

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-71 Сич Євгеній Владиславович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Марченко А. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
 Факультет електроніки та інформаційних технологій
 Кафедра комп'ютерних наук
 Секція інформаційних технологій проектування
 Спеціальність 122 «Комп'ютерні науки»
 Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
 «__» _____ 2021 р.

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

Сич Євгеній Владиславович

1 Тема роботи Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ

керівник роботи Марченко Анна Вікторівна, к.т.н., доцент,

затверджені наказом по університету від «14» квітня 2021 р. №0181-VI

2 Строк подання студентом роботи «7» червня 2021 р.

3 Вхідні дані до роботи технічне завдання на розробку інформаційної системи «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ»

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз програмних продуктів – аналогів, моделювання «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ», моделювання «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ»

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

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____ 15.02.2021 _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка цілей	18.02.2021 – 04.03.2021	
2	Складання технічного завдання	05.03.2021 – 10.03.2021	
3	Підготовка прототипу	11.03.2021 – 14.03.2021	
4	Створення макету дизайну мобільного додатку	15.03.2021 – 24.03.2021	
5	Розробка	25.03.2021 – 26.04.2021	
6	Робота над адаптивністю дизайну	27.04.2021 – 01.05.2021	
7	Тестування	01.05.2021 – 05.05.2021	
8	Підготовка до випуску	06.05.2021 – 13.05.2021	
9	Завантаження додатку у магазин “Google Play”	14.05.2021 – 18.05.2021	

Студент _____

(підпис)

Сич Є.В.

Керівник роботи _____

(підпис)

к.т.н., доц. Марченко А.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 11 найменувань, додатків. Загальний обсяг роботи – 133 сторінок, у тому числі 50 сторінок основного тексту, 2 сторінки списку використаних джерел, 83 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці мобільного ігрового додатку для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ.

В першому розділі наведено огляд останніх досліджень за темою роботи та проведено аналіз програмних продуктів - аналогів.

В другому розділі проведено моделювання «Мобільного ігрового додатку для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ», змодельовано контекстну діаграму, IDEF0 та діаграму варіантів використання. Проведено проектування моделі інформаційної бази.

В третьому розділі детально описано етапи практичної реалізації проекту та наведено приклади використання програмного додатку, що демонструють його працездатність.

Ключові слова: Android, квест, бухгалтер, гра, мобільний додаток, SQLite, Room, Navigation component.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Огляд останніх досліджень і публікацій	7
1.2 Аналіз програмних продуктів - аналогів.....	9
1.3 Постановка задачі	10
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ	11
2.1 Моделювання «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ»	11
2.1.1 Діаграми нотації IDEF0	11
2.1.2 Діаграма Use Case.....	14
2.2 Проектування моделі інформаційної бази	15
3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ.....	16
3.1 Розробка візуального інтерфейсу додатку.....	16
3.2 Програмна реалізація	20
3.3 Використання додатку	25
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	36
ДОДАТОК А.....	38
ДОДАТОК Б.....	49
ДОДАТОК В.....	61
ДОДАТОК Г.....	62
ДОДАТОК Д.....	122

ВСТУП

Актуальність: На даний момент складно уявити життя людей без спеціальних пристроїв, механізмів та технологій, використання яких спрощує та підвищує якість життя. Вже протягом більше ніж десяти років мобільні додатки займають далеко не останнє місце у списку речей, які значно допомагають у повсякденному житті. В наші дні смартфон є одним з найзручніших пристроїв для навчання, розваг, покупок. Саме через все вищесказане можна вважати, що мобільні додатки є достатньо актуальними.

На даний час існує достатньо велика конкуренція між підприємствами та установами майже у всіх сферах послуг. Однією з таких сфер є освіта, де успіх перш за все залежить від кількості залучених студентів. Відомо, що найбільший відсоток студентів в Україні це особи від 17 до 23 років, саме ця цільова група проводить найбільше часу у мережі за допомогою смартфонів. Тобто можна передбачити, що для донесення інформації до більшої кількості абітурієнтів буде доцільно використовувати мобільний додаток.

Мета роботи: розробка мобільного ігрового додатку для потенційних абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ на базі операційної системи Android.

Призначення додатку полягає у донесенні інформації про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ у ігровому форматі за допомогою мобільного додатку.

Гра буде складатися з 8-ми рівнів на яких будуть розташовані завдання, переможцем стає той учасник, що швидше за всіх подолає вісім рівнів.

Використання даного додатку допоможе більш обширно та цікаво доносити інформацію до потенційних абітурієнтів про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ.

За результатами проведеної роботи опубліковано тези доповіді на конференції «ІМА 2021» [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Все більше зростає конкуренція між навчальними закладами та їх підрозділами, щодо залучення нових студентів. Задля того, щоб абітурієнт обрав навчальний заклад та спеціальність, необхідно дати йому якомога більше інформації про напрямок і правильно донести її. На рахунок цієї теми проводилося немало досліджень.

У дослідженні «Вибір вищої освіти після школи» з'ясували як українські абітурієнти обирають вищий навчальний заклад, ставлення до вищої освіти, причини вибору спеціальності та, зокрема, якою інформацією користуються абітурієнти при виборі та від чого залежало їх рішення [2]. У розділі «Джерела інформації про місце навчання» були представлені результати оцінки джерел інформації про університети та спеціальності (рис. 1.1).

Якими джерелами ви користувалися для отримання інформації про ЗВО та спеціальності, до яких ви подавали заяви?

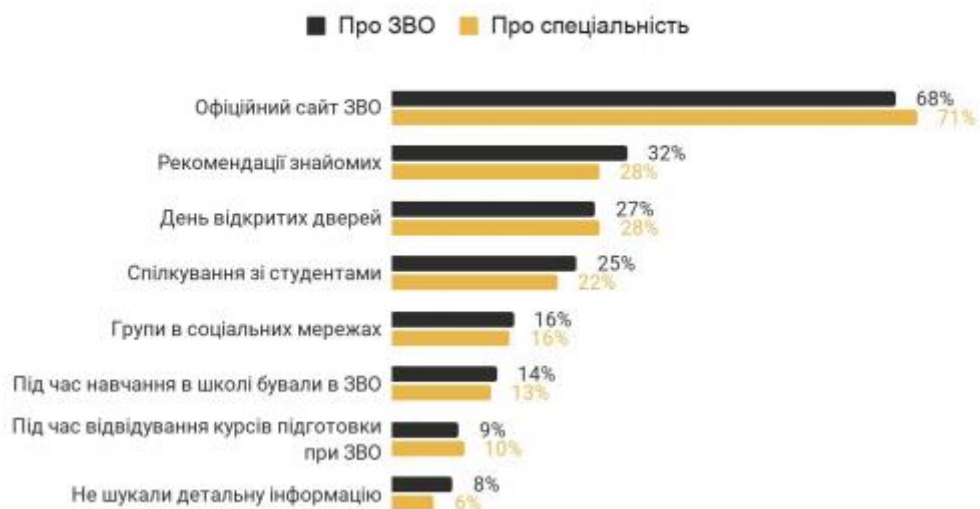


Рисунок 1.1 – Результати оцінки джерел інформації про місце навчання

Отже, найбільш популярним джерелом є офіційні сайти, отже і їх мобільні версії, оскільки згідно статистики зі статті Mobile Tech Trends 2021: Era of Immersive Technologies [10] користувачів смартфонів зростає з кожним роком (рис. 1.2).

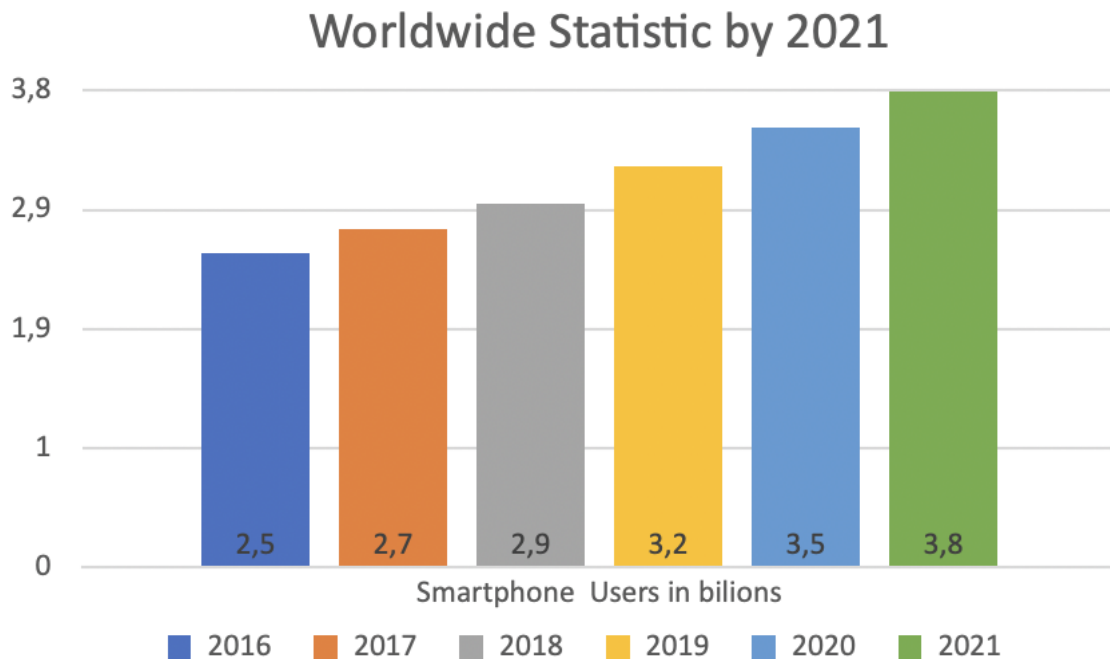


Рисунок 1.2 – Статистика зростання користувачів смартфонів

Приблизно на одному рівні у рейтингу, розташовані: рекомендації знайомих, день відкритих дверей та спілкування зі студентами. Звичайно, абітурієнти звертаються також до груп в соціальних мережах. Також у дослідженні зазначена думка про недоцільність розділення джерел отримання інформації на більш або менш корисні, оскільки різні джерела надають інформацію різного роду.

У статті «Розробка мобільних додатків: аналіз світового ринку» зазначено про зростання ринку мобільних додатків, збільшується: кількість світових завантажень; час, проведений користувачами в додатках [3].

Отже, як одне із джерел інформації про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ та для реклами інших (сайт, соціальні мережі, відеоролик) було прийнято рішення про розробку мобільного ігрового додатку для абітурієнтів кафедри.

1.2 Аналіз програмних продуктів - аналогів

На даний час було знайдено декілька аналогів додатку в магазині додатків Play Store та виділено декілька додатків із схожими рисами. Загалом ці додатки представляють собою опитування та тести в напрямку бухгалтерії. Найбільш схожими виявилися такі додатки: Accounting Quiz – AccQuiz, Accounting Quiz, Accounting Quiz Game.

Додаток Accounting Quiz – AccQuiz це безкоштовна вікторина для бухгалтерії, яка включає флеш-картки для вивчення вікторини. Вона охоплює різні теми фінансового обліку. Додаток підходить для початківців, які хочуть побачити бухгалтерський облік. Є можливість вивчити основи з 7-ми розділів:

- основи бухгалтерського обліку та бізнесу
- аналіз транзакцій
- процес регулювання
- цикл обліку
- товарний бухгалтерський облік
- облік запасів
- внутрішній контроль та готівка

Інший додаток-аналог Accounting Quiz - включає в себе безліч питань для тестування та практики бухгалтерської підготовки. Також є можливість зберегти запитання, класифікувавши їх як улюблені, і переглядати їх пізніше. Також в наявності глава з питаннями, які потребують часу на роздуми, а також окремі для практики.

