій. У рольовій грі погляди чи переконання учасників можуть не збігатися з поглядами їх персонажів, тоді як симуляції надають змогу учасникам бути самим собою. Порівняно з реальною професійною діяльністю, коли прийняте рішення або вчинок призводять до певних результатів в ігровій взаємодії можна змінити перебіг і спосіб діяльності, пробувати різні варіанти професійної поведінки, скласти прогнози стосовно її вияву в майбутньому. Це надає можливість майбутнім фахівцям обрати найбільш оптимальний та ефективний шлях формування власного професіоналізму і професійної компетентності.

Гра є універсальною формою, в якій відбуваються потужні процеси самовизначення, самовиявлення, самоствердження та самоперевірки. Ігри розвивають кмітливість та логіку.

Тому серед найбільш уживаних засобів інтерактивних технологій, які використовуються у вищій школі, є ігрові технології навчання. Оскільки існують різні види ігор, що застосовуються у навчальному процесі (ділові, рольові, дидактичні, імітаційні тощо), і кожний із цих видів має свою технологію реалізації й обумовлюється процесом підготовки до її впровадження (ігрове проектування), то можна говорити про різні ігрові технології.

Важлива особливість інтерактивних ігор – їх здатність мотивувати навчання студентів, сприяти їх соціалізації та професійному розвитку. Тому навчальний вплив інтерактивних ігор надає можливість студенту самому задавати стандарти, якими він вимірює ступінь задоволення власною поведінкою.

Зараз пропонується багато варіантів ігрових вправ для навчання школярів та студентів, однак механічне застосування інтерактивних ігор не гарантує успішності навчального процесу.

Переваги інтерактивних ігор:

- мотивують до навчання (пробуджують допитливість студентів до способів вирішення професійних та навчальних проблем в ігровому середовищі);
- сприяють розвитку особистості (створюють тривалу зацікавленість у саморозвитку та розкритті свого потенціалу);
- полегшують введення нових комунікативних і поведінкових норм;
- спонукають краще зрозуміти і навчитись використовувати всю складність психічних, соціальних та організаційних процесів спілкування між людьми;
- гра надає студентам можливість вирішувати складні проблеми, не залишаючись пасивними спостерігачами;
- знижують тривожність учасників, особливо на початковій стадії роботи.

Усі ігрові технології для підготовки фахівців у вищій школі можна об'єднати загальним поняттям «дидактично-ділові ігри». Вони є одночасно формою організації освітнього середовища, засобом підготовки майбутніх фахівців, методом їхнього професійного навчання, що реалізується викладачем за спеціально розробленим ігровим сценарієм (рис.1.5) [4]¹⁾.

Застосування дидактично – ділової гри як форми контекстного навчання покликано вирішувати такі педагогічні завдання:

- стимулювання пізнавальної мотивації студентів для забезпечення умов вияву професійної мотивації;

¹⁾[4] І.М.Мельничук. Особливості застосування інтерактивних ігор у вищому навчальному закладі. УДК: 37.02:372. 12с.

- розвиток теоретичного та професійно – практичного мислення у фаховій сфері майбутніх соціальних працівників;
- формування у студентів цілісного уявлення про майбутню професійну діяльність, її динаміку.

За ігровою методикою можна розподілити дидактичні ігри на дві великі групи: імітаційні (аналіз конкретних ситуацій, логічно – мислительні, креативно-пізнавальні, куди можна віднести «Мозковий штурм», «Брейн-ринг», «Зіпсований телефон» тощо) та рольові, що мають певний логічно – змістовий сюжет, професійно – ділову чи соціально – психологічну спрямованість, драматичність.

Тому об'єднують організаційно-діяльнісні, професійно – комунікативні, ситуативно-сюжетні, драматичні ігри, які виокремлюються як самостійні в інших класифікаціях, і симуляційно – ігрові ситуації.

У процесі реалізації інтерактивних ігрових технологій обох категорій можуть використовуватися предметні засоби, як основний та допоміжний матеріал (технічні засоби навчання, бланки, таблиці, роздатковий матеріал, маркери, дошка, кольорова крейда); методичні засоби інтерактивних технологій (методичні розробки і сценарії ігор та ігрових вправ) й ігрове проектування.

Залежно від мети, завдань та етапу застосування імітаційних чи рольових ігор вони характеризуються як навчальні (репродуктивні, пізнавально – розвиваючі, творчі, узагальнюючі, контролюючі) і тренувальні (у сенсі відпрацювання та формування необхідних умінь і навичок діяльності), що включають у себе діагностичні, комунікативні, соціально-психологічні, рефлексивні ігри.



Рисунок 1.5 – Класифікація інтерактивних ігрових технологій для вищої школи

Методика застосування інтерактивних ігрових технологій передбачає реалізацію таких основних етапів:

1) Підготовка до проведення заняття в ігровій формі:

- містить визначення мети гри та її часового обмеження;
- установлення відповідності її змістового наповнення навчальній дисципліні і темі заняття;
- розробку власного чи адаптація готового сценарію гри;
- похвилинне структурування етапів ігрової взаємодії (мотиваційної, діяльнісної, рефлексивної, контрольної - оцінювальної);
- забезпечення необхідним обладнанням та матеріалами;
- розробку чіткої покрокової інструкції для студентів – учасників ігрової взаємодії (мета, завдання, правила, ігрові дії, умови).

2) Безпосереднє проведення гри, яке передбачає діяльність студентів як виявлення їхньої внутрішньої (психологічної), зовнішньої (фізичної) та соціальної (професійно спрямованої) активності.

Діяльнісна активність регулюється усвідомлюваною метою і має такі ознаки, як:

- передбачення результату;
- усвідомлення можливості його досягнення;
- обґрунтоване планування своїх дій;
- вибір відповідних раціональних форм, методів, засобів та орієнтирів у міжособистісних відносинах;
- гармонізація внутрішньої і зовнішньої діяльності;
- оцінювання процесу та результатів своєї праці, що потребує вмінь нормувати, урахувати, контролювати, приймати оптимальні рішення, вирішувати загальні організаційні завдання (діагностувати, прогнозувати, стимулювати діяльність, підходити до неї комплексно, системно, цілісно) і надати змогу студентам перевірити доцільність діяльності та її співрозмірність поставленій меті.

3) Аналітично-оцінювальний етап, що передбачає рефлексію, узагальнення результатів гри, формулювання висновків, визначення студентами по-

зитивних аспектів результативності гри для їхнього особистісного професійного зростання в майбутньому, і встановлення викладачами недоліків, упущень, шляхів удосконалення ігрової взаємодії студентів для подальшого використання апробованої методики в майбутньому.

Ігрові технології можна розглядати як:

- інструмент для трансляції і засвоєння попереднього до професійної діяльності досвіду;
- аналіз моделей реальності (на зразках професійних дій представниками різних рольових та особистісних позицій);
- адаптацію до майбутньої професійної діяльності.

Застосування дидактичних ігор сприяє перетворенню студента з об'єкта навчання в суб'єкта професійної спрямованої праці, що викликає його цілеспрямовану діяльність та творчу участь у самостійному формуванні професійної компетентності.

Отже, аналіз сутності інтерактивних ігрових технологій надає змогу визначити основні особливості їх застосування у навчальному процесі ВНЗ.

1.3 Кросворд в навчальному процесі

Найбільш ефективним і зручним у використанні для формування термінологічної грамотності учнів є кросворд.

Кросворд – це ігрова методика, сутність якої полягає в розгадуванні слів за наведеними визначеннями.

Кожен навчальний предмет висуває свою класифікацію кросвордів, і це впливає з своєрідності даної навчальної дисципліни. Тому простіше за основу взяти головну навчальну мету заняття.

Кросворд допомагає:

- організувати самостійну роботу;
- викликати інтерес до теми, що вивчається;

– формуванню термінологічної грамотності, тобто більш міцному засвоєнню термінологічний апарату навчальної дисципліни [5]¹⁾.

При складанні кросвордів необхідно дотримуватися принципів наочності і доступності.

Правила складання кросворду:

- не допускається наявність незаповнених клітин в сітці кросворду;
- не допускаються випадкові буквосполучення і перетину;
- загадані слова повинні бути іменниками в називному відмінку однини;
- дволітерні слова повинні мати два перетину;
- трьохлітерні слова повинні мати не менше двох перетинів;
- не допускаються аббревіатури (ЗіЛ і інші), скорочення (дитбудинки і інші);
- не рекомендується велика кількість двобуквених слів.

Переваги кросворду як методу навчання:

- дозволяє деталізувати конкретні розділи та складні теми навчальної дисципліни;
- у кросворді, відгадували одиниці (терміни), повинні бути однозначними, лаконічними і конкретними. За рахунок цього і забезпечується швидке запам'ятовування терміну і його значення;
- активізується увагу, пам'ять, логічне мислення, мова.

Форми роботи з кросвордом:

- організація самостійної позаурочної діяльності учнів. Це сприяє розвитку вміння самостійно і швидко орієнтуватися в навчальному матеріалі. правильно і точно формулювати питання, визначати тип кросворда і необхідність його застосування саме до цієї теми, ство-

¹⁾[5] Учебный кроссворд: как и где составить (загол. з екрану). URL : <https://www.eduneo.ru/uchebnyj-krossvord-kak-i-gde-sostavit/>. (дата звернення – 01.12.2020).

ривати кросворд в друкованому та електронному вигляді, а так само сприяє розвитку творчих здібностей особистості;

- організація роботи на занятті. На етапі повторення матеріалу: п'ятихвилинні перевірочні роботи. На етапі перевірки знань: робота з термінами і поняттями навчальної дисципліни по конкретній темі. На етапі вивчення нового матеріалу: передбачення майбутнього результату.

Етапи складання навчальних кросвордів:

- вибираємо вид кросворда: краще використовувати асиметричні, нестандартні, з вільним розташуванням слів;
- складаємо список термінів (в залежності від типу заняття і його цілей).

Критерії оцінювання результату. Безсумнівно, критерії залежать від багатьох факторів і повинні визначатися викладачем індивідуально, можна лише виділити окремі базові пункти:

- частка відгадати слів;
- частка ключових термінів;
- точність і однозначність формулювань питань.

1.4 Аналіз існуючих інтерактивних засобів навчання

В процесі виконання дипломної роботи були розглянуті ігри схожого типу на площадці Google Play, а також на просторах Інтернету. З великого списку додатків можна виділити декілька успішних.

Перша інформаційна система – «Online Test Pad» (рис. 1.6):

- головна сторінка: є вся необхідна інформація про вибір теми кросворду або його створення;
- є можливість створювати кросворди, опитування або тести;
- сервіс є абсолютно безкоштовним;

– є можливість пройти кросворди, тести, опитування онлайн [6]¹⁾.

Короткий опис інформаційної системи. Інформаційна система дуже обширна, має багато функцій. Зручний інтерфейс, вміст сайту дуже наповнений. Є можливість реєстрації, входу. Можна вести бесіду. На сайті можна як створювати свої кросворди, опитування або тести, так і використовувати вже готові. Є можливість коментувати та ставити зірочки (рейтинг). На головній сторінці є пошук по сайту, що облегшує роботу з ним. Також є посилання на соціальні мережі.

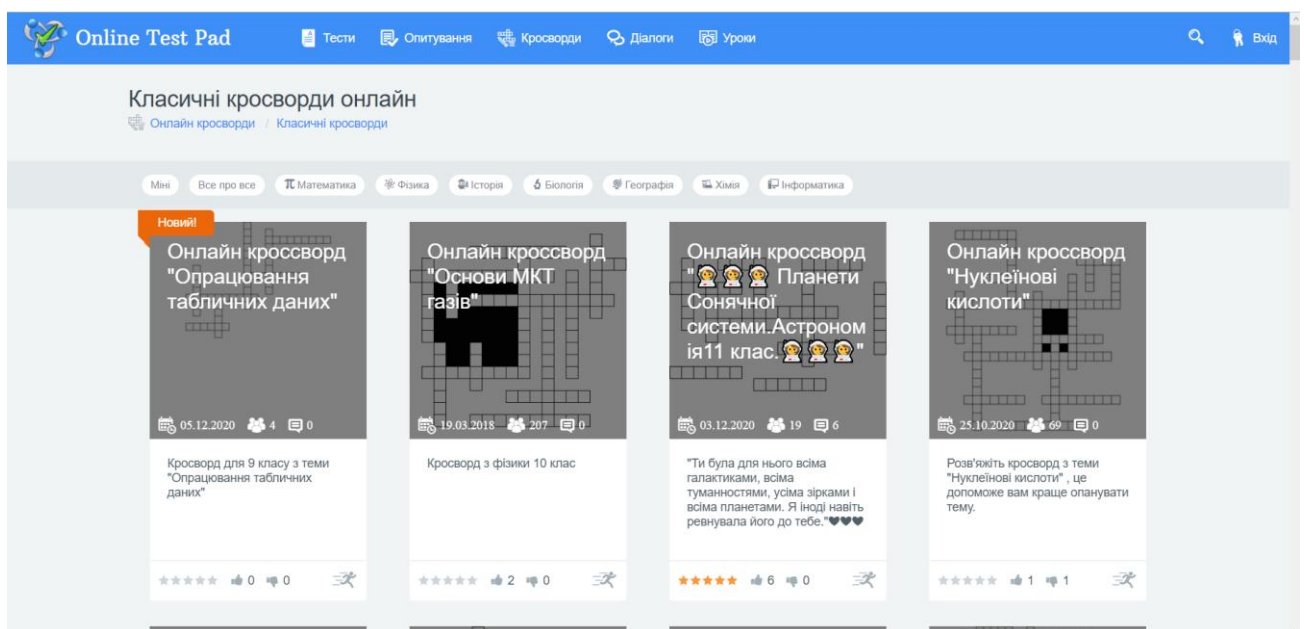


Рисунок 1.6 – Інформаційна система «Online Test Pad»

Друга інформаційна система «Джерело» (рис. 1.7):

- головна: тут можливо знайти головну сторінку з описом сайту, послуги, контакти;
- окремі розділи для вчителів та учнів;
- окрема сторінка з відповідями на запитання;

¹⁾[6] Класичні кросворди онлайн. (загол. з екрану). URL : <https://onlinetestpad.com/ua/crosswords/classic>. (дата звернення – 03.12.2020).

– можливість реєстрації [7]¹⁾.