Додаток Accounting Quiz Game є найбільш схожим аналогом з розробленим додатком. Додаток призначений для навчання або відточування своїх професійних знань. Майже 700 бухгалтерських запитань у форматі PDF, що містять запитання, відповіді та пояснення. Є можливість вибрати такі теми: Загальні принципи, типи рахунків, баланс, звіт про прибутки та збитки, дебіти та кредити, суб'єкти

господарювання США, звіт про власний капітал, звіт про рух грошових коштів, методи амортизації, облігації, бухгалтерія, коефіцієнти.

Дані додатки є досить ефективними для вивчення бухгалтерського обліку, але всі приведені вище додатки не адаптовані під українську бухгалтерську та податкову системи [4]. Також відрізняється концепція даних додатків від розроблюваного, адже їх метою є вивчення або покращення навиків бухгалтерії.

1.3 Постановка задачі

Мета роботи - розробити мобільний додаток-гру профорієнтаційного призначення під операційну систему Android для абітурієнтів кафедри БОО ННІ БіЕМ СумДУ.

Гра має складатися з 8-ми рівнів з завданнями (надаються замовником) пов'язаними з бухгалтерським обліком у вигляді квесту.

Щоб реалізувати додаток необхідно використовувати мову програмування за допомогою якої можна працювати з Android software development kit . Для вибору доступні дві мови програмування – Kotlin та Java, вибір було зроблено саме на Kotlin, оскільки ця мова є більш адаптованою під розробку Android додатків та рекомендується до розробки компанією Google [5].

Середовищем розробки було обрано Android Studio так, як дана IDE має значно ширший набір інструментів для створення мобільних додатків ніж аналоги такі як Eclipse або IntelliJ IDEA.

Гра починається з сторінки привітання з кнопкою – «Старт». Потім учасник квесту має пройти вісім рівнів з фіксуванням часу. Після проходження гри фіксується час за який були виконано всі завдання, учасник має зробити скріншот екрану з результатом та надіслати у соцмережі кафедри БОО ННІ БіЕМ СумДУ

Повне технічне завдання на розробку наведено в додатку А.

Планування робіт представлено у додатку Б.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Моделювання «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ»

Після того, як було проведено аналіз предметної області, визначено системні потреби та актуальність, можемо перейти до моделювання та проектування Android додатку. На даному етапі було розроблено діаграму нотації IDEF0 [6], в якій було показано процеси реалізації проекту, їх послідовність та діаграми Use-Case, де демонструються всі функції, які може використовувати користувач.

2.1.1 Діаграми нотації IDEF0

Нотація IDEF0 показує бізнес-логіку. Для простішого представлення бізнес-логіки прямокутник являє собою проект. В який по периметру, із усіх сторін направлено стрілки, які означають типи даних:

- ліві стрілки – вхідні дані для роботи системи;
- праві стрілки – це вихідні дані;
- верхні стрілки – це дані керування;
- нижні стрілки – це дані механізму.

Процес «Проходження гри»:

вхідні дані – відповідь, завдання.

вихідні дані – час проходження.

управління – кількість життів, завдання, варіанти відповіді.

механізми – СУБД SQLite, ОС Android.

На рис. 2.2 відображено контекстну діаграму процесу проходження гри.

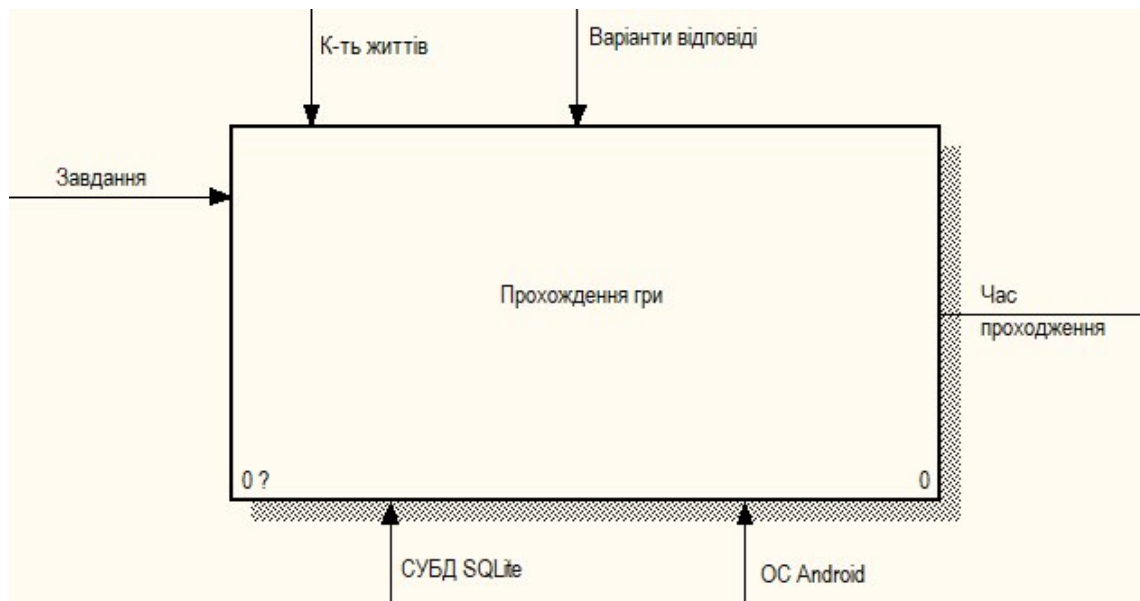


Рисунок 2.2 – Контекстна діаграма IDEF0

Наступний крок є декомпозиція [7] проходження гри, яку можна поділити на шість головних процесів:

- отримання завдання;
- введення відповіді;
- перевірка відповіді;
- підказка
- втрата життя
- перехід до наступного рівня
- завершення гри.

За результатами декомпозиції, розроблено діаграму 1-го рівня. Дані для діаграми:

- вхідні дані – відповідь, завдання.
 - вихідні дані – час проходження.
 - управління – кількість життів, завдання, варіанти відповіді.
- механізми – СУБД SQLite, ОС Android.

На рис. 2.3 відображено діаграму першого рівня.

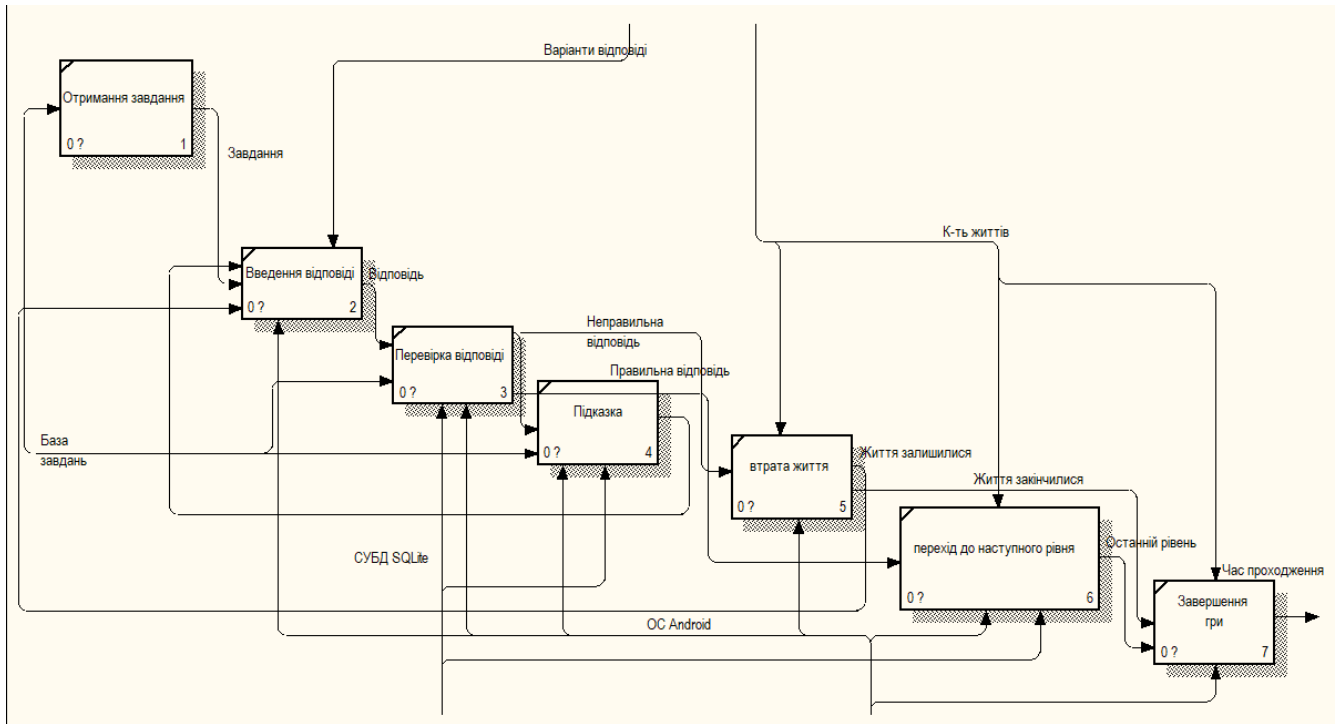


Рисунок 2.3 – Діаграма першого рівня

2.1.2 Діаграма Use Case

Щоб наглядно продемонструвати взаємодію користувача з системою можна скористуватися діаграмою Use Case [8]. Для побудови діаграми визначено двох акторів: базу даних та користувача. Також було визначено такі сценарії:

- переглянути контактні дані;
- почати гру;
- виконати всі завдання;
- зробити скріншот результату гри.

Use Case діаграму розроблену за вищевказаними сценаріями відображено на рис. 2.1.

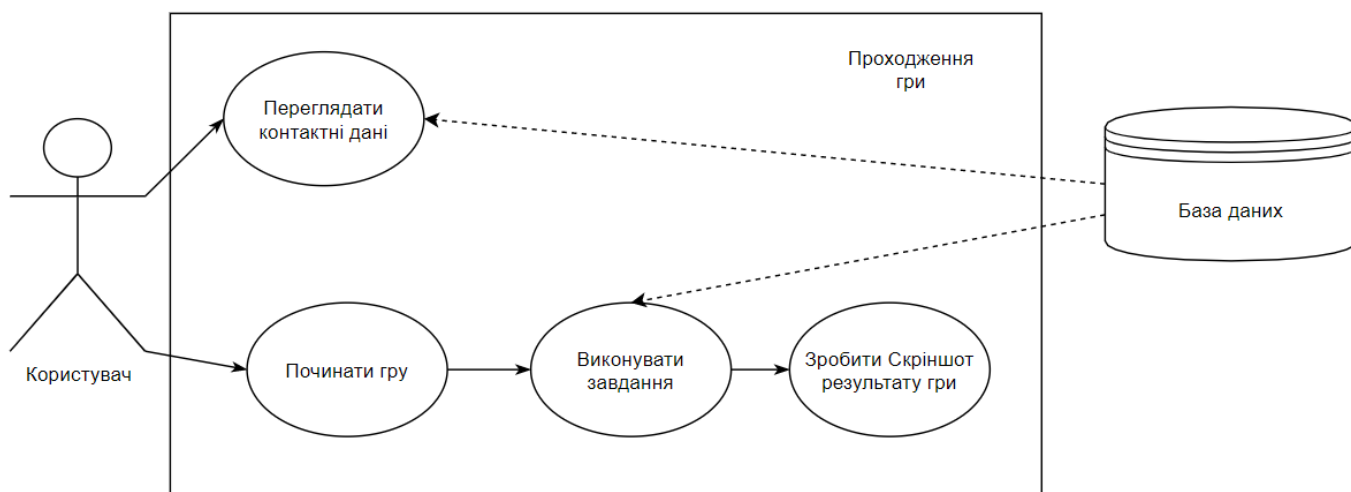


Рисунок 2.1 – Діаграма Use Case

2.2 Проектування моделі інформаційної бази

Будь-який продукт, який містить великі об'єми інформації для зберігання потребує правильно проєктованої бази даних. У даному додатку для зберігання маємо такі дані, як завдання для кожного рівня, які будуть зберігатися в репозиторії проєкту. У зв'язку з вимогами замовника додаток має бути незалежним від мережі Інтернет, отже було прийнято рішення зберігати дані локально у вигляді Kotlin файлу.

Предметна область бази даних: об'єкти – завдання та факт.

- атрибути об'єкта завдання: номер рівня, текст завдання, правильна відповідь, опис завдання.
- атрибути об'єкта факт: номер рівня, текст факту, вірність факту, належність до групи фактів.

На рисунку 2.4 показано ER-діаграму інформаційної бази.

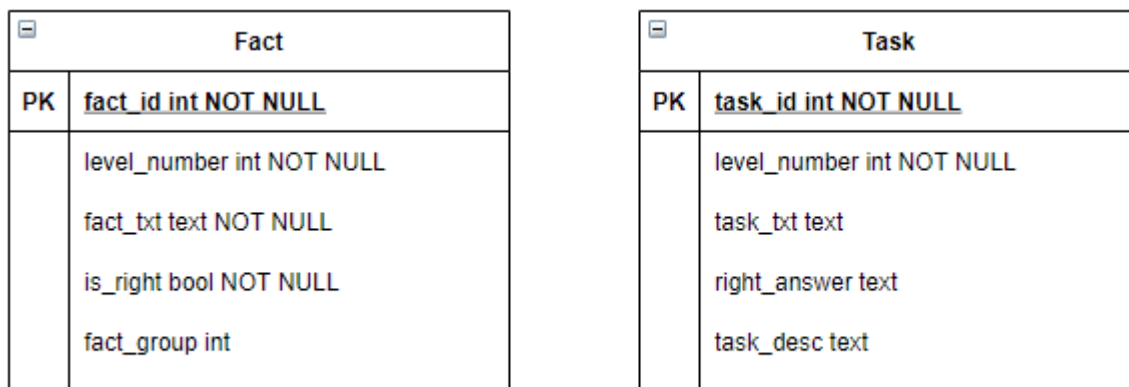


Рисунок 2.4 – ER-діаграма інформаційної бази

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

3.1 Розробка візуального інтерфейсу додатку

Інтерфейс мобільного додатку реалізуємо у Figma (рис. 3.1). Додатку являє собою набір екранів з інформацією, екранів з рівнями гри та допоміжних діалогових вікон.

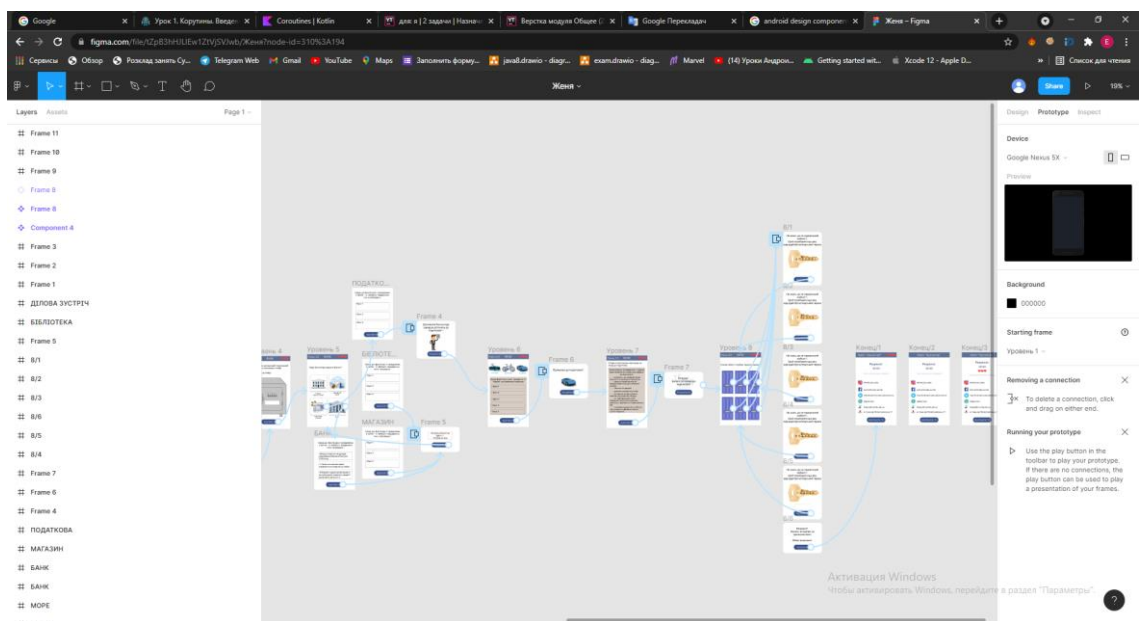


Рисунок 3.1 – Прототип додатку у Figma

При запуску додатку буде відображатися екран з кнопкою старту, після натискання на яку з'являється екран з правилами (рис. 3.2).

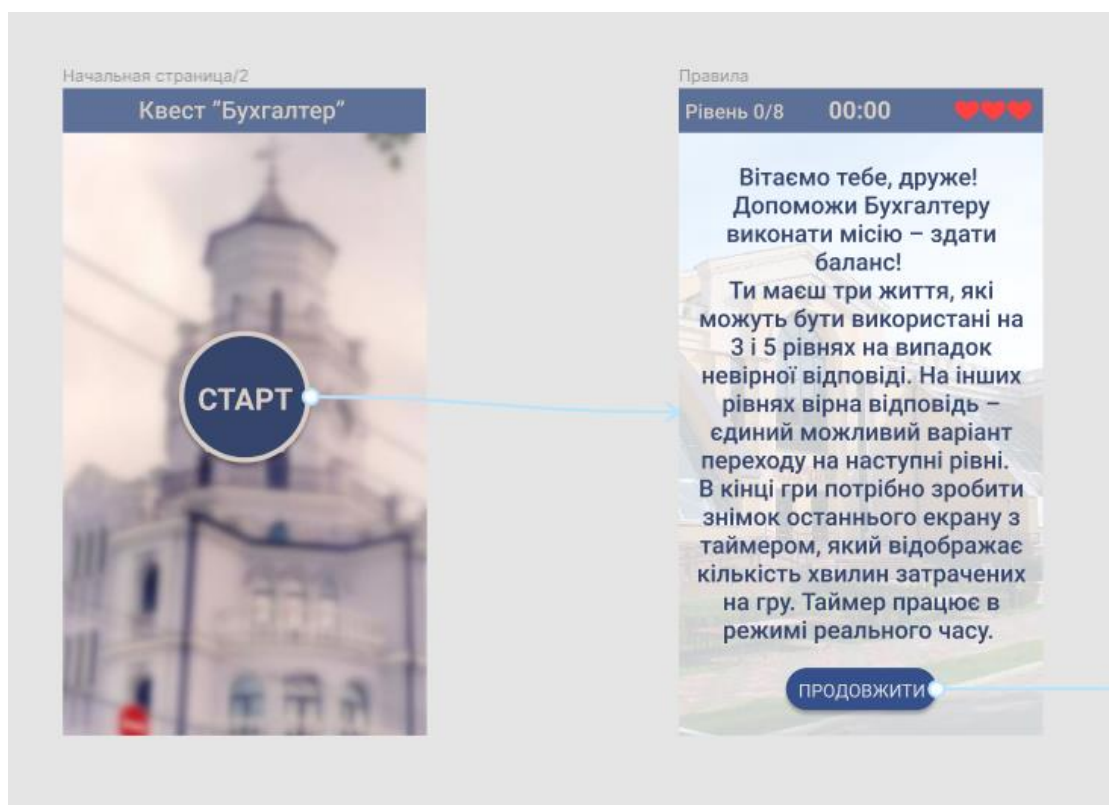


Рисунок 3.2 – Початкові екрани

Екрани першого та другого рівнів аналогічні по розташуванню елементів(рис. 3.3).

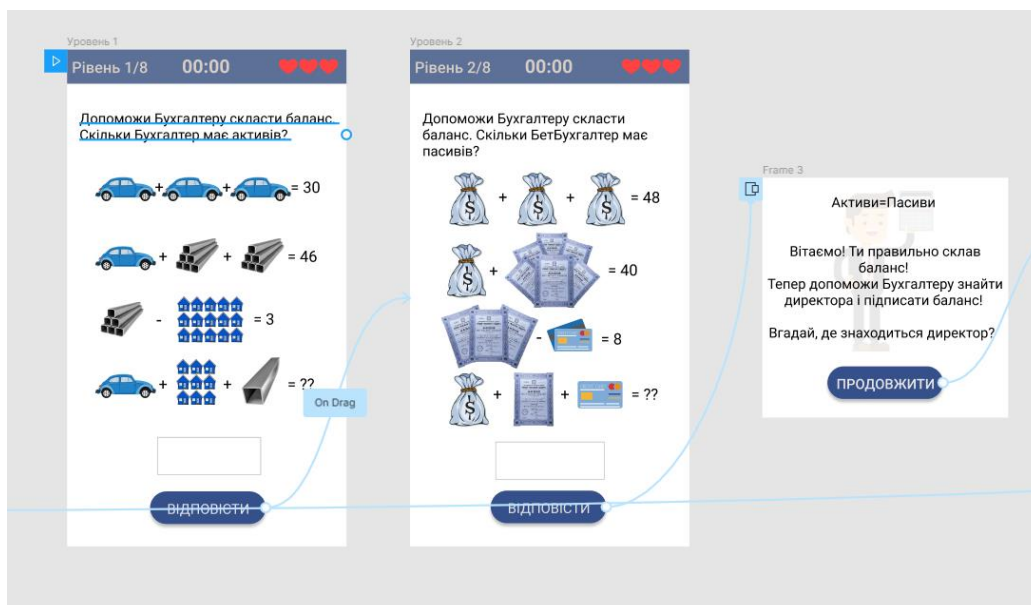


Рисунок 3.3 – Перший та другий рівні з сюжетним діалогом

Третій (рис. 3.4) та п'ятий (рис. 3.5). рівні мають екран з чотирма пунктами по натисканню на які з'являються діалогові вікна з завданнями.