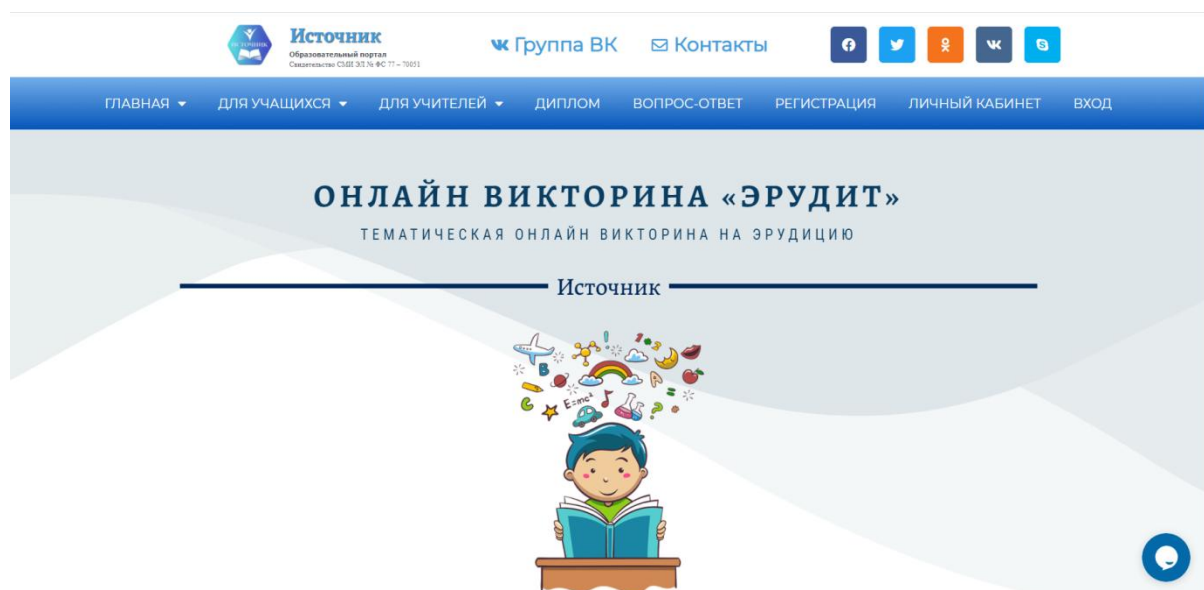


Рисунок 1.7 – Інформаційна система «Джерело»

Короткий опис сайту: дизайн сайту має досить приємну легкість. Головна сторінка сайту включає в себе всю необхідну інформацію. На сайті «Джерело» в першу чергу є багато інформації. На головній сторінці немає нічого зайвого. Перелік пропонованих програм розташований дуже компактно, і за рахунок цього можна побачити якомога більше інформації за раз. Це дуже зручно і економно за часом. Після реєстрації з'являється доступ до особистого кабінету учасника, це дає змогу отримувати результати тестів на свою поштову скриньку та можливість оформити диплом.

Розглянемо та проаналізуємо мобільний додаток «Всесвітня історія вікторина» (рис. 1.8).

Гра у жанрі вікторини. Призначена для всіх, хто хоче перевірити свої знання з шкільного курсу всесвітньої історії.

¹⁾[7] Онлайн вікторина «Эрудит». (загол. з екрану). URL : <https://source2016.ru/event/>. (дата звернення – 03.12.2020).

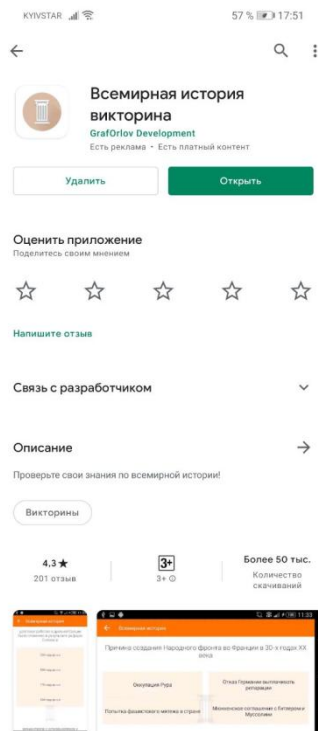


Рисунок 1.8 – Мобільний додаток «Всесвітня історія вікторина»

Має вибір на кількість питань. Також можна вибирати кількість гравців (змагання), та гра на виживання або на час. Гра має більше 50 000 завантажень.

Має 8 режимів:

- 10 питань;
- 20 питань;
- 30 питань;
- 1 хвилина;
- 3 хвилини;
- 5 хвилин;
- на виживання;
- на виживання (3 спроби)[8]¹⁾.

¹⁾[8] Всемирная история викторина (загол. з екрану). URL : <https://play.google.com/store/apps/details?id=com.kalinovskiy.historyquiz>. (дата звернення – 03.12.2020).

Розглянемо наступний мобільний додаток – «Кросворди на російській» (рис. 1.9).

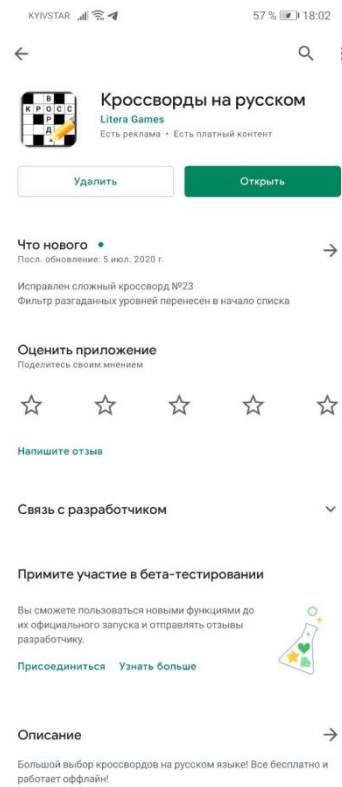


Рисунок 1.9 – Мобільний додаток «Кросворди на російській»

Гра має більше 1 000 000 завантажень. Короткий опис додатку: більше трьохсот кросвордів. Додаток може працювати в офлайн – режимі.

Категорії на вибір:

- класичні;
- тематичні;
- всі слова на задану літеру;
- фігурні;
- складні.

Ключові особливості:

- декілька різних категорій кросвордів;

- всі кросворди можуть працювати офлайн;
- легкі і складні кросворди по складності;
- підказки для відкриття літер і слів [9]¹⁾.

1.5 Постановка завдання

На основі проведеного дослідження подібних інформаційних систем були сформульовані вимоги до проектованої системи.

Метою моєї дипломної роботи є розробка мобільного додатку «Вікторина» на базі платформи Android Studio та розробка кросворду за допомогою інтегрованого середовища розробки Visual Studio.

Для досягнення мети були поставлені наступні задачі:

- здійснити постановку задачі;
- зробити огляд програмних засобів розробки кросплатформових мобільних ігор;
- спроектувати додатки;
- реалізувати і протестувати додатки.

¹⁾[9] Кроссворды на русском (загол. з екрану). URL : https://play.google.com/store/apps/details?id=com.appspot.orium_blog.crossword. (дата звернення – 03.12.2020).

2 АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ РОЗРОБКИ

У наш час існує величезна кількість як платних, так і безкоштовних інструментів по розробці ігор та програмних продуктів, починаючи від простих бібліотек для відомих мов програмування, закінчуючи великими редакторами з великим функціоналом.

Для створення якісної гри необхідно використовувати якісні інструменти. Тому я вирішила використовувати готові інструменти для розробки ігор, додатків та програмних продуктів.

Оскільки середовища для розробки додатків стали дуже популярні, кількість їх на ринку прибуває з кожним днем. У даному розділі – розбір найбільш популярних ігрових платформ, які зарекомендували себе на ринку.

2.1 Операційна система Android

Android OS – операційна система для смартфонів, планшетних комп'ютерів, електронних книг, цифрових програвачів, "розумних" наручних годинників, ігрових приставок, нетбуків, смартбуків, окулярів Google, телевізорів, ситем автоматичного керування автомобілем та інших пристроїв. ОС заснована на ядрі Linux і власної реалізації віртуальної машини Java від Google. Спочатку розроблялася компанією Android Inc., яку в 2005 році купила Google. Згодом Google ініціювала створення альянсу Open Handset Alliance (ОНА), який зараз займається підтримкою і подальшим розвитком платформи.

Android дозволяє створювати Java – додатки, що керують пристроєм через розроблені Google бібліотеки. Android Native Development Kit дозволяє перенести (але не налагоджувати) бібліотеки і компоненти додатків, написані на Сі та інших мовах.

Кожній новій версії Android присвоювалася назва, яка відповідає назві будь-якого десерту. Ничже показана статистика використання версій Android на квітень 2020 року (рис. 2.1).

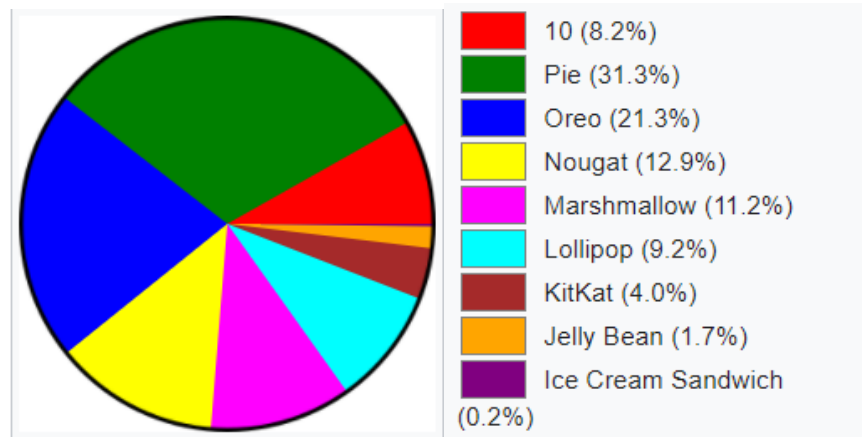


Рисунок 2.1 – Статистика використання версій Android

Найпопулярнішими є Android 9.0 «Pie», Android 8.0 / 8.1 «Oreo», Android 7.0 / 7.1 «Nougat»[10]¹⁾.

Android 9.0 «Pie» вийшла 7 березня 2018 року. Версією ядра є Linux, версія API: 28. Основні характеристики:

- RTT API дозволяє вимірювати відстань між мобільним пристроєм;
- у сповіщеннях відображаються отримані зображення;
- у користувачей є змога блокувати групи каналів;
- оновлений зовнішній вигляд повзунка гучності і дати на екрані блокування;
- в меню живлення додана клавіша для створення скріншоту;
- у Ambient Display доданий індикатор заряду батареї;
- введено єдиний шаблон для реалізації аутентифікації користувача за від-

¹⁾[10] Список версій Android (загол. з екрану). URL : https://uk.wikipedia.org/wiki/Список_версій_Android. (дата звернення – 07.12.2020).

битками пальців для всіх за стосунків.

Android 8.0 / 8.1 «Oreo» вийшла 21 серпня 2017 (8.0), 5 грудня 2017 (8.1).

Версією ядра є Linux: 4.10. Версія API: 26,27.

Основні характеристики Android 8.0:

- режим картинка-в-картинці;
- підтримка кількох дисплеїв;
- перероблено систему сповіщень;
- можливість відкласти сповіщення на певний час, або видалити їх;
- поліпшене автозаповнення полів на кшталт імені користувача та номера кредитної картки за допомогою фреймворку;
- закріплення ярликів і віджетів усередині будь-якого застосунку.

Основні характеристики Android 8.1:

- Android Go – спрощена версія дистрибуції Android, яка призначена для смартфонів низького та ультрабюджетного класу;
- API нейронних мереж;
- різноманітні поліпшення й оновлення безпеки.

Android 7.0 / 7.1 «Nougat» вийшла 22 серпня 2016 року. Версією ядра є Linux: 3.10. Версія API: 24, 25.

Основні характеристики Android 7.0:

- режим багатовіконного поділу екрану, в якому два додатки можуть поділити його навпіл;
- іконки швидкого доступу виводяться на компактній панелі;
- фільтрація вхідних дзвінків за телефонним номером;
- фонове перемикання завдань;
- нічний режим дозволяє досягти оптимального відображення інформації на екрані за допомогою автоматичного підвищення контрастності та регулювання яскравості;

- покращена функція енергозаощадження «Doze». Раніше вона працювала тільки, коли телефон був нерухомий, але у цій версії Google стверджує, що «Doze економить батарею всякий раз, коли екран вимикається»;
- новий дизайн папок: значки всередині рамки збудовані у сітку.

Основні характеристики Android 7.1:

- нічний режим «Night light» (фільтр синього кольору);
- можливість швидкої відповіді на повідомлення свайпом по сканеру відбитків пальців;
- автоматичні (фонові) оновлення безпеки;
- підтримка Daydream VR.

Архітектура Android формується з набору компонентів. Кожен компонент побудований на основі елементів нижчого рівня(рис. 2.2). У нижній частині малюнка можна побачити, що ядро Linux пропонує основні драйвери для апаратних компонентів системи.

Крім того, ядро відповідає за пам'ять, управління процесами, підтримку мережі і т. д. Середовище виконання Android, що є надбудовою над ядром, відповідає за породження і виконання додатків Android. Кожна програма працює у власному процесі зі своєю віртуальною машиною Dalvik. Крім бібліотек ядра, що пропонують деяку функціональність Java SE, існує також набір рідних бібліотек на C / C ++, що створюють основу для фреймворка додатка [11]¹⁾.

Ці системні бібліотеки в більшості своїй відповідають за складні в обчислювальному сенсі завдання (промальовування графіки, відтворення звуку, доступ до бази даних), що не дуже підходящі для віртуальної машини Dalvik.

API в них обгорнуті за допомогою класів Java у фреймворк програми. Фреймворк додатки пов'язують разом системні бібліотеки і середу виконан-

¹⁾[11] Архітектура Android (загол. з екрану). URL : https://studbooks.net/2052989/informatika/arhitektura_android. (дата звернення – 08.12.2020).

ня, створюючи таким чином призначений для користувача бік Android. Фреймворк управляє додатками і пропонує продуману середу, в якій вони працюють.