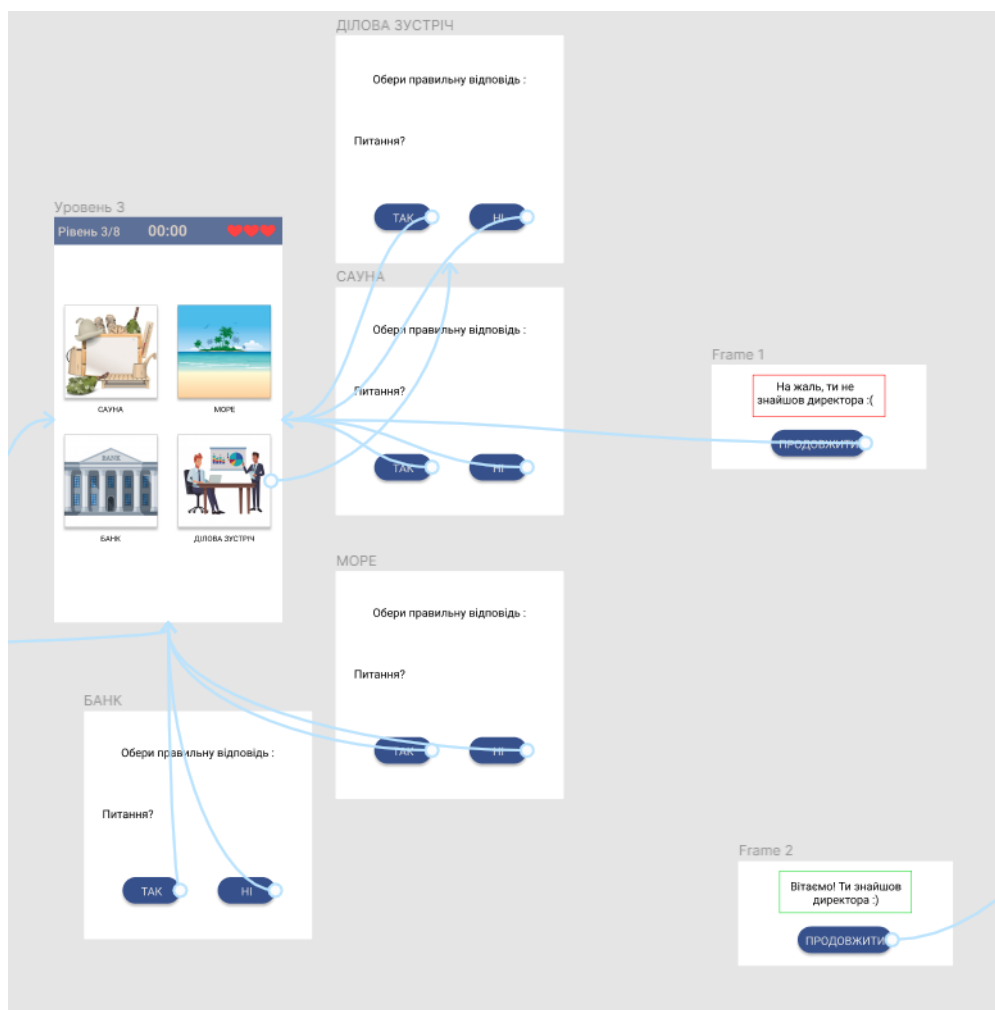


Рисунок 3.4 – Третій рівень з діалогами

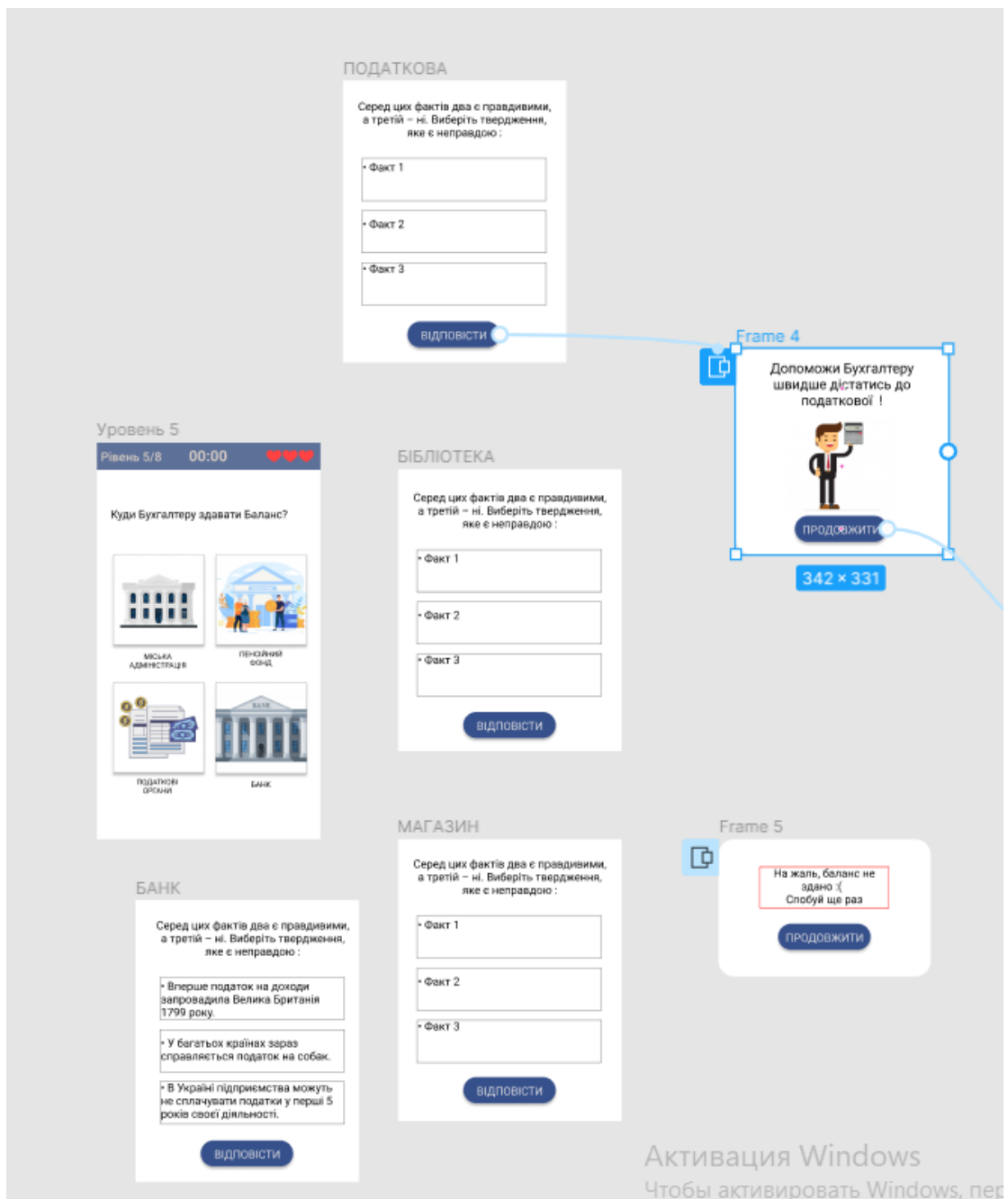


Рисунок 3.5 – Третій рівень з діалогами

На екрані результатів показано результат гри та контакти (рис. 3.6).

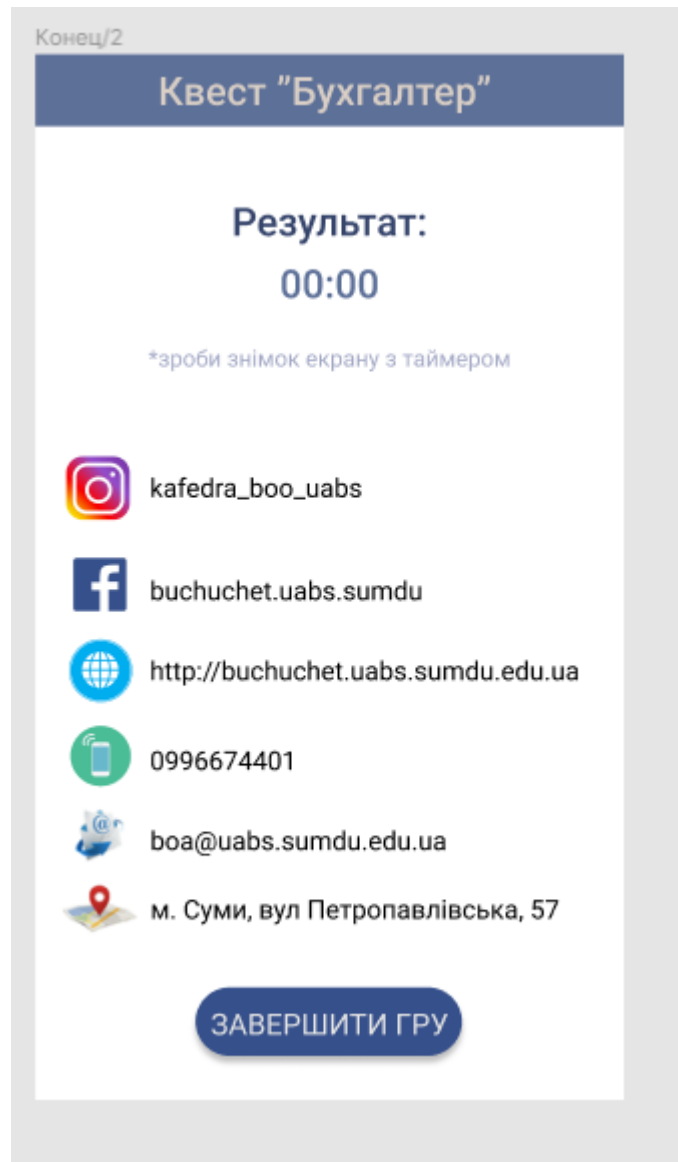


Рисунок 3.6 – Дизайн екрану «Інформація»

3.2 Програмна реалізація

Програмна реалізація додатку виконується мовою програмування Kotlin в IDE Android Studio. Для зберігання даних використовується СУБД SQLite з додатковим шаром абстракції – Room (рис 3.8-3.11).

```

1 package com.betaccountant.db.model
2
3 import ...
4
5
6 @Entity
7 data class Fact(
8     @PrimaryKey(autoGenerate = true)
9     val factId: Int? = null,
10    val levelNumber: Int,
11    val text: String,
12    val isTrue: Boolean,
13    val groupId: Int = 0
14 )

```

Рисунок 3.8 – Клас опису таблиці Fact

```

1 package com.betaccountant.db.model
2
3 import ...
4
5
6 @Entity
7 data class Image(
8     @PrimaryKey(autoGenerate = true)
9     val imageId: Int,
10    @ForeignKey(entity = Task::class, parentColumns = ["levelNumber"], childColumns = ["levelNumber"])
11    val taskId: Int,
12    val url: String,
13 )
14
15

```

Рисунок 3.9 – Клас опису таблиці Image

```

1 package com.betaccountant.db.model
2
3 import ...
4
5
6 @Entity
7 data class Story(
8     @PrimaryKey(autoGenerate = true)
9     val storyId: Int,
10    @ForeignKey(entity = Task::class, parentColumns = ["levelNumber"], childColumns = ["levelNumber"])
11    val taskId: Int,
12    val storyTxt: String,
13    val rightAnswer: String?,
14 )
15
16

```

Рисунок 3.10 – Клас опису таблиці Story

```

Refactor Build Run Tools VCS Window Help BetAccountant - Task.kt [BetAccountant]
com > betaccountant > db > model > Task
Fact.kt Image.kt Story.kt Task.kt
1 package com.betaccountant.db.model
2
3 import ...
4
5
6 @Entity
7 data class Task(
8     @PrimaryKey(autoGenerate = true)
9     val taskId: Int = 0,
10    val levelNumber: Int,
11    val question: String?,
12    val rightAnswer: String? = null,
13    val taskDesc: String? = null,
14    val isAddTask: Boolean? = false
15 )
16

```

Рисунок 3.11 – Клас опису таблиці Task

Для реалізації навігації у додатку використовується Navigation Component (рис 3.12-3.13).

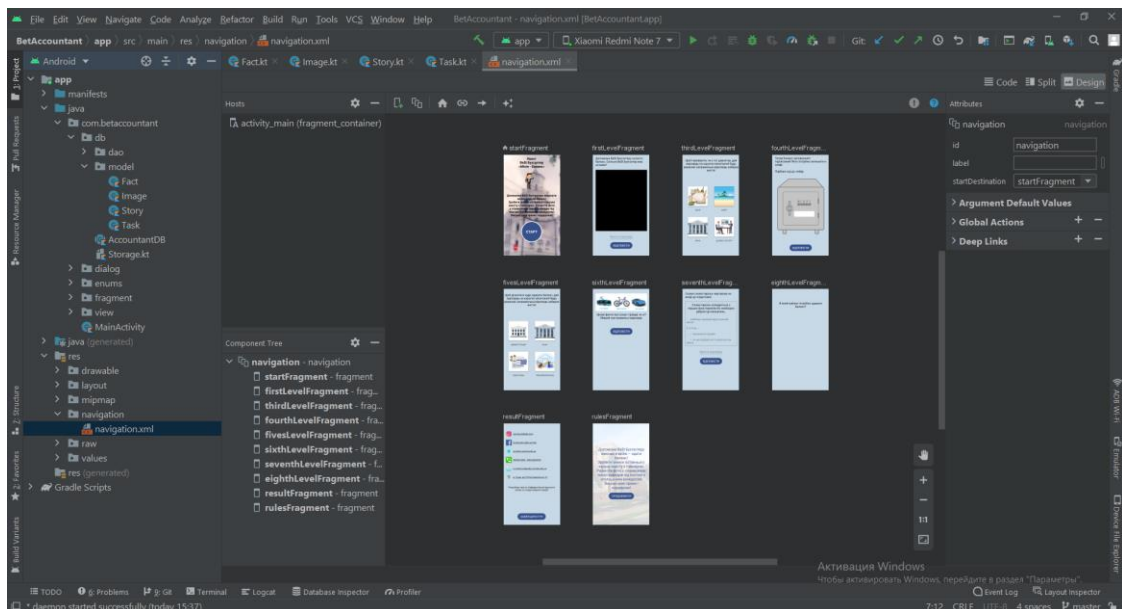


Рисунок 3.12 – Візуальний граф навігації

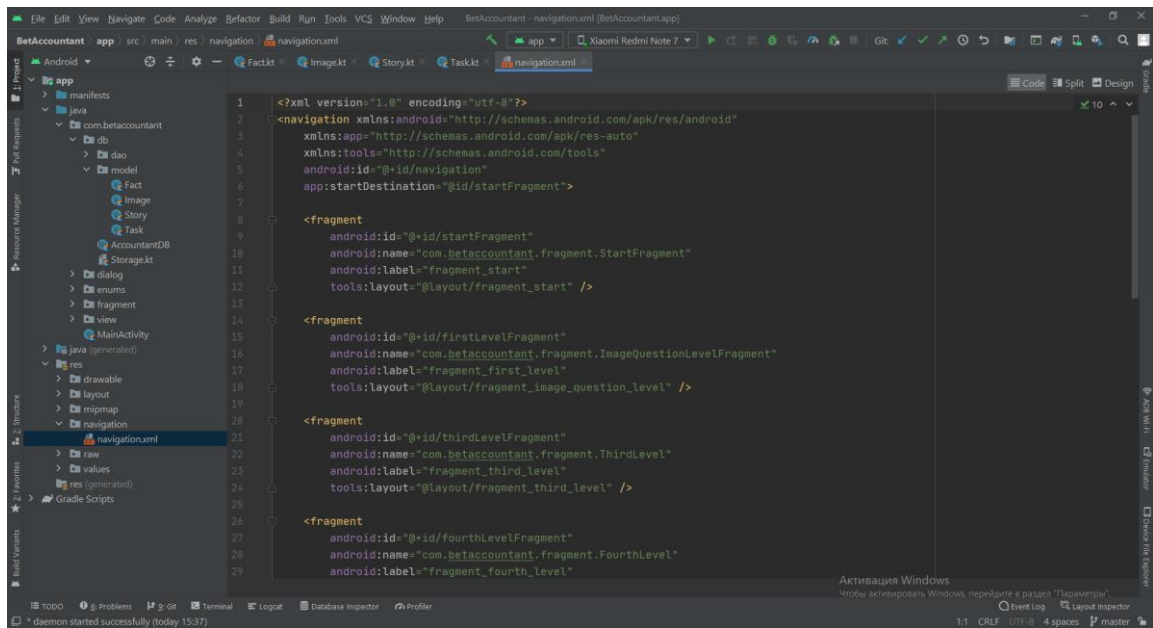


Рисунок 3.13 – Граф навігації у вигляді розмітки

Розробка інтерфейсу додатку відбувається за допомогою мови розмітки XML та інструментів Android Studio (рис. 3.14).

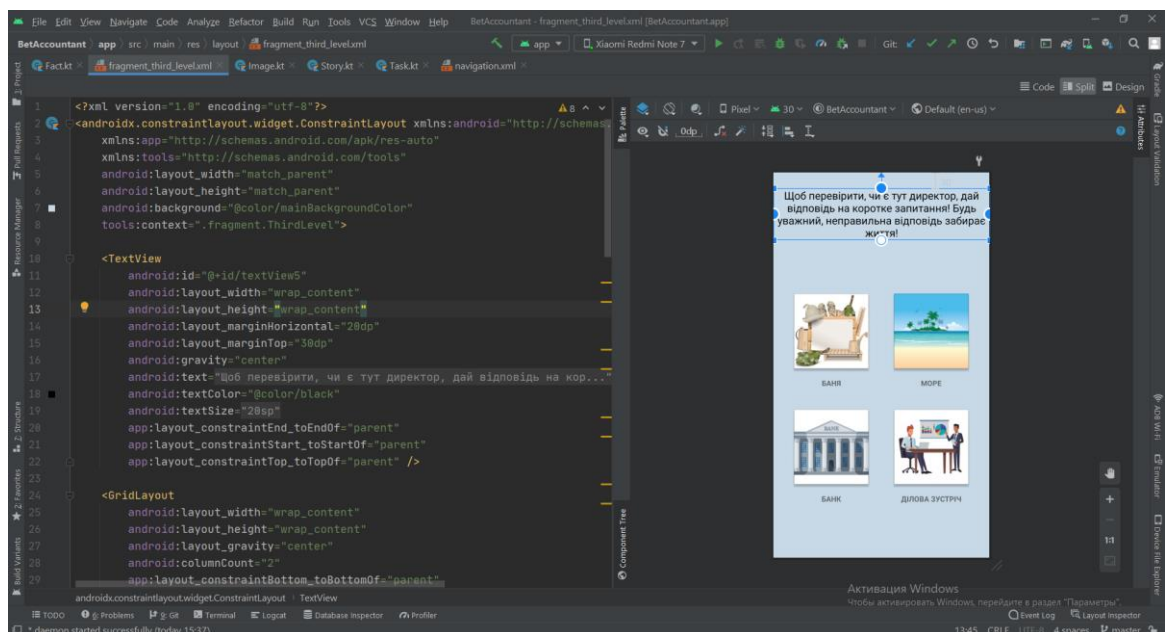


Рисунок 3.14 – Розмітка екрану 3-го рівня

Для контролю версій використовуємо систему контролю версій Git та платформу Github (рис. 3.15-3.16).

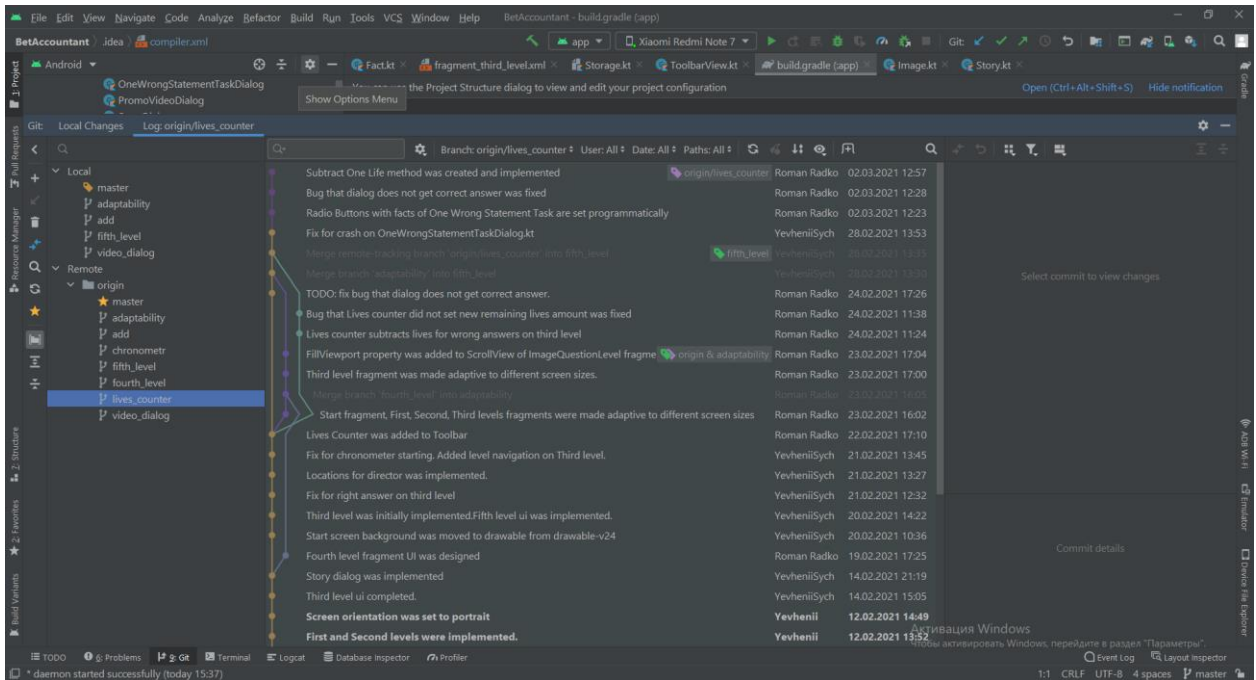


Рисунок 3.15 – Локальне дерево комітів

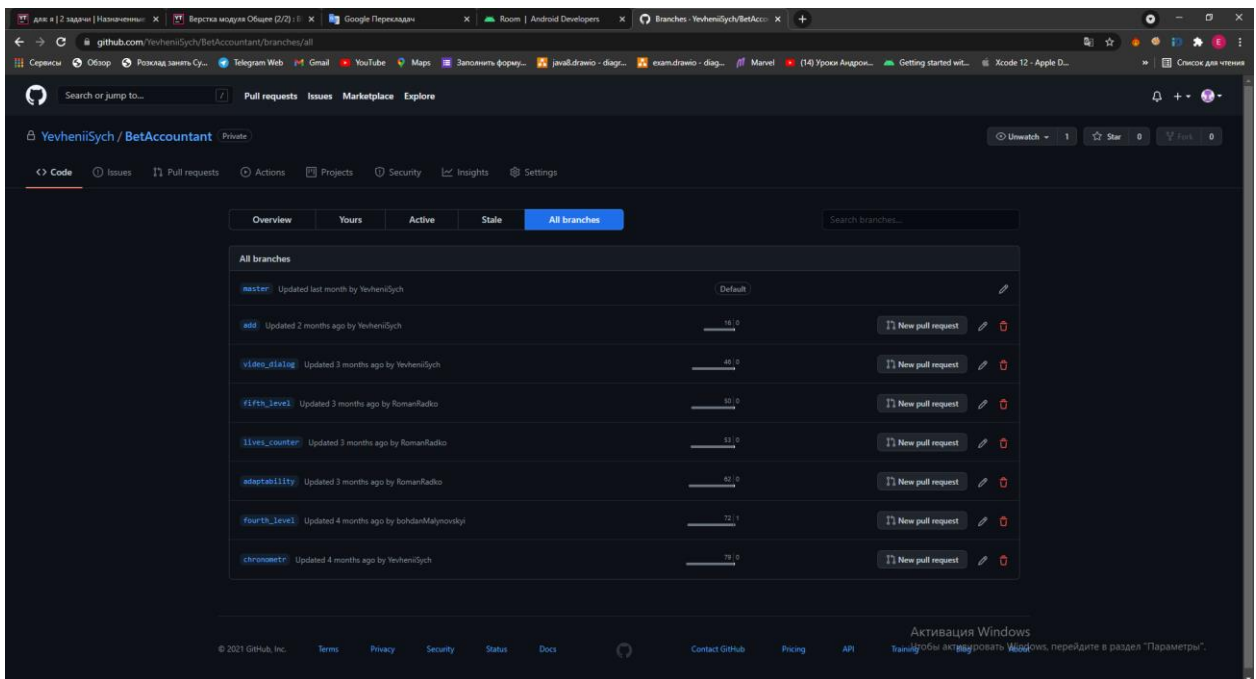


Рисунок 3.16 – Гілки розробки на віддаленому репозиторії

3.3 Використання додатку

Початковий екран додатку «Best Бухгалтер» містить загальну інформацію про гру та кнопку старту. Для початку квесту потрібно натиснути «Старт» (рис. 3.17).