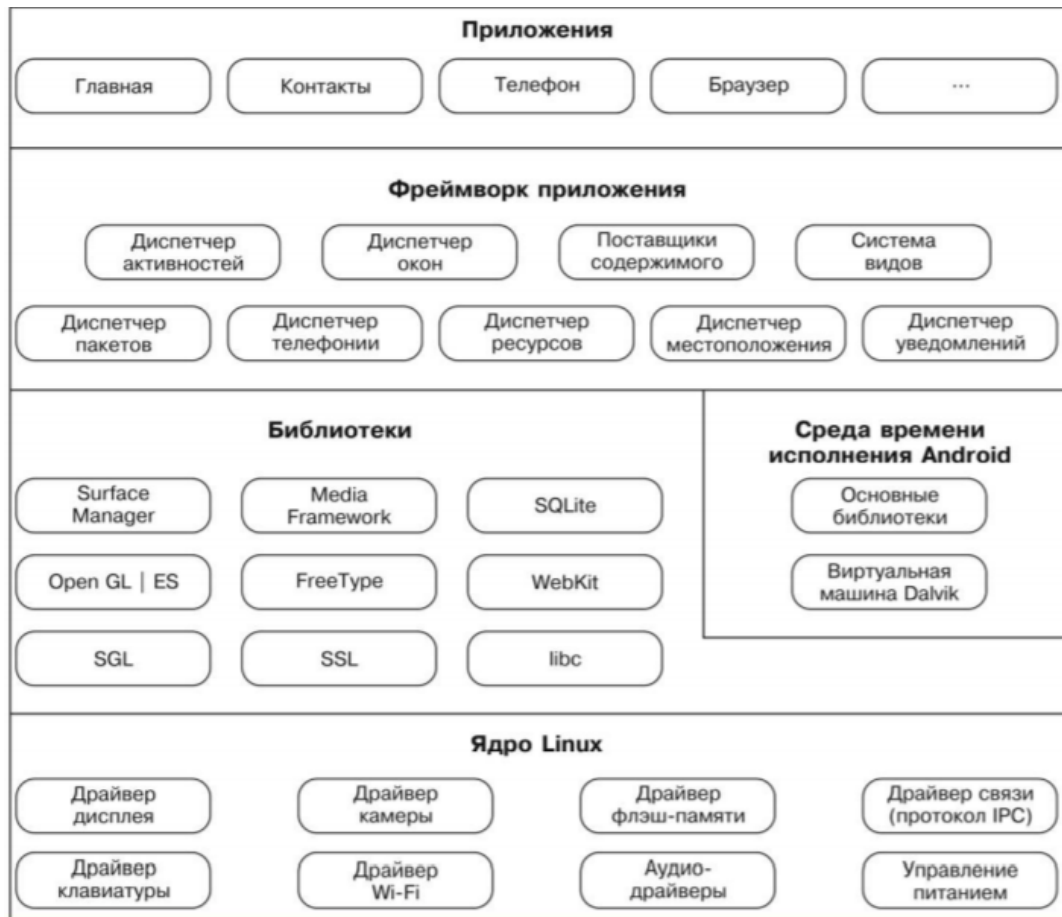


Рисунок 2.2 – Архітектура ОС Android

Розробники створюють додатки для цього фреймворка за допомогою набору програмних інтерфейсів на Java, що охоплюють такі області, як розробка користувальницького інтерфейсу, фонові служби, оповіщення, управління ресурсами, доступ до периферії і т. д. Всі ключові програми, поставляються разом з ОС Android (наприклад, поштовий клієнт), написані за допомогою цих API.

Для розробки ігрових додатків під операційну систему Android можна використовувати стандартний інструмент для розробки додатків Android Studio. Так само існує кілька комерційних і відкритих ігрових движків і фреймворків, такі як Unity, Unreal engine, Torque2D, Cocos 2D-X, libGDX, Xamarin, Corona SDK.

Переваги ОС Android:

- Android, на відміну від iOS, є відкритою платформою, що дозволяє реалізувати на ній більше функцій;
- в Android-апаратах, як правило, присутній microSD-кардрідер, що робить можливим швидке перенесення файлів з комп'ютера на телефон;
- незважаючи на початкову заборону на установку програм з «неперевічених джерел» (наприклад, з карти пам'яті), це обмеження відключається штатними засобами в налаштуваннях апарата, що дозволяє встановлювати програми на телефони та планшети без інтернет-підключення (наприклад, користувачам, які не мають Wi-Fi – точки доступу і не бажають витратити гроші на мобільний інтернет, який зазвичай коштує дорого), а також дозволяє всім бажаючим безкоштовно писати програми для Android і тестувати на своєму апараті;
- Android доступний для різних апаратних платформ, таких як ARM, MIPS, x86;
- існують альтернативні Google Play магазини додатків: Amazon Appstore (англ.), Opera Mobile Store, Yandex.Store.

Недоліки ОС Android:

- наявність в деяких Android – пристроях сервісів Google, що забезпечують можливість передачі ідентифікаційної інформації на сервери компанії, наприклад, інформацію про переміщення користувача в реальному часі.
- у версії Android 1.6 розробники додали Native Development Kit, який дозволяє писати власні низькорівневі модулі для системи на мовах C /

C ++, спираючись на стандартні Linux – бібліотеки. Хоча, наприклад, стандартна бібліотека мови Cі на платформі Android, відома як Bionic, якраз не є стандартною і повністю сумісною з libc.

При роботі над даним мобільним додатком «Вікторина», була використана IDE Android Studio. Далі буде розглянуто саме цей інструмент.

2.2 Характеристика програмних засобів

2.2.1 Інтегроване середовище Android Studio

Android Studio – це інтегроване середовище розробки (IDE) для роботи з платформою Android. Анонс був 16 травня 2013 року на конференції Google I / O (рис. 2.1).

IDE знаходиться у вільному доступі починаючи з версії 0.1, опублікованій в травні 2013, потім перейшла в стадію бета-тестування, починаючи з версії 0.8, яка була випущена в червні 2014 року. Перша стабільна версія 1.0 була випущена в грудні 2014 року, тоді ж припинилася підтримка плагіна Android Development Tools (ADT) для Eclipse.

Android Studio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains (офіційний засіб розробки Android додатків).

Дане середовище розробки доступне для Windows, OS X і Linux. Нові функції з'являються з кожною новою версією Android Studio.

Наразі доступні наступні функції:

- робота за макетами;
- редактор руху об'єктів (MotionLayout). API MotionLayout розширює можливості ConstraintLayout. Допомогає розробникам Android управляти складними рухами і анімацією віджетів. Використання цього API стало простіше завдяки новому Motion Editor - потужному інтерфейсу для створення, редагування і попереднього перегляду. Motion Editor створює і змінює за нас складні XML-файли, при цьому

- він підтримує редагування переходів, кейфреймів, атрибутів перегляду. Редактор автоматично створює код;
- інспектор макетів (Layout Inspector). За допомогою Layout Inspector стає зрозумілішим налагодження UI. Дає змогу отримувати доступ до постійно оновлюваних даних і інформацію про те, як витрачаються ресурси;
 - валідація макетів (Layout Validation). При розробці коду під декілька розмірів екрану, за допомогою вікна Layout Validation можна побачити як виглядають макети на різних конфігураціях і екранах одночасно, і як коректно вони відображаються.
 - розробка і профілювання: CPU Profiler. Надає великий обсяг інформації про потокову активність і трасування програми;
 - функції інтелектуального редактору: для створення файлів для R8 (який представлений в Android Gradle plugin 3.4.0), студія дає функції інтелектуального редактору: підсвічування синтаксису, автодоповнення і перевірка помилок). Редактор інтегрується з проєктом, забезпечує доповнення символів всіх класів, методів та полів;
 - збірка (плагін Gradle 4.0 включає в себе підтримку Android Studio's Build Analyzer (незалежно від мінімального рівня API) і створює залежності між динамічними модулями функцій;
 - Java 8 і десахаризація в D8 / R8 (попередні версії Android Gradle підтримували різні мовні функції Java 8 для всіх рівнів API за допомогою процесу, званого десахаризацією. Тепер можна використовувати стандартні мовні API, які раніше були доступні тільки в останніх версіях Android) [12]¹⁾.

¹⁾[12] Android Developers Blog (загол. з екрану). URL : <https://android-developers.googleblog.com/2020/05/android-studio-4.html>. (дата звернення – 08.12.2020).

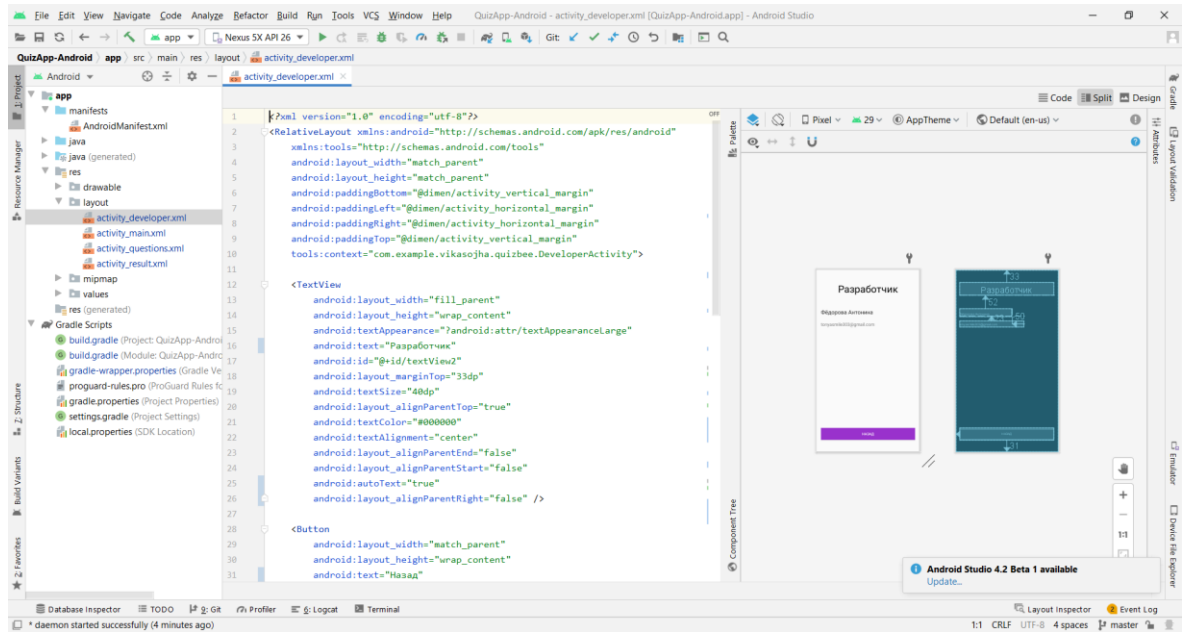


Рисунок 2.1 – Зовнішній вигляд IDE Android Studio

– feature – on – feature залежності (рис. 2.2).

При використанні плагіна Gradle 4.0.0 і вище можна вказати, що модуль Dynamic Feature залежить від іншого модуля. Можливість визначити цей взаємозв'язок гарантує, що додаток буде мати необхідні модулі для розблокування додаткових функціональних можливостей, а це призведе до меншої кількості запитів і простішій структурі програми.

Наприклад, функція: video може залежати від функції: camera. Якщо користувач хоче розблокувати можливість запису відео, додаток автоматично при запиті: video завантажить модуль: camera;

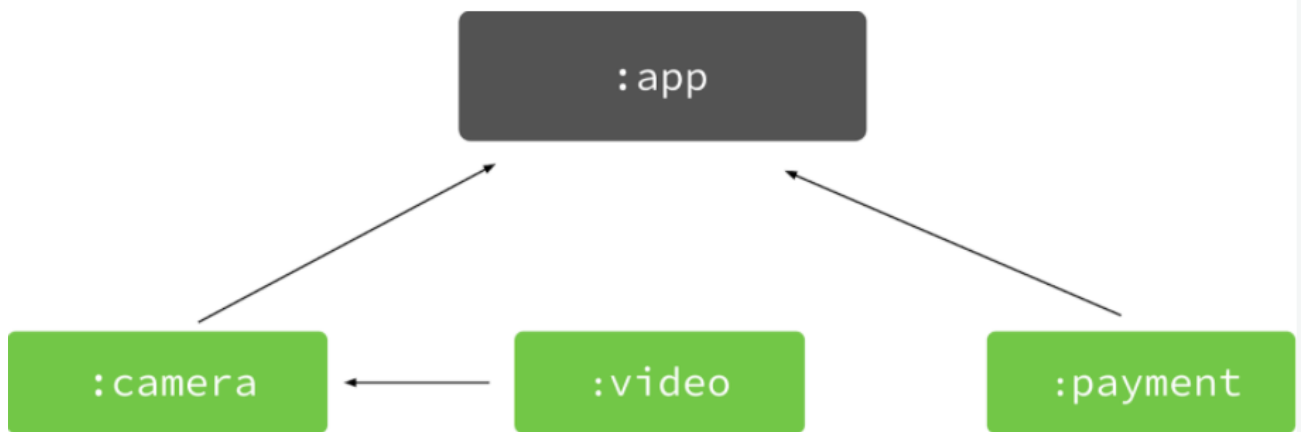


Рисунок 2.2 – Схема залежностей властивостей

- підтримка файлів Kotlin DSL (студія має вбудовану підтримку файлів сценаріїв збірки Kotlin DSL (*.kts). Сценарії збірки Kotlin пропонують повний набір швидких виправлень і підтримуються Project Structure dialog);
- метадані залежностей (при створенні програми Gradle включає метадані, що описують залежності компільованих бібліотек).

При завантаженні програми Play Console перевіряє ці метадані, отримуючи інформацію про відомі проблеми з SDK і залежностями, використовуваними в додатку, а в деяких випадках надає зворотний зв'язок для вирішення цих проблем.

Кожен проект в Android Studio містить один або кілька модулів з файлами вихідного коду і файлами ресурсів. Типи модулів включають:

- модулі – додатки для Android;
- модулі бібліотеки;
- модулі механізму – додатки Google.

За змовчуванням, Android Studio виводить на екран файли проекту в уявленні проекту Android (рис. 2.3). Це уявлення організовано модулями, щоб забез

печити швидкий доступ до файлів ключового джерела проекту.

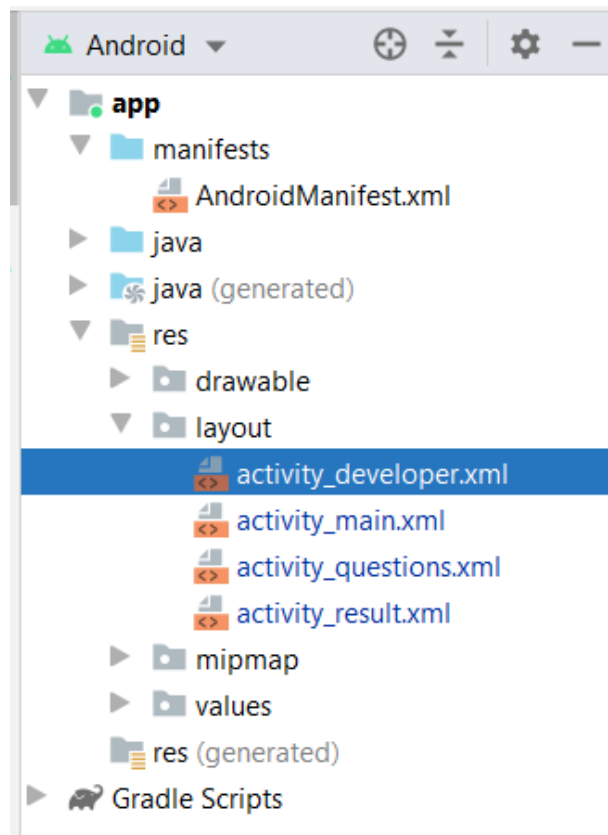


Рисунок 2.3 – Вигляд проекту в Android Studio

Всі файли типу "build" видно на верхньому рівні відповідно зі сценаріями Gradle, і кожен модуль програми містить наступні папки:

- manifests: містить файл AndroidManifest.xml;
- java: містить файли вихідного коду Java, включаючи тестовий код JUnit;
- res: містить всі ресурси не коду , такі як розмітки XML, рядки UI (призначений для користувача інтерфейс) і растрові зображення.

Android Studio підтримує безліч систем управління версіями (VCS – version control systems), включаючи Git, GitHub, CVS, Mercurial, Subversion і Google Cloud Source Repositories.

2.2.2 Інтегроване середовище Microsoft Visual Studio

Інтегроване середовище розробки Visual Studio – це стартовий майданчик для написання і налагодження коду, а також подальшої публікації додатків.