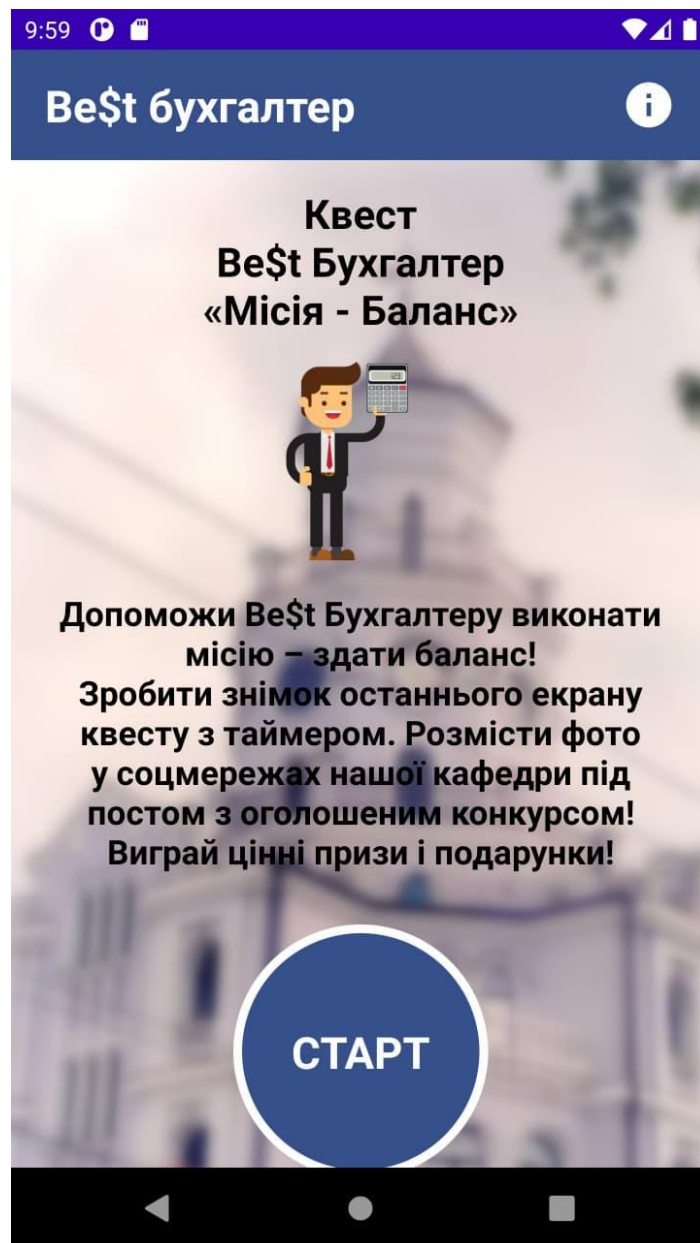


Рисунок 3.17 – Початковий екран

На першому рівні користувачу необхідно вирішити систему лінійних рівнянь (рис. 3.18(а)), записати результат (рис. 3.18(б)) та натиснути на кнопку «відповісти». У разі неправильної відповіді (рис. 3.18(в)) можна спробувати ще.

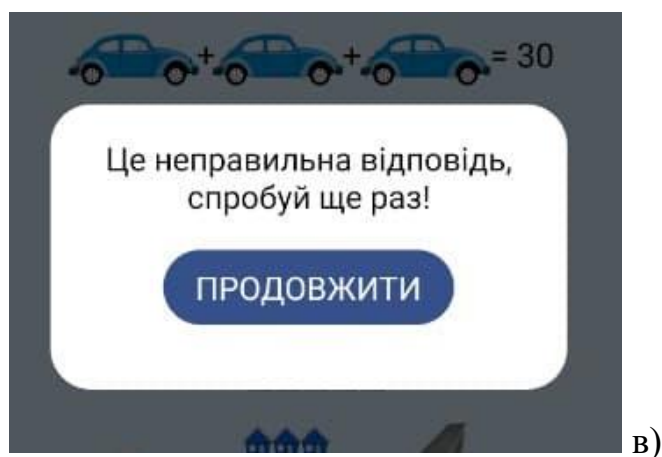
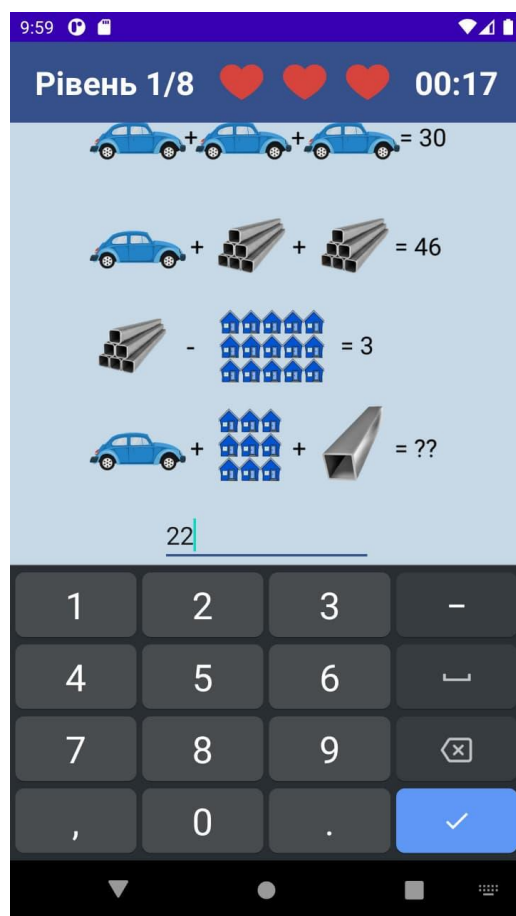
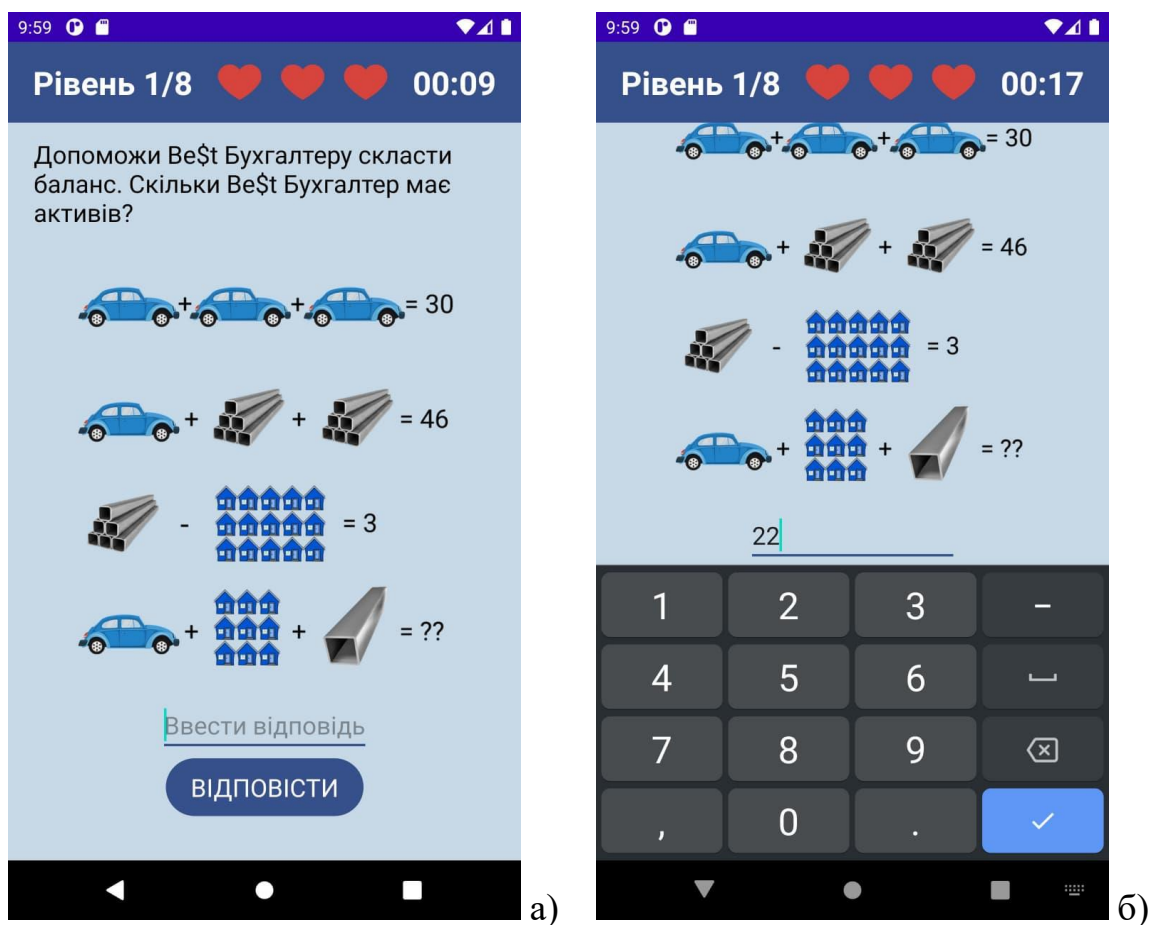
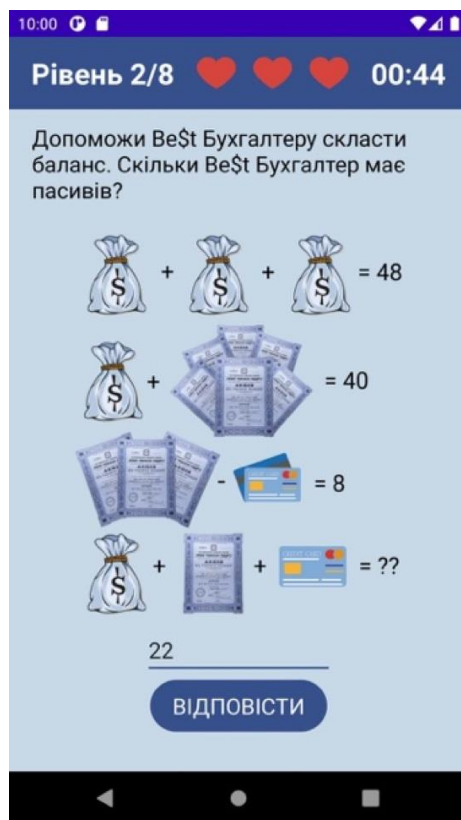
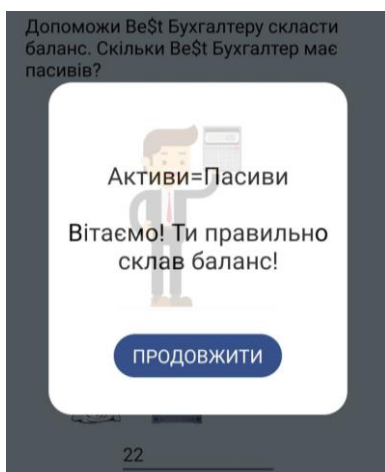


Рисунок 3.18 – Рівень 1/8 (а), введення відповіді (б), повідомлення про неправильну відповідь (в)

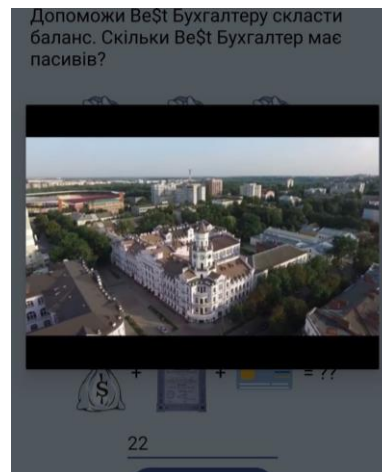
Аналогічне завдання на рівні 2/8 (рис. 3.19(а)). Після правильної відповіді, відкривається діалогове вікно для переходу на наступний рівень квесту (рис 3.19(б)), користувач натискає «Продовжити» та переглядає промо-ролик кафедри бухгалтерського обліку та оподаткування Сумського державного університету (рис. 3.19(в)), що буде представлений до перегляду протягом гри, з проходженням рівнів.



а)



б)



в)

Рисунок 3.19 – Введення даних на рівні 2/8 (а), діалогове вікно переходу на наступний рівень (б), промо-відео кафедри БОО (в)

Після перегляду відео, користувачу надається інформація про наступний рівень (рис. 3.20).