Інтегроване середовище розробки (IDE) являє собою багатофункціональну програму, яку можна використовувати для різних аспектів розробки програмного забезпечення. Крім стандартного редактора і відладчика, які існують в більшості середовищ IDE, Visual Studio включає в себе компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для спрощення процесу розробки [13]¹⁾.

Декілька основних інструментів середовища:

- оглядач рішень дозволяє переглядати файли коду, пересуватися по ньому і управляти ними. Оглядач рішень дозволяє впорядкувати код шляхом об'єднання файлів в рішення і проекти;
- у вікні редактора (центр), відображається вміст файлу. Тут можна редагувати код або розробляти призначений для користувача інтерфейс, наприклад вікно з кнопками або текстові поля;
- Team Explorer дозволяє відслідковувати робочі елементи і використовувати код спільно з іншими користувачами за допомогою технологій управління версіями, таких як Git і система управління версіями Team Foundation (TFVC).

Можливості Visual Studio:

- Visual Studio пропонує набір інструментів, що дозволяють з легкістю створювати хмарні додатки на базі Microsoft Azure. Вона спрощує настройку, складання, налагодження, упаковку і розгортання додатків і служб в Microsoft Azure прямо з IDE;

¹⁾[13] Документація по Visual Studio (загол. з екрану). URL : <https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2019&preserve-view=true>. (дата звернення – 08.12.2020).

- створювати веб-додатки за допомогою технологій ASP.NET, Node.js, Python, JavaScript і TypeScript. Visual Studio розпізнає такі платформи для створення веб-додатків, як Angular, jQuery, Express і інші. Платформи ASP.NET Core і .NET Core підтримуються на комп'ютерах з Windows і Linux, а також на комп'ютерах Mac. ASP.NET Core є суттєве оновлення для MVC, WebAPI і SignalR, яке працює на платформах Windows, Mac і Linux. Платформа ASP.NET Core була розроблена з нуля і надає компактний і змінюваний стек .NET для розробки сучасних хмарних веб-додатків і служб;
- за допомогою Visual Studio можуть створювати додатки та ігри для платформ macOS, Linux і Windows, а також для мобільних пристроїв Android, iOS і ін.
- оглядач сервера дозволяє переглядати екземпляри і ресурси SQL Server в локальній або віддаленій середовищі, в Azure, Salesforce.com, Microsoft 365 і на веб-сайтах, а також керувати імі. SQL Server Data Tools (SSDT) – це потужне середовище розробки для SQL Server, бази даних SQL Azure і сховища даних SQL Azure. За допомогою SSDT ви можете створювати і обслуговувати бази даних, а також виконувати їх налагодження і рефакторинг. Можна працювати з проектом бази даних або безпосередньо з підключеним екземпляром бази даних (локально або віддалено).

2.2.3 Інтегроване середовище Eclipse Java IDE

Eclipse є безкоштовною програмною платформою з відкритим вихідним кодом, контролюється організацією Eclipse Foundation. Написана на мові програмування Java і основною метою її створення є підвищення продуктивності процесу розробки програмного забезпечення. IDE розроблені на базі платформи Eclipse застосовуються для створення програмного забезпечення

на різних мовах програмування, так як Eclipse є платформою для розробки будь-яких інтегрованих середовищ програмування і розширень для себе ж, за принципом "Додатки для Eclipse розробляються в самій Eclipse".

Особливості платформи Eclipse :

- кросплатформенність – працює під операційними системами Windows, Linux, Solaris і Mac OS X. Використовуючи Eclipse можна програмувати на багатьох мовах, таких як Java, C і C ++, PHP, Perl, Python, Cobol та інших;
- є фреймворком для розробки інших інструментів і пропонує великий набір API для створення модулів;
- використовуючи підхід RCP (Rich Client Platform) Eclipse є інструментом для створення практично будь-якого клієнтського програмного забезпечення;
- робота над проектом Eclipse ведеться в декількох напрямках, основні три – робота над платформою Eclipse, розробка Java IDE, розробка плагінів для розширення функціональності Eclipse;
- гнучкість і розширюваність досягається завдяки модульності платформи.

Архітектура платформи Eclipse:

- основним елементом є Eclipse Runtime, в якій виконуються коди розширень і модулів. Вона забезпечує всю базову функціональність платформи управління розширеннями і оновленнями, взаємодія з операційною системою, забезпечення роботи системи допомоги;
- наступним елементом є власне IDE – вона відповідає за управління основними елементами програми, їх розташуванням і настройками, управління проектами, налагодження та складання проектів, пошук по файлах і командну розробку.
- у стандартну поставку Eclipse SDK включені два плагіна – Java Development Tools або JDT, і Plug-in Developer Environment або PDE,

таким чином ми отримуємо повністю готову IDE для Java програмування і для розробки розширень для Eclipse.

Eclipse SDK – це мінімальна версія, ідеальна для знайомства з платформою і навчання. Надалі, визначившись з цілями, ви можете завантажити і використовувати будь-яку відповідну для ваших завдань складання, вже укомплектовану необхідними розширеннями.

У Eclipse також є платна версія – MyEclipse, але це скоріше окремий проект, побудований на базі Eclipse. MyEclipse пропонує повнофункціональну платформу для розробки ПО, а також додаткові пакети.

Eclipse, на відміну від платної версії, – це ядро, до якого підключаються додаткові плагіни для створення IDE мрії.

2.2.4 Порівняний аналіз представлених IDE

Android Studio – це нове середовище розробки Android, заснована на IntelliJ IDEA. Подібно Eclipse з Android Development Tools для Eclipse, Android Studio надає інтегровані засоби розробки Android для розробки і налагодження.

Краща IDE, доступна для розробки Android, з обмеженими можливостями апаратного забезпечення на пристрої розробки.

Microsoft Visual Studio – це інтегроване середовище розробки (IDE) від Microsoft. Він використовується для розробки комп'ютерних програм для Microsoft Windows, а також для веб-сайтів, веб-додатків і веб-сервісів. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation, Windows Store і Microsoft Silverlight. Вона може створювати як власний код, так і керований код.

Visual Studio підтримує різні мови програмування і дозволяє редактору коду і отладчику підтримувати (в тій чи іншій мірі) практично будь-яку мову

програмування, якщо існує специфічна для мови служба. Вбудовані мови включають в себе C, C ++ і C ++ / CLI (через Visual C ++), VB.NET (через Visual Basic .NET), C # (через Visual C #) і F # (з Visual Studio 2010). Підтримка інших мов, таких як M, Python і Ruby, серед інших доступна через мовні служби, встановлені окремо. Він також підтримує XML / XSLT, HTML / XHTML, JavaScript і CSS.

Eclipse – це розширювана платформа розробки з часом виконання і інфраструктурами додатків для створення, розгортання та управління програмним забезпеченням протягом усього життєвого циклу програмного забезпечення. Багато людей знають Eclipse як Java IDE, але це набагато більше, ніж Eclipse. Насправді Eclipse складається з понад 60 різних проектів з відкритим вихідним кодом, організованих в 6 різних категорій:

- розвиток підприємництва;
- впровадження і розробка пристроїв;
- багата клієнтська платформа;
- багаті інтернет-додатки;
- рамки додатків;
- управління життєвим циклом додатків (ALM);
- сервіс-орієнтована архітектура (SOA).

Порівняний аналіз усіх трьох IDE зображено на табл. 2.1.

Таблиця 2.1 – Порівняний аналіз IDE

IDE	Android Studio	Microsoft Visual Studio	Eclipse
1	2	3	4
Платформи	Mac Windows Linux	Mac Windows .NET Framework Інтернет	Mac Windows Linux

Продовження таблиці 2.1

1	2	3	4
Можливості	Програмування Закінчення коду Інтегрований емулятор Android	Автозаповнення Закінчення коду Встроєний відладчик API для розробки Кольорове кодування C# Scripting NuGet Компілятор Exe Рефакторинг	Розширюється плагінами/ розширеннями Автозаповнення Середовище для розробки Java Рефакторинг Інтеграція SCM
Код	Відкритий	Закритий	Відкритий
Ціна	Безкоштовно	Community – безкоштовна Enterprise – 250\$/місяць Professional – 45\$/місяць	Безкоштовно

Для створення гри «Вікторина» було обрано IDE Android Studio, для створення кросворду – Visual Studio.

3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Проектування інформаційної системи додатку за допомогою діаграми опису бізнес-процесів IDEF0

IDEF0 – це методологія графічного опису систем і процесів діяльності організації як безлічі взаємозалежних функцій. Вона дозволяє досліджувати функції організації, не пов'язуючи їх з об'єктами, що забезпечують їх реалізацію. У стандарті IDEF0 за допомогою входу показують об'єкти - інформаційні та матеріальні потоки, які перетворюються в бізнес-процесі. За допомогою управління показуються об'єкти - матеріальні та інформаційні потоки, які перетворюються на процесі, по потрібні для його виконання. Використовуючи механізми IDEF0 можна відображати інструменти та ресурси, за допомогою яких бізнес-процес реалізується (наприклад, технічні засоби, люди, інформаційні системи і т.д.). Вихід бізнес-процесу, описаного в стандарті IDEF0, повністю відповідає за змістом виходу процесу, описаного за допомогою DFD-схеми [14]¹⁾.

Головною метою та бізнес процесом додатку є створення вікторини в одно-користувальницькому режимі. Контекстна діаграма цього бізнес процесу у стандарті IDEF0, наведена на рис. 3.1.

Опис контекстної діаграми:

- ім'я бізнес-процесу: проведення однокористувацького режиму;
- основні учасники: користувач;
- вхідна подія: запит на вибір предмету вікторини; відповіді на питання вікторини;
- вихідна подія: результат вікторини.

¹⁾[14] Методологія IDEF0. (загол. з екрану). URL : https://studme.org/87184/ekonomika/metoologiya_idef0/. (дата звернення – 10.12.2020).

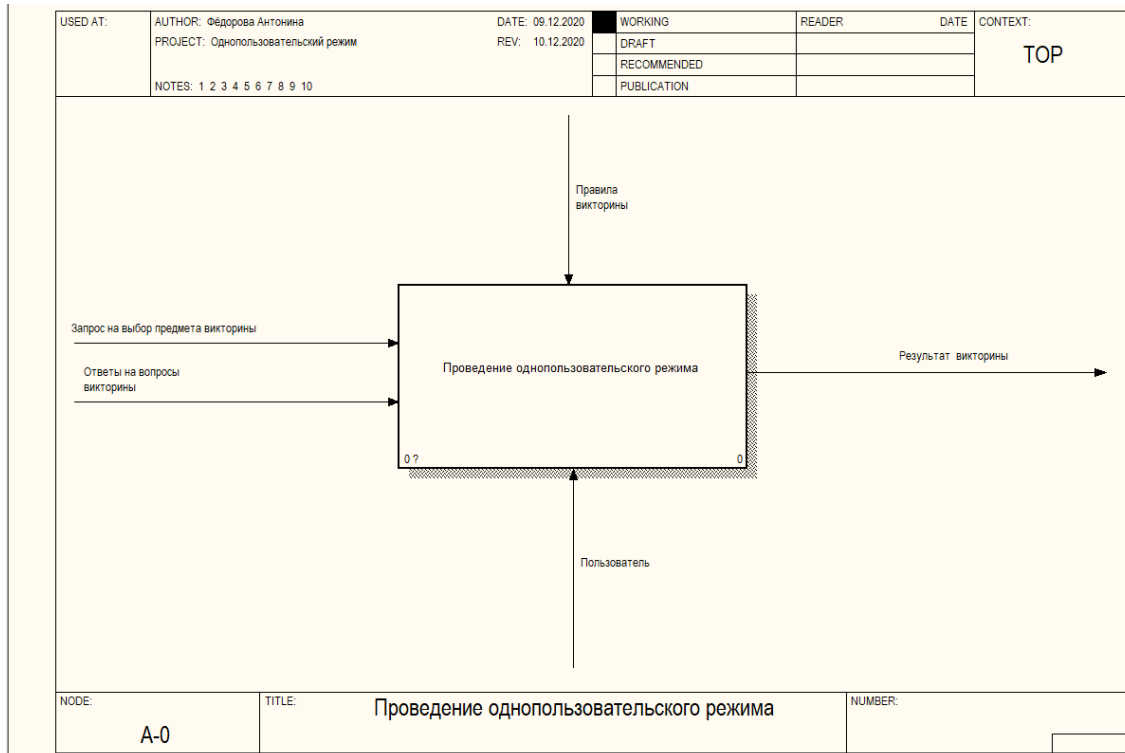


Рисунок 3.1 – Контекстна діаграма бізнес-процесу «Проведення одно-користувальницького режиму»

Внаслідок декомпозиції контекстної діаграми були виділені наступні роботи:

- формування вікторини;
- аналіз результатів відповідей.

Діаграма першого рівня декомпозиції наведена на рис. 3.2.

Опис першого рівня декомпозиції:

- характеристика бізнес – процесу «Формування вікторини»:
 - основні учасники: користувач;
 - вхідна подія: запит на вибір предмету вікторини;
 - вихідна подія – вибрані питання вікторини;
- характеристика бізнес – процесу «Аналіз результатів відповідей»:
 - основні учасники: користувач;

- вхідна подія: вибрані питання вікторини; відповіді на питання вікторини;
- вихідна подія: результат вікторини.

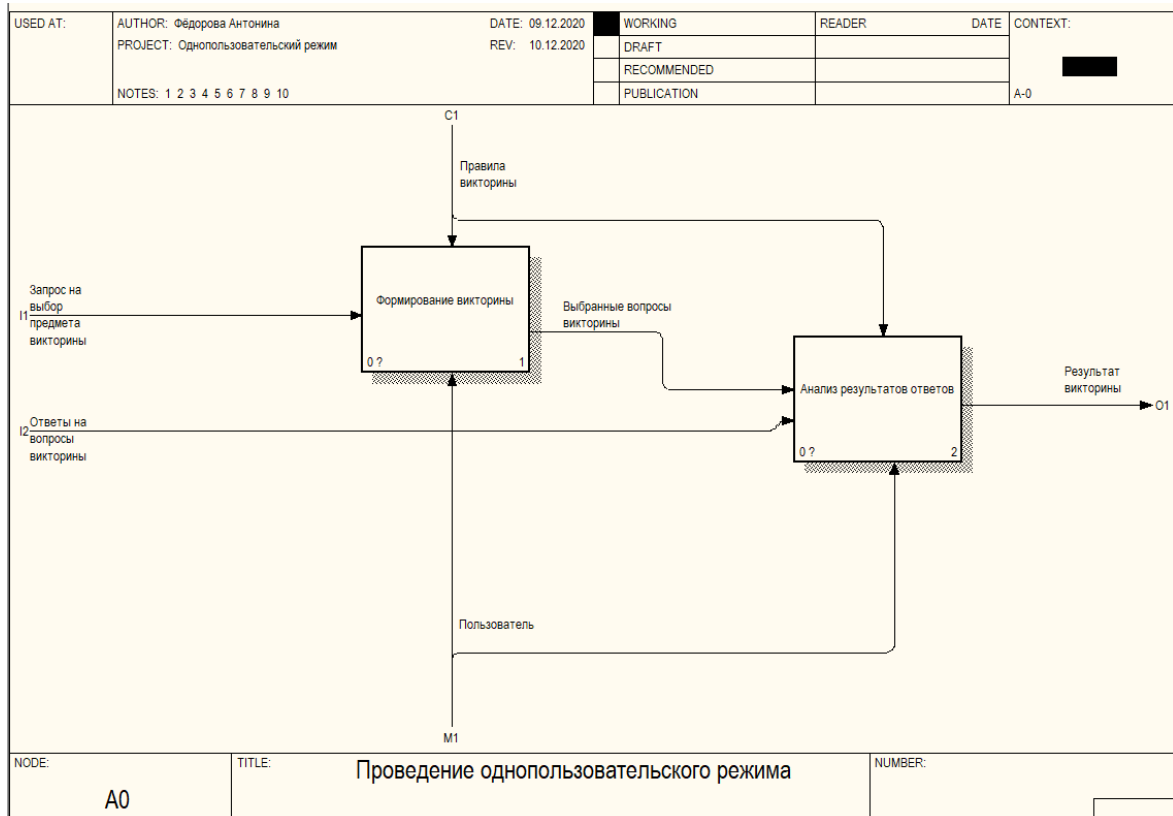


Рисунок 3.2 – Перший рівень декомпозиції бізнес-процесу «Проведення одно користувальницького режиму»

Внаслідок декомпозиції бізнес-процесу «Формування вікторини» були виділені наступні роботи:

- вибір предмету вікторини;
- отримання питань вікторини.