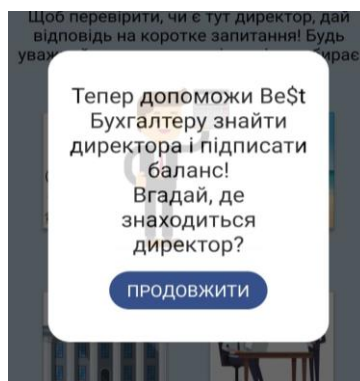
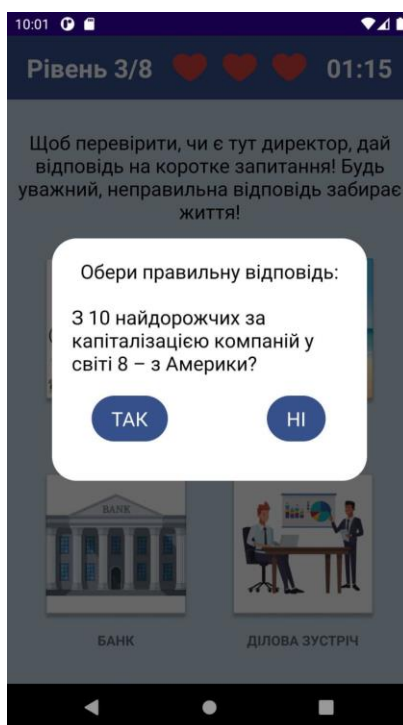


Рисунок 3.20 – Діалогове вікно

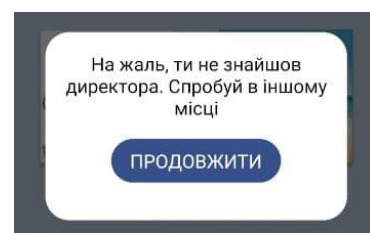
На третьому рівні (рис. 3.21(а)) необхідно знайти директора в одному з чотирьох місць, відповідаючи на питання (рис. 3.21(б)), у разі помилки можна спробувати в іншому місці (рис. 3.21(в)). На 3 і 5 рівнях на випадок невірної відповіді гравець має три життя, які можуть бути використані (рис. 3.22). Якщо всі вони використані і четверта відповідь невірна, то гравець повертається на початок гри і грає с початку.



а)



б)



в)

Рисунок 3.21 – Рівень 3/8 (а), питання із варіантами відповіді (б), повідомлення про неправильну відповідь (в)



Рисунок 3.22 – Відслідковування згорання життя

При наданні правильної відповіді, користувач знаходить директора (рис. 3.23).

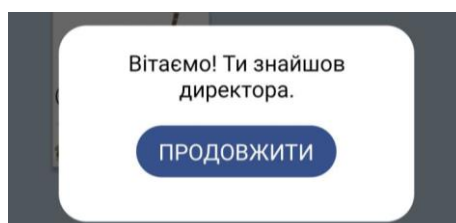


Рисунок 3.23 – Повідомлення про успішне завершення рівня

На четвертому рівні (рис. 3.24(а)) користувачу потрібно підібрати код до сейфу (рис. 3.24(б)), є можливість підбору необхідну кількість разів до правильної відповіді (рис. 3.24(в)).

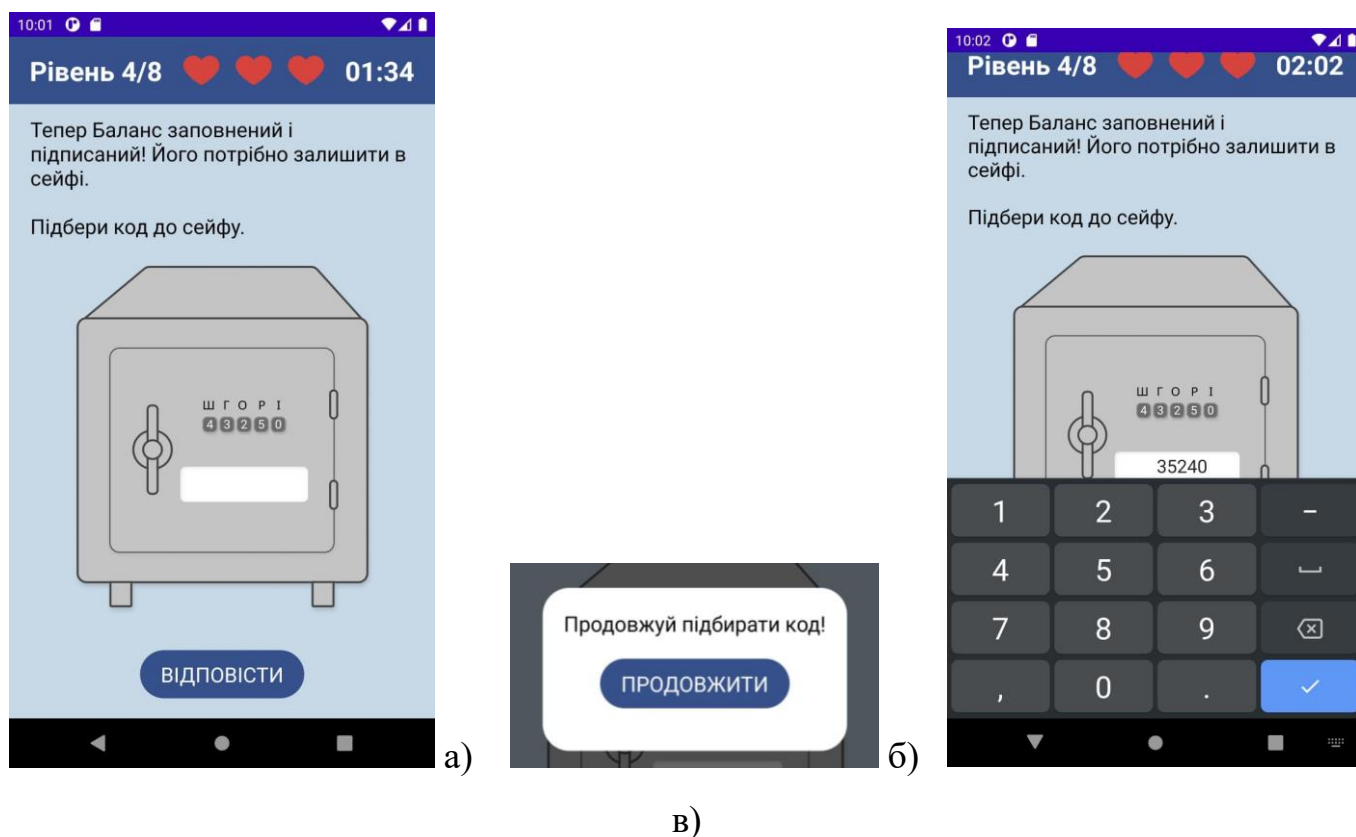
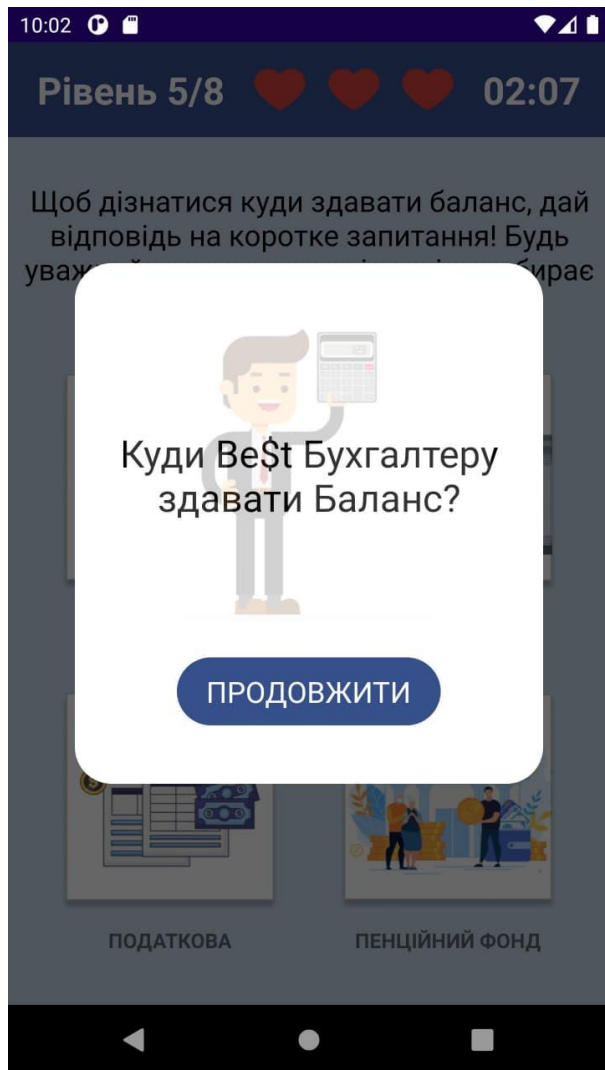
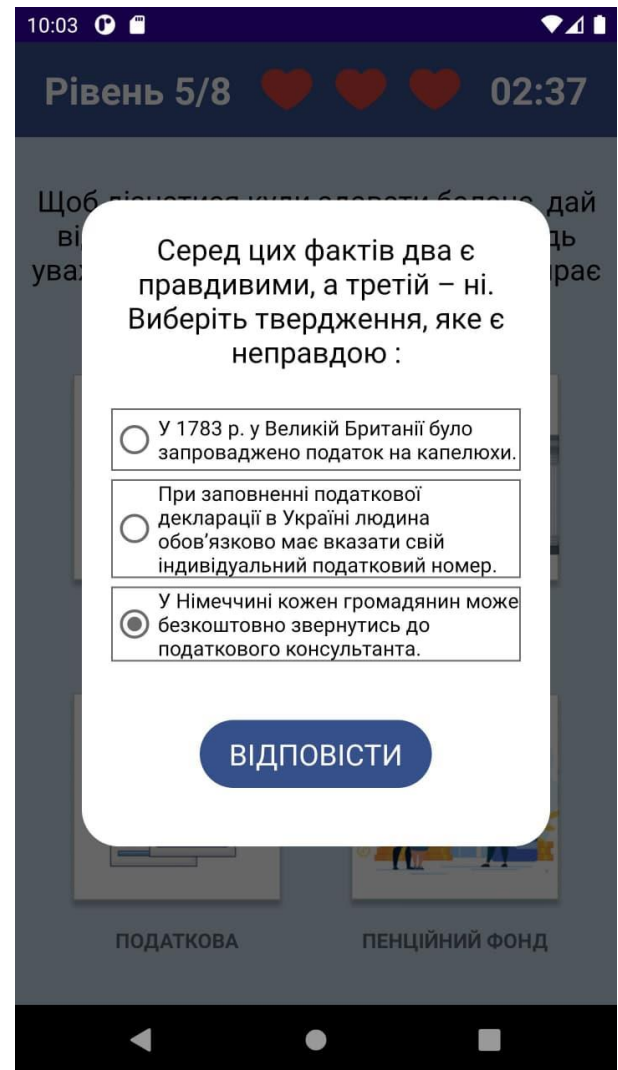


Рисунок 3.24 – Рівень 4/8 (а), неправильна відповідь (б), введення коду (в)

На п'ятому рівні гравцю необхідно здати баланс у потрібне місце (рис. 3.25(а)), обираючи твердження, що є неправдою (рис. 3.25(б)).



а)



б)

Рисунок 3.25 – Рівень 5/8 (а), обрати твердження (б)

У разі правильної відповіді, користувач переходить на рівень 6/8 (рис. 3.26).

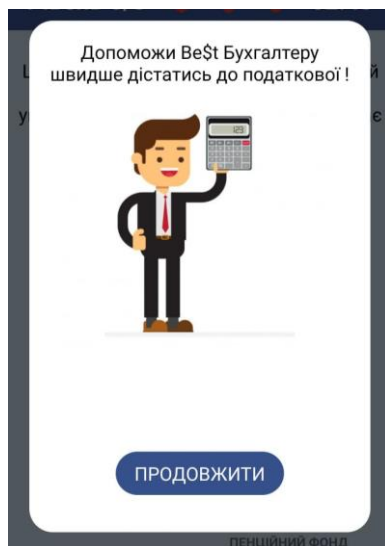


Рисунок 3.26 – Завдання для рівня 6/8

На шостому рівні необхідно вибрати три правдивих твердження (рис. 3.27). Точність відповіді впливає на транспорт, яким гравець дістанеться до податкової.