Діаграма декомпозиції бізнес-процесу «Формування вікторини» наведена на рис. 3.3.

Опис декомпозиції бізнес - процесу «Формування вікторини»:

- характеристика бізнес – процесу «Вибір предмету вікторини»:

- основні учасники: користувач;
 - вхідна подія: запит на вибір питання вікторини;
 - вихідна подія: предмет вікторини;
- характеристика бізнес – процесу «Отримання питань вікторини»:
- основні учасники: користувач;
 - вхідна подія: предмет вікторини;
 - вихідна подія: вибрані питання вікторини.

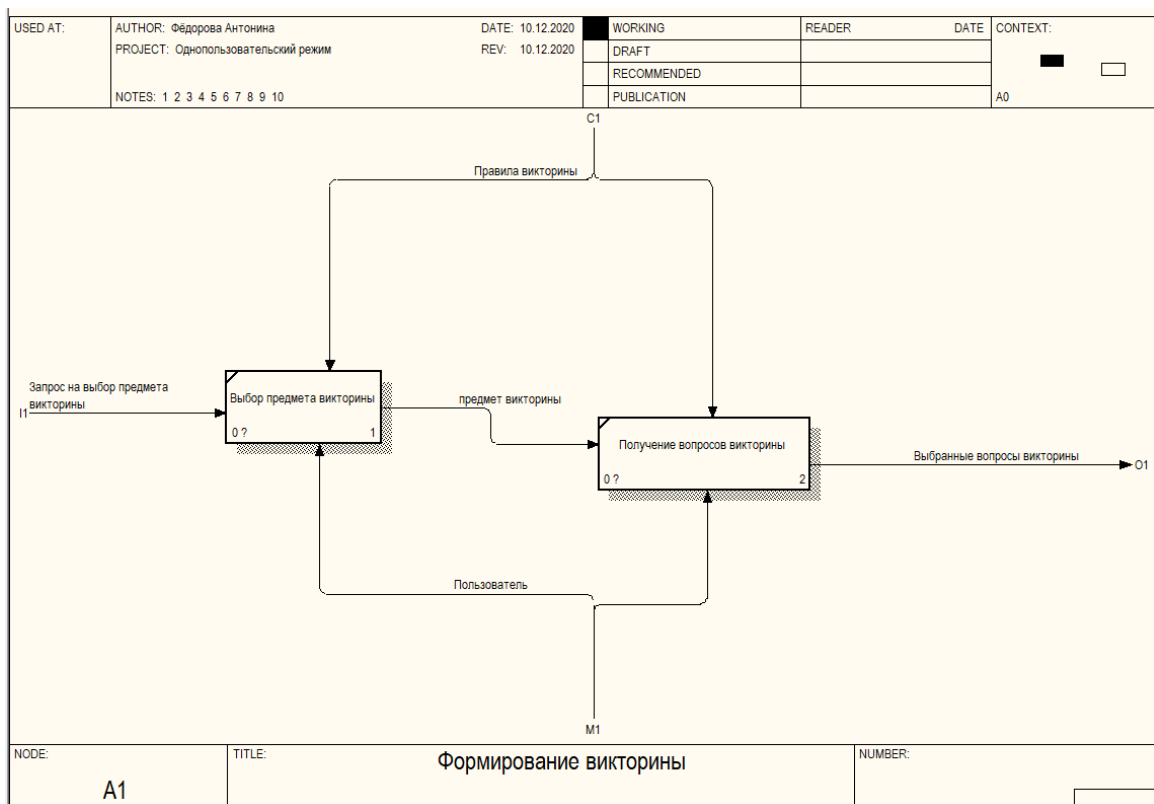


Рисунок 3.3 – Декомпозиція бізнес-процесу «Формування вікторини»

Внаслідок декомпозиції бізнес-процесу «Отримання відповідей вікторини» були виділені наступні роботи:

- надання відповідей на вікторину;
- обробка відповідей;
- формування результатів.

Діаграма декомпозиції бізнес-процесу «Аналіз результатів вікторини» наведена на рис. 3.4.

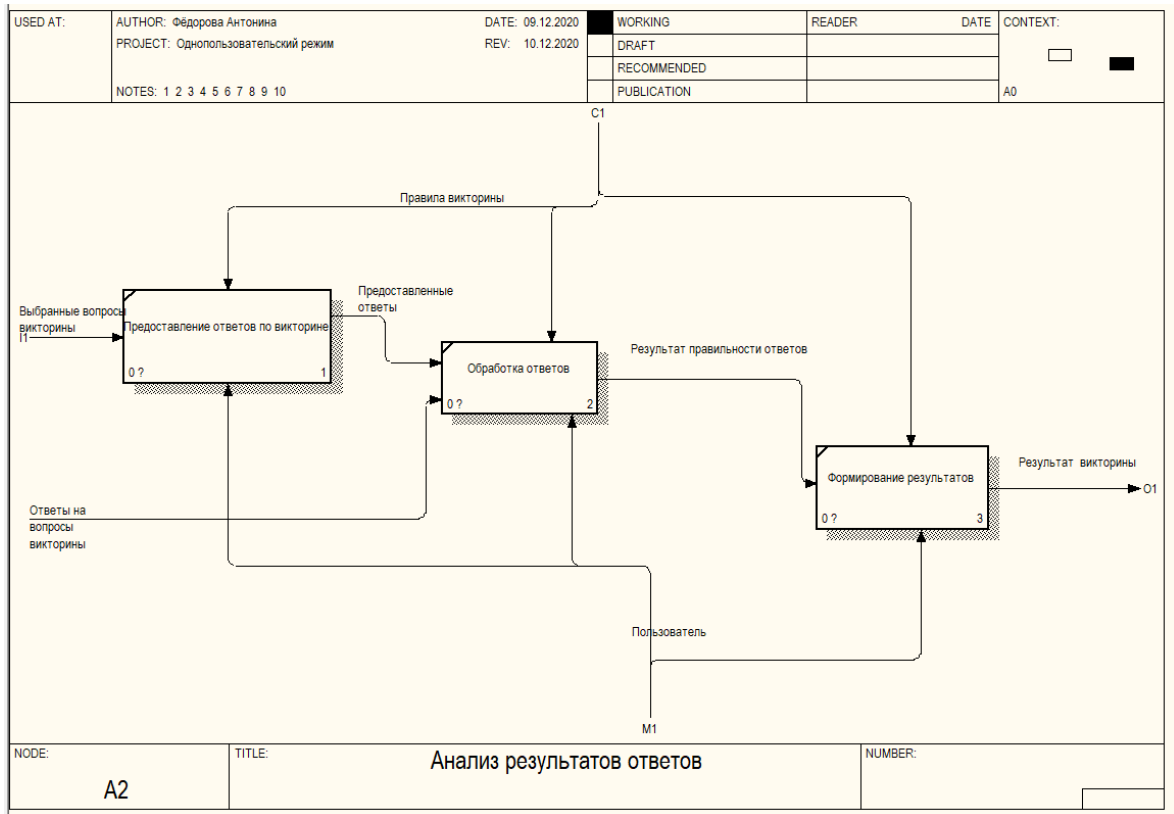


Рисунок 3.4 – Декомпозиція бізнес-процесу «Аналіз результатів вікторини»

Опис декомпозиції бізнес - процесу «Аналіз результатів вікторини»:

- характеристика бізнес – процесу «Надання відповідей на вікторину»:
 - основні учасники: користувач;
 - вхідна подія: вибрані питання вікторини;
 - вихідна подія: надані відповіді;
- характеристика бізнес – процесу «Обробка відповідей»:
 - основні учасники: користувач;
 - вхідна подія: надані відповіді; відповіді на питання вікторини;
 - вихідна подія: результат правильності відповідей;
- характеристика бізнес – процесу «Формування результатів»:

- основні учасники: користувач;
- вхідна подія: результат правильності відповідей;
- вихідна подія: результат вікторини.

Блок-схема мобільного додатку представлена на рис. 3.5.



Рисунок 3.5 – Блок – схема мобільного додатку «Перевір себе»

3.2 Розроблення діаграми варіанту використання

Для розробки варіантів використання необхідно побудувати діаграму варіантів використання розроблюваного додатку (рис. 3.6). Діаграми варіантів використання описує функціональне призначення системи або, іншими словами, те, що система буде робити в процесі свого функціонування.

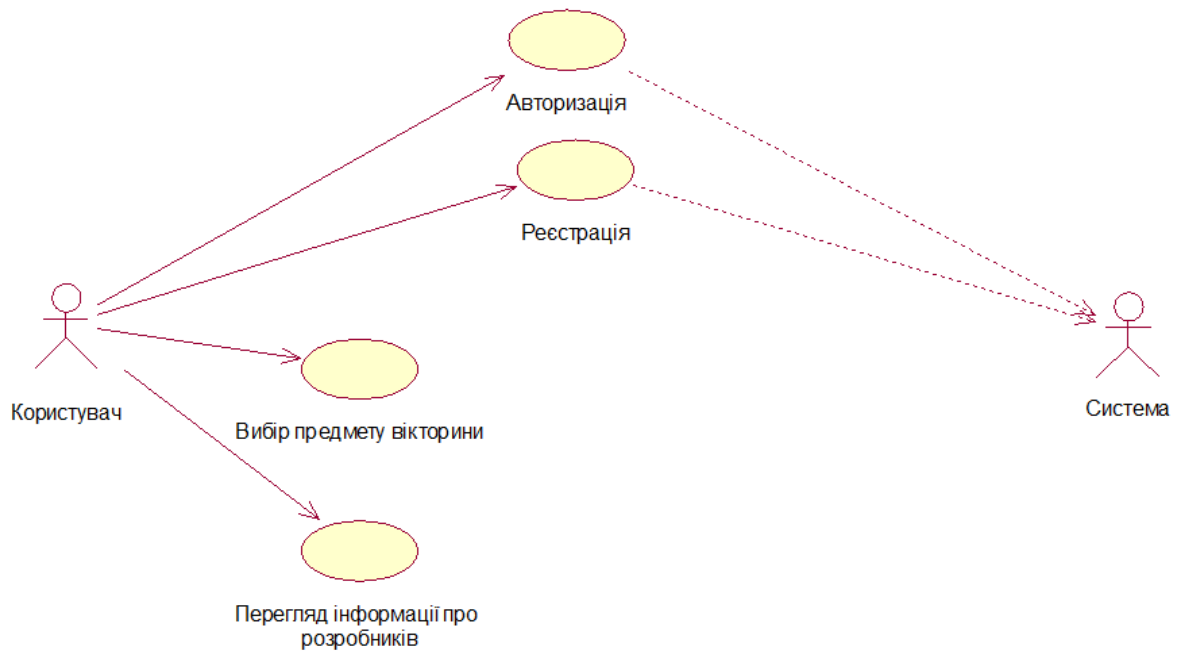


Рисунок 3.6 – Діаграма варіантів використання

Діаграма варіантів використання є вихідним концептуальним поданням чи концептуальною моделлю системи в процесі її розробки і впровадження. Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей або акторів, які взаємодіють з системою за допомогою, так званих варіантів використання. При цьому актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на модельовану систему так, як визначить сам розробник.

У свою чергу, варіант використання (use case) служить для опису сервісів, які система надає акторові. Іншими словами, кожен варіант використання визначає певний набір дій, який чинять системою при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодія акторів із системою.

У ході проектування діаграми використання були виявлені наступні основні варіанти використання:

- Вибір предмету вікторини. Користувач вибирає предмет вікторини.
- Авторизація користувача. Користувач авторизується системі для того, щоб при збереженні результату у таблиці рекордів було його ім'я.

Характеристика варіанту використання «Вибір предмету вікторини»:

- контекст використання: вибір предмету вікторини;
- дійові особи: користувач;
- передумова: особа користувача авторизована;
- тригер: користувач хоче вибрати предмет вікторини;
- сценарій: перейти на головне вікно програми, вибрати предмет;
- постумова: якщо варіант використання виконується успішно – вибирається тема вікторини, інакше – змін немає.

Характеристика варіанту використання «Авторизація користувача»:

- контекст використання: авторизація користувача;
- дійові особи: користувач;
- передумова: особа користувача зареєстрована;
- тригер: користувач бажає зареєструватися;
- сценарій: відкрити додаток, ввести логін і пароль;
- постумова: якщо варіант використання виконується успішно – користувач входить у головне меню, інакше – з'являється повідомлення про те, що логін або пароль невірні.

3.3 Проектування бази даних мобільного додатку

Основним компонентом БД є таблиця. Таблиця використовується для структуризації та зберігання інформації. В середині однієї БД існують зв'язки між таблицями, кожна з яких задає спільне користування даними таблиці. Ді-

аграма "Сутність-Зв'язок" графічно представляє структуру даних проектованої системи [15]¹⁾.

Визначивши функціональні вимоги, використовуючи моделювання SQL, спроектована база даних, що реалізує модель «Сутність-Зв'язок». При розробці системи була створена база даних, за допомогою СУБД Microsoft SQL Server.

Перший етап процесу проектування бази даних називається концептуальним проектуванням бази даних. Він полягає у створенні концептуальної моделі даних предметної області. Ця модель даних створюється на основі функціональних вимог користувачів. Концептуальне проектування бази даних абсолютно не залежить від таких подробиць її реалізації, як тип обраної СУБД, набір створюваних прикладних програм, використовувані мови програмування, тип обраної обчислювальної платформи, а також від будь-яких інших особливостей фізичної реалізації. Концептуальне проектування – створення концептуального уявлення бази даних, що включає визначення типів найважливіших сутностей та існуючих між ними зв'язків і атрибутів. Послідовність етапів проектування концептуальної моделі даних: визначення сутностей; визначення взаємозв'язків між сутностями; визначення атрибутів сутностей; завдання первинних і альтернативних ключів.

3.3.1 Опис таблиць бази даних і зв'язків між ними

Для інформаційної системи в базі даних були визначені наступні сутності: Користувач, Предмет, Питання, Результат. Кожна сутність містить первинний ключ, призначений для ідентифікації екземпляра сутності. Первинний ключ має бути вибраним таким чином, щоб за значеннями атрибутів, можна було точно ідентифікувати сутність, крім того жоден з атрибутів пер-

¹⁾[15] Представление данных с помощью модели «сущность-связь». (загол. з екрану). URL : http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_1.html. (дата звернення – 29.10.2020)

винного ключа не повинен мати нульове значення. Значення атрибутів первинного ключа не повинні змінюватися.

Для визначення первинного ключа часто використовують унікальні номери, які можуть автоматично генеруватися системою при додаванні екземпляра сутності в БД. Застосування унікальних номерів полегшує процес індексації та пошуку в БД.

Далі необхідно визначити атрибути для кожної сутності, а також записати для всіх сутностей первинні ключі. Тепер розставити зв'язки між сутностями [16]¹⁾. Уявімо базу даних мобільного додатку у вигляді моделі «Сутність-Зв'язок» (рис. 3.7).

Сутність Користувач містить наступні атрибути: номер (первинний ключ), ім'я користувача, логін, пароль.

Сутність Предмет містить наступні атрибути: номер (первинний ключ), назва предмету.

Сутність Питання містить наступні атрибути: номер (первинний ключ), питання, відповідь 1, відповідь 2, відповідь 3, правильна відповідь, неправильна відповідь.

Сутність Результат містить наступні атрибути: номер (первинний ключ), кількість правильних відповідей, кількість неправильних відповідей, результат.

Між таблицями Користувач і Предмет існує зв'язок типу «один-до-багатьох»: одному користувачу може відповідати кілька предметів.

Розглянемо типи даних кожного атрибута сутності.

Відомості про типи даних наведені нижче у вигляді таблиць (табл. 3.1-3.5).

¹⁾ [16] Представление данных с помощью модели «сущность-связь. (загол. з екрану). URL: http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_1.html. (Дата звернення – 06.11.2020).

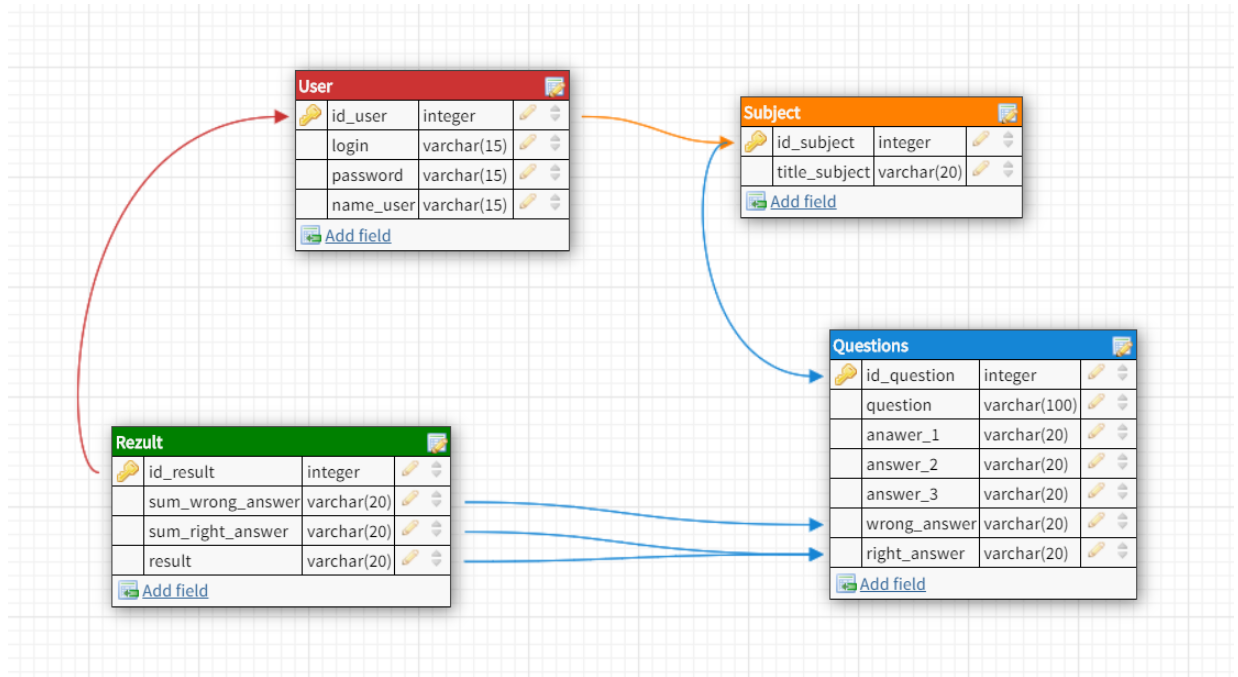


Рисунок 3.7 – Діаграма «Сутність-Зв'язок» мобільного додатку «Перевір себе»

Таблиця 3.1 – Список всіх сутностей бази даних інформаційної системи

User	Користувач
Subject	Предмет
Questions	Питання
Result	Результат

Таблиця 3.2 – Користувач (user)

№	Поле	Атрибут	Тип
1	id_user	Унікальний ідентифікатор користувача	integer
2	name_user	Ім'я користувача	varchar(15)
3	login	Логін користувача	varchar(15)
4	password	Пароль користувача	varchar(15)

Таблиця 3.3 – Предмет (subject)

№	Поле	Атрибут	Тип
1	id_subject	Унікальний ідентифікатор предмету	integer
2	tittle_subject	Назва предмету	varchar(20)
3	name_user	Ім'я користувача	varchar(15)

Таблиця 3.4 – Питання (questions)

№	Поле	Атрибут	Тип
1	id_question	Унікальний ідентифікатор питання	integer
2	question	Питання	varchar(100)
3	answer_1	Відповідь 1	varchar(20)
4	answer_2	Відповідь 2	varchar(20)
5	answer_3	Відповідь 3	varchar(20)
6	wrong_answer	Неправильна відповідь	varchar(20)
7	right_answer	Правильна відповідь	varchar(20)

Таблиця 3.5 – Результат (result)

№	Поле	Атрибут	Тип
1	id_result	Унікальний ідентифікатор результата	integer
2	sum_wrong_anwer	Кількість неправильних відповідей	varchar(20)
3	sum_right_anwer	Кількість правильних відповідей	varchar(20)
4	result	Підсумковий результат	varchar(20)

3.4 Алгоритм генерації кросворду

3.4.1 Метод повного перебору

До алгоритмів повного перебору звертаються тоді, коли немає інших

способів вирішити певне завдання. Наприклад, нікому в голову не прийде вирішувати квадратне рівняння перебором всіх можливих значень, навіть якщо відомо, що рішенням є цілі числа. Повний перебір – це рішення "в лоб", побудоване на підстановці і перевірці всіх можливих варіантів з області допустимих значень. На словах у вирішенні завдання повного перебору немає нічого складного, але на ділі алгоритми, які дійсно шукають і знаходять рішення подібним способом за "прийнятний час" досить таки непрості. До найбільш популярного завдання, вирішення якого методом повного перебору розглядають в школах і ВНЗ (в рамках дисциплін, пов'язаних з комбінаторики і програмуванням) можна віднести заповнення прямокутної області фігурками з тетрису, пошук виходу з лабіринту, завдання про рюкзак, пошук пов'язаних областей і інші. Багато задач на графах також можна вирішувати за допомогою повного перебору, хоча для їх вирішення придумані спеціальні методики – оптимальні з точки зору кількості операцій алгоритми, які здійснюють пошук рішень на порядок швидше, ніж звичайний повний перебір. До таких завдань відносять пошук найкоротшого маршруту між вершинами (алгоритм Дейкстри) або пошук максимального пропускнуго потоку в графі. Якщо узагальнити все вищесказане, то для вирішення деякої задачі методом повного перебору потрібно розробити алгоритм перебору всіх можливих кандидатів на рішення і реалізувати процедуру перевірки коректності рішення для кожного з них.

Обчислювальна складність алгоритмів повного перебору оцінюється за такою узагальненою формулою:

$$O(N! / (N-R)!) \quad (3.1)$$

де, N – кількість елементів в області пошуку рішень.

R – кількість елементів, що утворюють одне ціле рішення задачі.

Якщо повернутися до задачі пошуку всіх варіантів заповнення крос-

ворду, то її складність, в найпростішому варіанті, можна оцінити так:

$$N * (N-1) * \dots * (N - (W-1)) \quad (3.2)$$

де, N – кількість слів у словнику, а W – кількість шаблонів слів в кросворді. На кожному наступному кроці належить перебрати на одне слово менше, ніж на попередньому, оскільки повторюватися у виборі слів не можна. А це і є не що інше, як:

$$N! / (N-W)! \quad (3.3)$$

Природно, що слів у словнику має бути не менше, ніж їх шаблонів в кросворді, інакше рішень взагалі немає. Якщо слів в кросворді більше десятка, а в словнику для його заповнення, наприклад, більше тисячі слів, то загальне число варіантів, які потрібно перебрати для пошуку всіх рішень вразить кількістю розрядів навіть "бувалих" суперкомп'ютери. З усього це впливає одне, що якщо завдання не вирішується ніяк, крім повного перебору, то алгоритм цього повного перебору повинен бути "оптимальним". Слово "оптимальний" я взяв у лапки, оскільки строге доведення оптимальності таких алгоритмів завдання ще більш складна, ніж розробка самих алгоритмів. У контексті цієї статті замість терміна "оптимальний" я буду використовувати термін "задовільний". Задовільним будемо вважати той алгоритм, який знаходить хоча б одне рішення задачі за задовільний для користувача час. Цей термін "розмитий" і має мало спільного з точними науками, до яких відноситься алгоритмізація, але в нашому випадку вибирати не доводиться.

3.4.2 Оптимізація алгоритму повного перебору

Оптимізація будь-якого алгоритму повинна проходити, як мінімум, за двома напрямками: в напрямку мінімізації кількості елементарних операцій і в напрямку оптимізації швидкості виконання цих елементарних операцій. Другий напрямок в основному пов'язано з оптимізацією використовуваних структур даних і операцій доступу до них. Найбільш популярними методами скорочення часу виконання елементарних операцій є кешування часто використовуваних даних, відкладання ініціалізації і різного роду обчислень до моменту, коли в них з'явиться необхідність, побітно порівняння і т.п. Потрібно розуміти, що витрата пам'яті і швидкість виконання операцій – це дві протилежні характеристики. За рахунок створення додаткових "хитрих" структур даних в оперативній пам'яті можна збільшити швидкодію і навпаки: якщо не зберігати результати проміжних обчислень і намагатися всі необхідні розрахунки кожен раз робити заново, то можна виграти в обсягах пам'яті, але істотно програти в швидкості роботи програми. Зізнаюся, особливих вишукувань, пов'язаних з цього роду оптимізаціями я не проводив, просто намагався не робити зайвих рухів і непотрібних дій.

Алгоритми повного перебору досить часто реалізують за допомогою рекурсії: на кожному кроці рекурсії здійснюють пошук частини рішення для одного елемента, і, знайшовши його, рекурсивно запускають той же пошук для наступного елемента. Якщо для поточного елемента рішення знайти не вдається, то здійснюється перехід до попереднього елемента (повернення з рекурсії на один крок), де знайдено рішення скасовується і починається пошук іншого варіанту.

Як тільки знайдені рішення для всіх елементів – знайдено всі рішення. Далі, алгоритм зберігає варіант рішення і продовжує шукати інші. В ході аналізу рекурсивного алгоритму повного перебору дуже важливо вміти давати оцінку, чи містить поточна гілка рекурсії рішення, або її можна відсікти.

Наприклад, чи варто намагатися розставляти інші слова в кросворді, коли ви натрапили на нерозв'язне перетин: в словнику немає слів потрібної довжини для підстановки в шаблон з дотримань умов перетину. Аналіз і відсікання свідомо "безрезультатних" гілок рекурсії називається методом гілок і меж. Оптимізація алгоритмів повного перебору може проводитися з використанням евристичного підходу до вирішення завдання. Цей підхід пов'язаний з набором припущень, правильність яких довести або складно або формально неможливо, але дотримання їх дозволяє знаходити рішення досить швидко. Нагадаю, що мета підприємляемой оптимізації – якомога швидше знайти хоча б одне рішення, а не всі варіанти заповнення кросворду. Таким чином, оптимізація рекурсивного алгоритму повного перебору, спрямована на мінімізацію кількості елементарних операцій (операцій порівняння, вибору, тощо) здійснюється за рахунок застосування двох підходів:

- пошук і відсікання гілок рекурсії, які свідомо не містять рішень – метод гілок і меж;
- вибір, в першу чергу, тих гілок рекурсії, які містять найбільшу кількість рішень – евристичний підхід.

3.4.3 Оптимізація алгоритму генерації кросворду

У двох словах представлений алгоритм робить наступне: спочатку розставляє слова в одному напрямку – перша лінія (`GenerateFirstLine`), а потім намагається розставити слова в іншому напрямку з урахуванням перетинів – друга лінія (`GenerateSecondLine`). Зрозуміло, що в ході заповнення першої лінії не потрібно враховувати ніякі перетину, а досить просто підбирати слова відповідної шаблоном довжини.

Перша оптимізація – оптимізація використовуваних структур даних: розбити весь словник на розділи, кожен з яких містить слова тільки однієї

довжини, і організувати швидкий доступ до розділу по заданій довжині. Зроблено це з використанням словника на базі хеш-таблиці, значеннями якого є списки слів – розділи, а ключем – довжина слів у розділі. Оскільки перебір слів першої лінії виконується швидше (не враховуються перетину), то першою лінією є той напрямок, кількість шаблонів слів якого більше.

Представлений алгоритм є рекурсивним: на кожному кроці здійснюється пошук відповідного варіанту для конкретного шаблону слова і, якщо пошук вдалий, здійснюється перехід до наступного шаблону. Рішення для послідовності кроків в рамках однієї лінії слабо залежні, оскільки пов'язані тільки перевіркою, чи не використано конкретне слово зі словника при заповненні шаблону на попередніх кроках (заповнення декількох шаблонів слів одним і тим же значенням заборонено!). Але кроки другої лінії дуже і дуже залежать від знайдених рішень першої лінії.

Друга оптимізація – евристичний підхід. Робимо припущення, що якщо шаблони слів, що заповнюються на першій лінії впорядкувати за зростанням, а шаблони слів другої лінії – по спадаючій довжини, то це допоможе швидше знайти перший готовий варіант заповненого кросворду. Мотивація наступна:

Найбільша кількість повернень на попередні кроки рекурсії буде відбуватися на межі першої і другої лінії.

Чим більше довжина поточного шаблону слова, тим серйозніше впливає його значення на хід пошуку за все залишився рішення.

Таким чином, найбільш довгі шаблони слів сконцентровані біля кордону двох ліній рекурсивного алгоритму. Перебір слів при спробі заповнити довгі шаблони буде приводити до швидкої зміни областей пошуку рішень, а значить, збільшить ймовірність наштовхнутися хоча б на одне з них на самому початку повного перебору варіантів.

Третя оптимізація – метод гілок і меж. Якщо пошук значення для шаблону слова в рамках другої лінії не приносить результату, на попередньому

кроці заповнювався шаблон тієї ж лінії (тієї ж орієнтації), то немає ніякого сенсу повертатися на попередній крок рекурсії, якщо довжина попереднього шаблону більше (пам'ятаємо, що довжини шаблонів відсортовані за зменшенням). Сенсу в цьому немає, оскільки в даному випадку всі попередні рішення другої черги незалежні з рішенням для поточного шаблону слова. Має сенс повернутися до того шаблону першої черги, з яким є перетин. Якщо пересічний кілька, то вибираємо найближчий крок рекурсії – шаблон з найбільшою ординатою.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРАКТИВНИХ МЕТОДІВ НАВЧАННЯ

4.1 Реалізація мобільного додатку «Перевір себе»

При запуску програми (рис. 4.1) запускається головна активність, для користувача надається вибір, зареєструватися або ввести ім'я (для зареєстрованих).

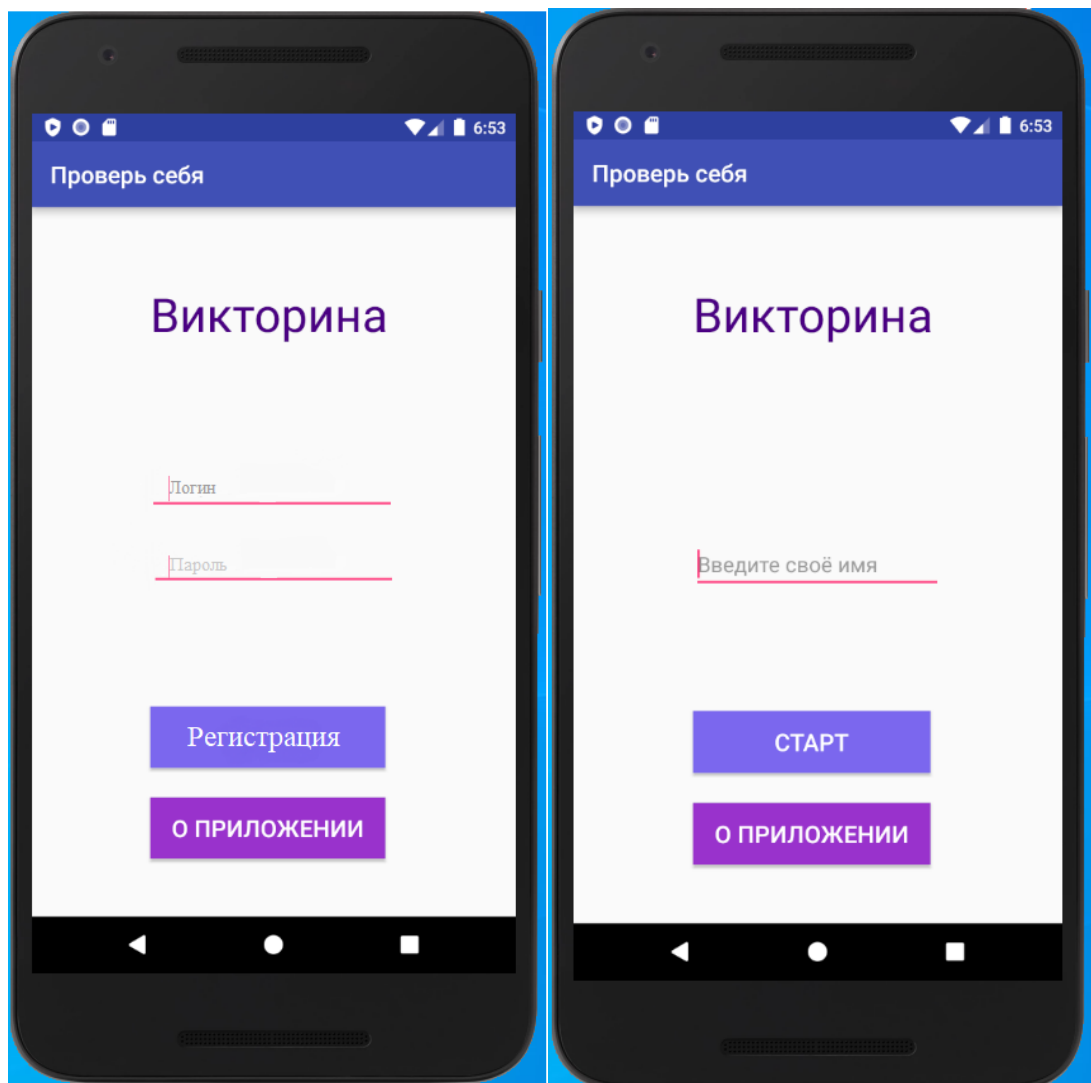


Рисунок 4.1 – Запуск додатку

Якщо додаток запускається у перший раз – Користувача пропонується зареєструватися (ввести логін, пароль і натиснути кнопку «Реєстрація»).

Якщо додаток вже запускався – достатньо ввести своє ім'я і натиснути кнопку «Старт».

Також на головній присутня кнопка «Про додаток». Якщо на неї натиснути - з'явиться вікно, в якому є інформація про розробника (рис. 4.2).

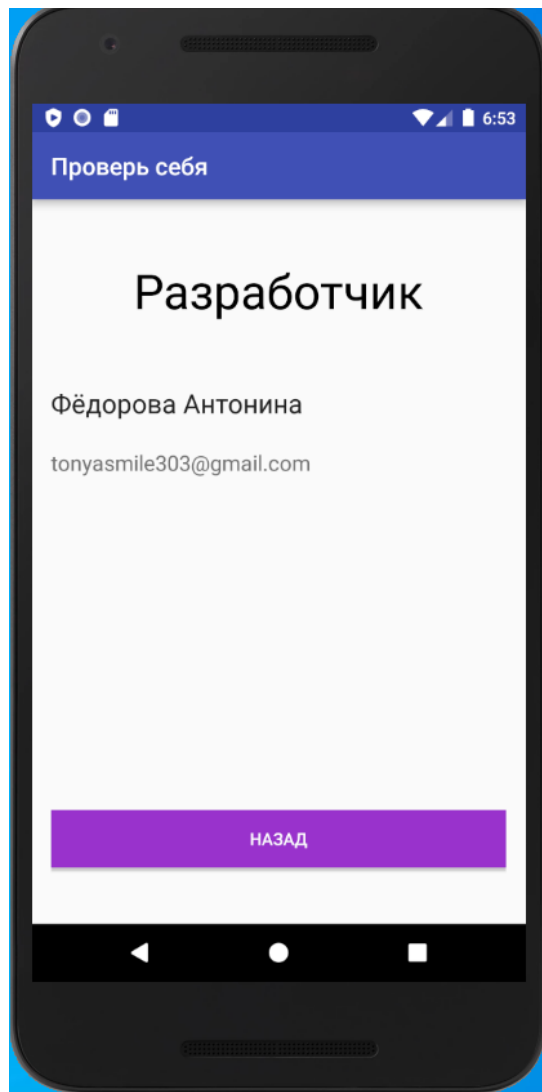


Рисунок 4.2 – Вікно «Про додаток»

Після того, як Користувач ввів свої данні – потрібно натиснути кнопку «Старт». Після цього – відкривається вікно вибору предмета, тестування з

якого Користувач буде проходити (рис.4.3). Предмети на вибір: програмування, нейронні мережі, ООП, Web – дизайн, моделювання.

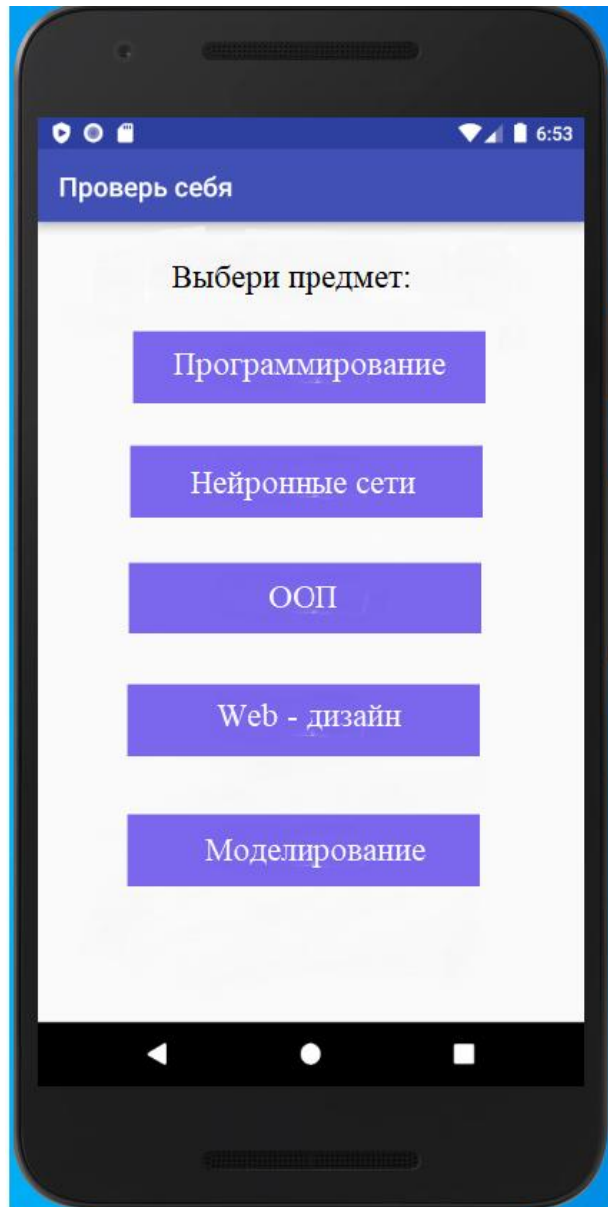


Рис.4.3 – Вікно вибору предмета «Вибери предмет»

Після того, як Користувач обрав предмет, з якого буде проходити тестування, починається тестування (рис. 4.4).

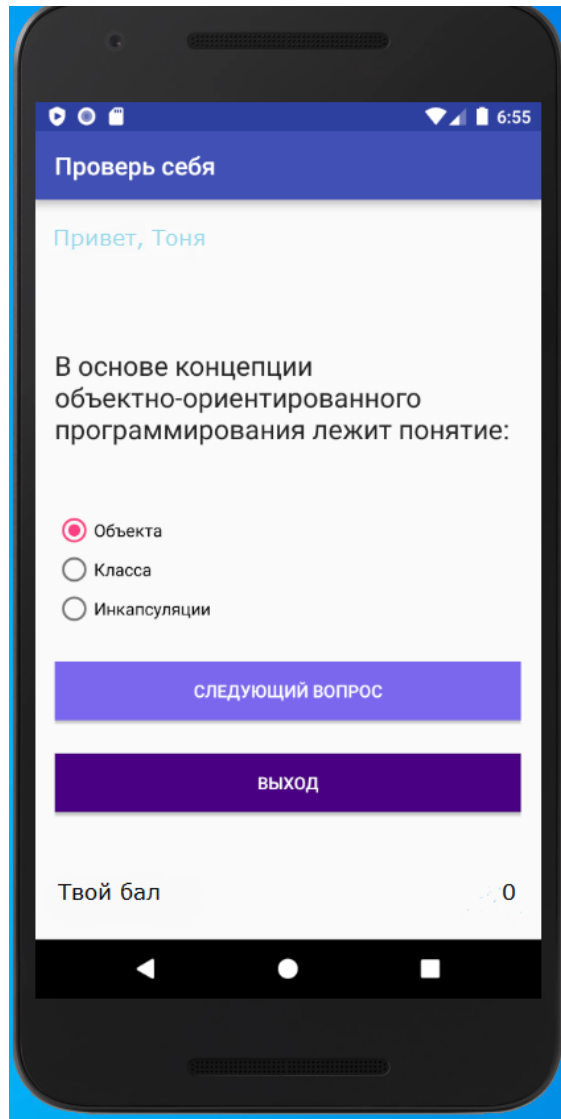


Рисунок 4.4 – Вікно тестування

На вікні тестування знаходяться такі елементи:

- вітання Користувача;
- питання;
- 3 варіанти відповідей;
- кнопка «Наступне питання», яке направляє користувача на наступну сторінку тестування;
- кнопка «Вихід» для того, щоб вийти з додатку;
- і знизу – кількість отриманих балів на момент тестування.

Після того, як Користувач пройшов тестування – з’являється вікно результатів (рис.4.5).

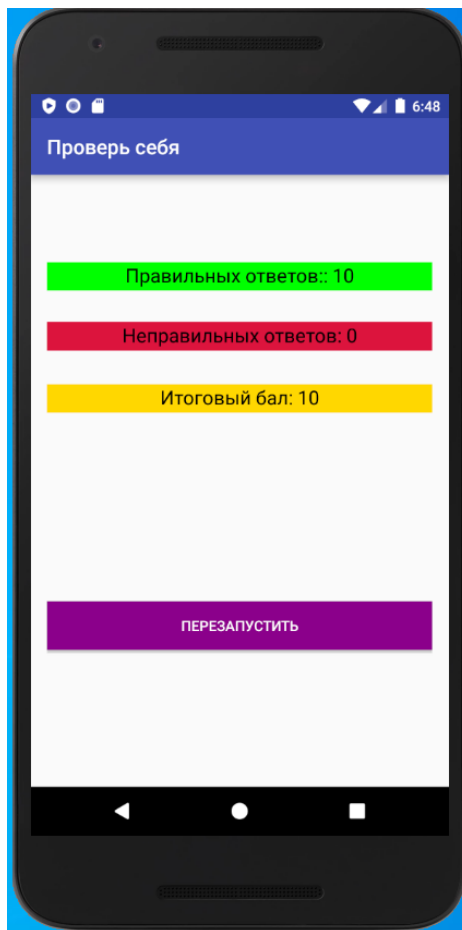


Рисунок 4.5 – Вікно результатів

Вікно результатів показує:

- кількість правильних відповідей;
- кількість неправильних відповідей;
- підсумковий бал;
- кнопка «Перезапустити».

4.2 Реалізація проекту «Кросворд»

4.2.1 Опис структури проекту

Класи і структури, що представляють модель кросворду, об'єднані в простір імен Model. Інші елементи знаходяться або в кореновому просторі імен проекту, або в службовому Properties.

Клас Grid. Шаблон (Сітка) кросворду. Цей клас представляє не що інше, як шаблон кросворду: прямокутний масив логічних. До завдань класу входить зовсім небагато: зміна розміру шаблону, редагування осередків (вставка осередки для символу або порожнього місця) і формування списку шаблонів слів, як по горизонталі, так і по вертикалі.

Метод buildWordList. В ньому конструктор класу StreamReader реалізує зчитування тексту з файлу. За допомогою методу String.Split створюється масив підстрок, розбиваючи вхідну строку по одному або декільком розділювачам.

Метод InitializeBoard. В ньому формується сітка кросворду. Метод ToUpper конвертує текст у верхній регістр. Метод formatCell формує білі ячейки. Метод board_CellPainting малює сітку і ячейки і заповнює їх кольорами.

Форма Form є головною формою і ініціалізує та відображає сітку кросворда. Форма Clues – форма питань та підказок для розв'язування кросворду.

Папка Puzzles зберігає файл, в якому знаходиться вся інформація про кросворд: як заповнювати: по горизонталі чи по вертикалі, саме питання, та відповідь

Інфраструктура додатку. До цієї групи належать додаткові компоненти програми, які виконують службові ролі. У просторі імен Properties знаходяться компоненти Resources (ресурси) і Settings (налаштування проекту). У кореновому просторі імен визначено компонент Program, який реалізує точку входу в додаток – статичну функцію Main.

4.2.2 Опис роботи програми

Після запуску програми на екрані з'являються 2 головних вікна (рис. 4.6, рис. 4.7).

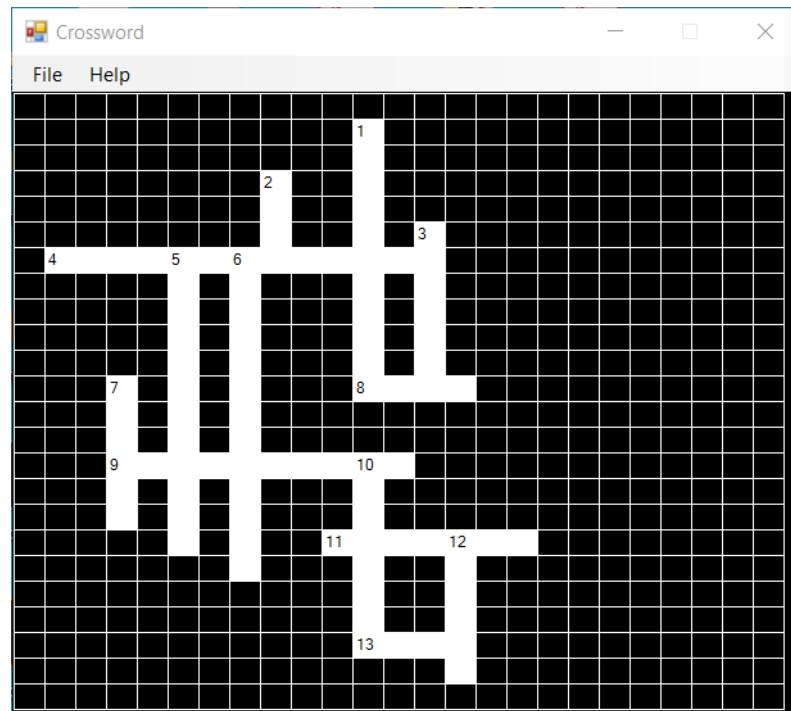
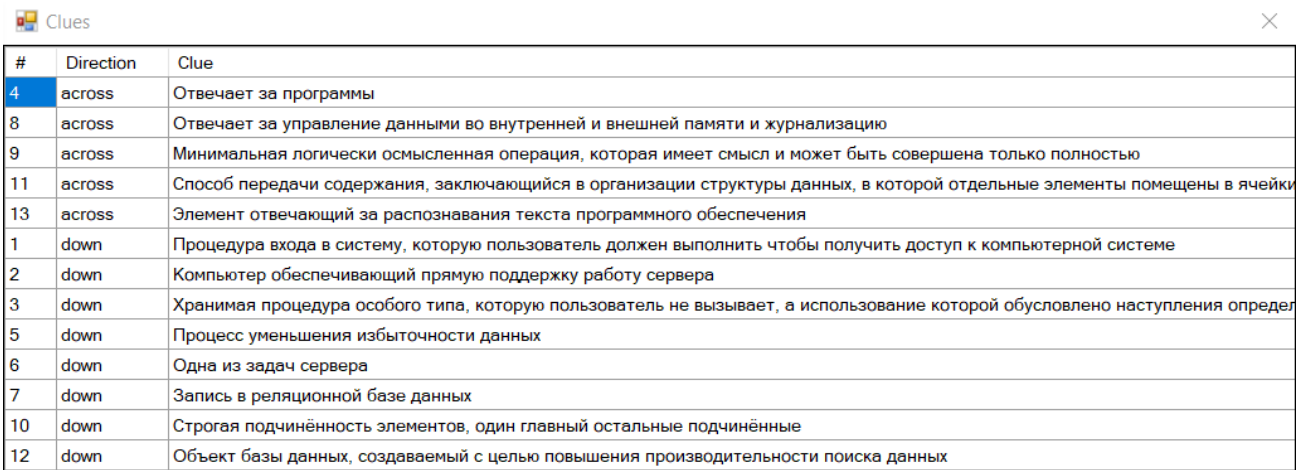


Рисунок 4.6 – Головне вікно програми

Перше представлене вікно (рис.4.6) – сітка кросворду для заповнення. Кожне слово, в залежності від позиції, пронумеровано. Якщо слово вірне – літери горять зеленим, інакше – червоним кольором.

Вкладка File – дозволяє відкрити любий файл кросворду, який є заздалегідь підготовленим, і також функція виходу.

Вкладка Help зберігає в собі інформацію про розробника і про сам продукт.



#	Direction	Clue
4	across	Отвечает за программы
8	across	Отвечает за управление данными во внутренней и внешней памяти и журнализацию
9	across	Минимальная логически осмысленная операция, которая имеет смысл и может быть совершена только полностью
11	across	Способ передачи содержания, заключающийся в организации структуры данных, в которой отдельные элементы помещены в ячейки
13	across	Элемент отвечающий за распознавания текста программного обеспечения
1	down	Процедура входа в систему, которую пользователь должен выполнить чтобы получить доступ к компьютерной системе
2	down	Компьютер обеспечивающий прямую поддержку работу сервера
3	down	Хранимая процедура особого типа, которую пользователь не вызывает, а использование которой обусловлено наступления определ
5	down	Процесс уменьшения избыточности данных
6	down	Одна из задач сервера
7	down	Запись в реляционной базе данных
10	down	Строгая подчинённость элементов, один главный остальные подчинённые
12	down	Объект базы данных, создаваемый с целью повышения производительности поиска данных

Рисунок 4.7 – Вікно питань та підказок

Друге представлене вікно (рис.4.7) – зображує питання кросворду, підказки до розташування слова (по горизонталі або по вертикалі).

На рис. 4.8 зображено вікно розгаданого кросворду. Про те, що кросворд розгадали правильно, свідчить колір літер. Вони є зеленого кольору.

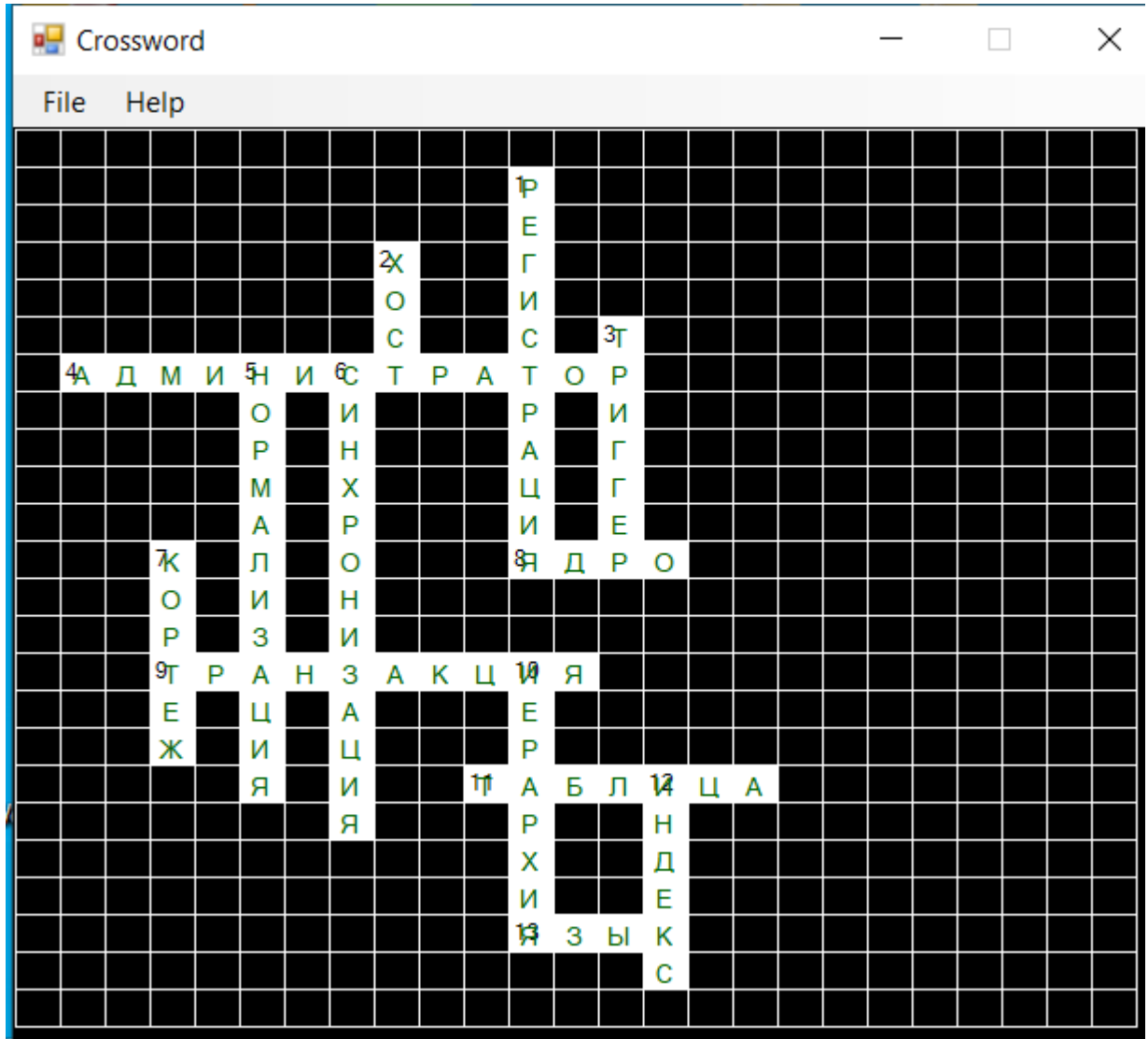


Рисунок 4.8 – Вікно розгаданого кросворду

ВИСНОВКИ

Результатом першої частини магістерської роботи є реалізований мобільний ігровий додаток для проведення швидких та зручних інтерактивних тестувань у предметній області. Розроблений додаток задовольняє всі вимоги, які були виявлені при аналізі предметної області та існуючих рішень.

Під час роботи над дипломним проектом були розв'язанні такі задачі:

- зроблено змістовний опис і аналіз предметної області, структурних і функціональних особливостей об'єкта управління: описана діяльність об'єкта управління, бізнес-процеси предметної області змодельовані у стандарті IDEF0;

- проведено огляд і аналіз існуючих систем.

- розроблені функціональні вимоги: побудовані діаграми варіантів використання;

- спроектована структура бази даних.

Функціональних можливостей цього додатку достатньо, для того, аби надати впливу на якість освітнього процесу, а саме:

- збільшити залученість учнів;

- покращити запам'ятовуваність нового матеріалу;

- зменшити навантаження на викладача.

Результатом другої частини є реалізований програмний продукт – кросворд.

Кросворд – головоломка, що представляє собою переплетення рядів клітинок, які заповнюються словами за заданим значенням.

Зазвичай значення слів задаються описово під цією фігурою, спочатку значення слів, які повинні вийти по горизонталі, потім – по вертикалі.

В ході виконання дипломної роботи було отримано повнофункціональне програмне забезпечення для генерації кросвордів, що працює у операційній системі Windows. Інтерфейс створеної програми зручний, простий, ві-

дображає її можливості

Розроблений додаток являє собою інструмент інтерактивного створення форм кросвордів і їх наповнення. Іншими словами – це генератор кросвордів. Подібні додатки можуть використовуватися різними виданнями, які або спеціалізуються на випуску кросвордів, або іноді прикрашають ними свою продукцію.

Розроблене програмне забезпечення задовольняє всім вимогам, поставленим на етапі постановки завдання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. DSpace JSPUI (загол. з екрану). URL : <http://repo.knmu.edu.ua/handle/123456789/8425>. (дата звернення – 01.12.2020).
2. 3.3Огляд інтерактивних методів (загол. з екрану). URL: lt.multycourse.com.ua/ua/page/19/67#2. (дата звернення – 01.12.2020).
- 3.Інтерактивні методи навчання (загол. з екрану). URL: <https://sites.google.com/site/nmcmuk/naukova-dialnist/interaktivni-metodi-navcanna>. (дата звернення – 01.12.2020).
4. І.М.Мельничук. Особливості застосування інтерактивних ігор у вищому навчальному закладі. УДК: 37.02:372. 12с.
5. Учебный кроссворд: как и где составить (загол. з екрану). URL : <https://www.eduneo.ru/uchebnyj-krossvord-kak-i-gde-sostavit/>. (дата звернення – 01.12.2020).
6. Класичні кросворди онлайн. (загол. з екрану). URL : <https://onlinetestpad.com/ua/crosswords/classic>. (дата звернення – 03.12.2020).
7. Онлайн викторина «Эрудит». (загол. з екрану). URL: <https://source2016.ru/event/>. (дата звернення – 03.12.2020).
8. Всемирная история викторина (загол. з екрану). URL : <https://play.google.com/store/apps/details?id=com.kalinovskiy.historyquiz>. (дата звернення – 03.12.2020).
9. Кроссворды на русском (загол. з екрану). URL : https://play.google.com/store/apps/details?id=com.appspot.orium_blog_crossword. (дата звернення – 03.12.2020).

10. Список версій Android (загол. з екрану). URL : https://uk.wiipdia.org/wiki/Список_версій_Android. (дата звернення – 07.12.2020).
11. Архитектура Android (загол. з екрану). URL : https://studbooks.net/2052989/informatika/arhitektura_android. (дата звернення – 08.12.2020).
12. Android Developers Blog (загол. з екрану). URL : <https://androiddevelopers.googleblog.com/2020/05/android-studio-4.html>. (дата звернення – 08.12.2020)
13. Документация по Visual Studio (загол. з екрану). URL : <https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2019&preserve-view=true>. (дата звернення – 08.12.2020).
14. Методология IDEF0. (загол. з екрану). URL : https://studme.org/87184/ekonomika/metoologiya_idef0/. (дата звернення – 10.12.2020).
15. Представление данных с помощью модели «сущность-связь. (загол. з екрану). URL : http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_1.html. (дата звернення – 09.10.2020)
16. Представление данных с помощью модели «сущность-связь. (загол. з екрану). URL: http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_1.html. (дата звернення – 06.12.2020).