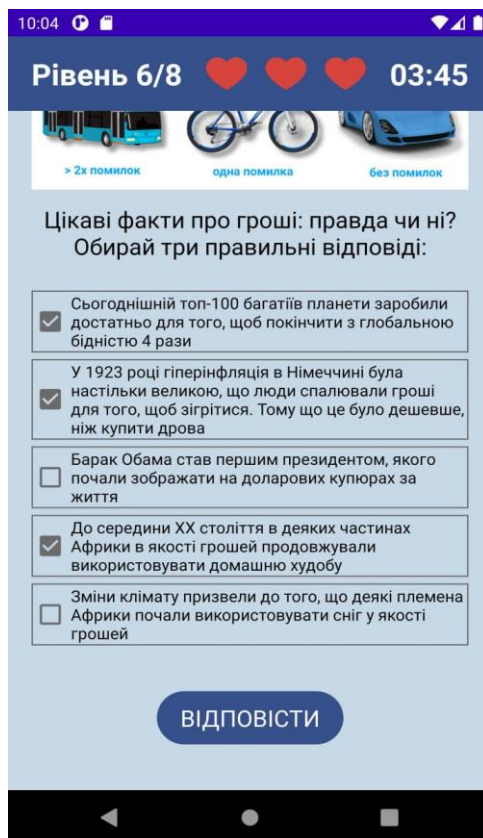


Рисунок 3.27 – Рівень 6/8

На сьомому рівні потрібно відгадати слово-пароль (рис. 3.28(a)). У разі правильної відповіді баланс потрапляє до податкової (рис. 3.28(б)).

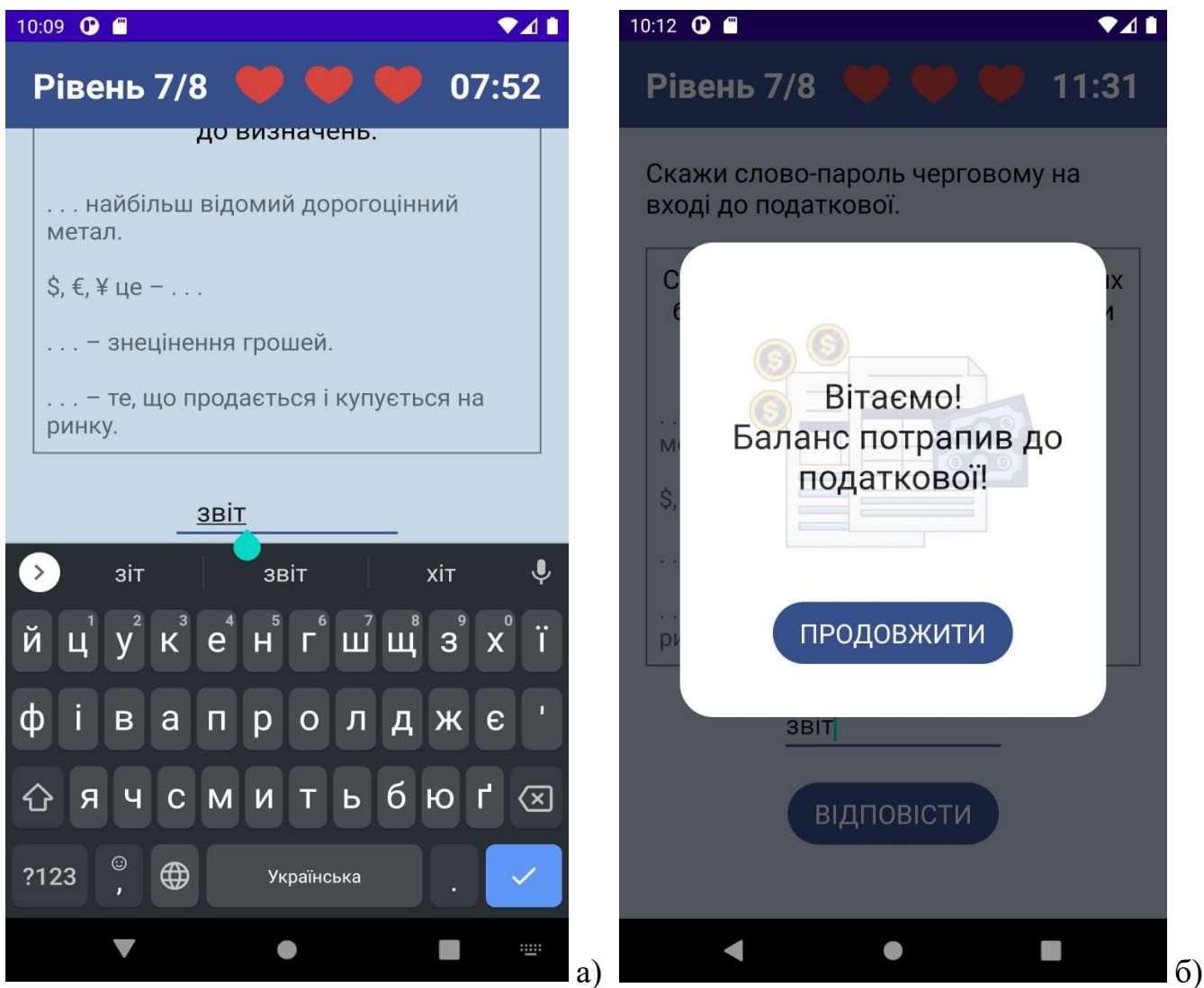


Рисунок 3.28 – Введення відповіді (а), баланс потрапив до податкової (б)

На восьмому рівні (рис. 3.29(а)) гравцеві необхідно знайти кабінет для здачі балансу (рис. 3.29(б)).

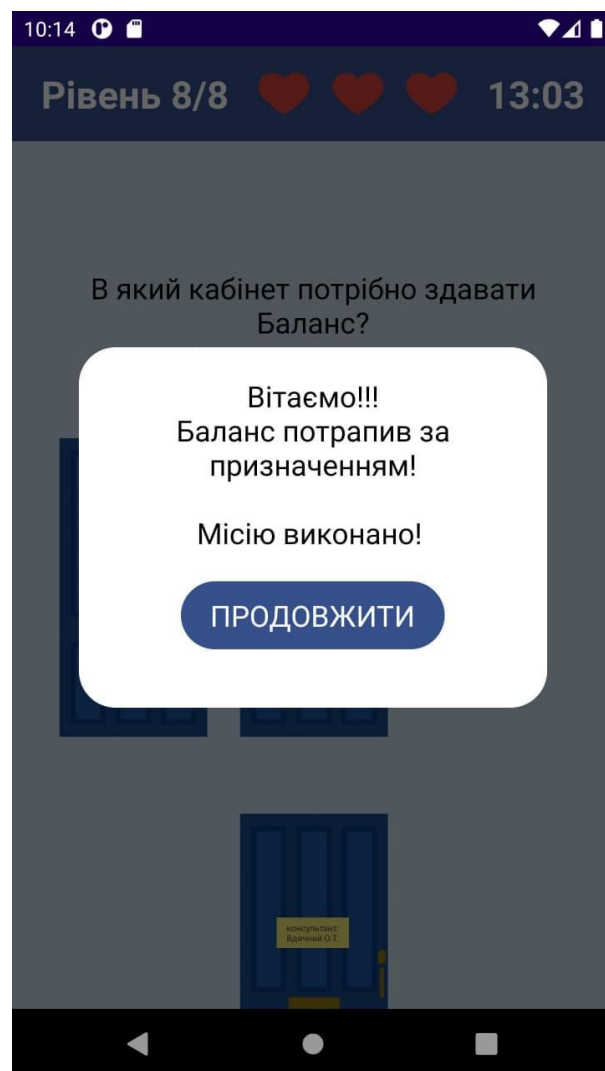
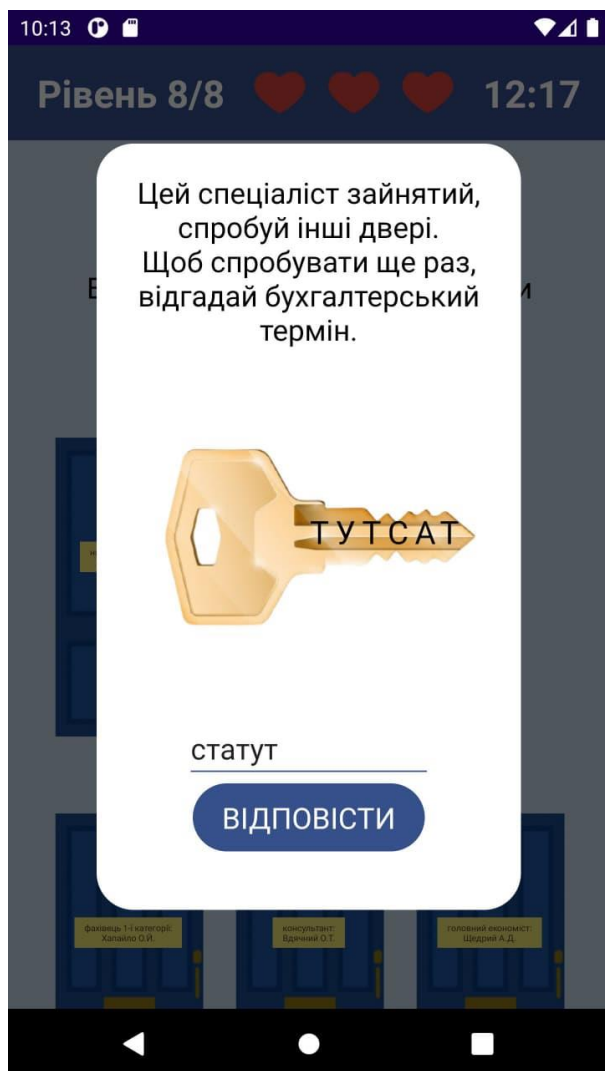


Рисунок 3.29 – Введення відповіді на рівні 8/8(а), місію виконано (б)

Після завершення місії, користувачу надається інформація про результат квесту та контакти кафедри БОО (рис. 3.30).



Рисунок 3.30 – Завершення квесту

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи бакалавра є мобільний ігровий додаток для потенціальних абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ на базі операційної системи Android.

Детальний аналіз вимог замовника та існуючих альтернатив дозволив розробити технічне завдання та визначити функціональні вимоги до додатку. Виконано проектування та моделювання функціонування системи, в результаті чого розроблено діаграми та моделі, що детально описують процеси гри та варіанти використання додатку. Розроблено прототип та дизайн додатку.

Додаток реалізований в середовищі розробки Android Studio мовою програмування Kotlin. Розробка мобільного додатку відбувалася з урахуванням значень розмірів екранів мобільних пристроїв під керуванням операційної системи Android.

Мобільний додаток впроваджений в профорієнтаційну роботу кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ. Ефективність впровадження додатку підтверджено актом, що наведений у додатку Д.

Мобільний додаток знаходиться у відкритому доступі і може бути завантажений з Google Play. Програмний код реалізації додатку поданий на реєстрацію рішення про авторське право.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сич Є.В., Марченко А.В. Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ. Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 19 – 20 квітня 2021 р. – Суми: Сумський державний університет, 2021.
2. Вибір вищої освіти після школи. — К.: Аналітичний центр Cedos, 2020. — 80 с.
3. РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ: АНАЛІЗ СВІТОВОГО РИНКУ [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://koloro.ua/ua/blog/issledovaniya/Razrabotka-mobilnih-prilojenii.html>.
4. Закон України про систему оподаткування [Електронний ресурс] - Режим доступу до ресурсу: <https://ips.ligazakon.net/document/T125100?an=1>
5. Android's Kotlin-first approach [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/kotlin/first>.
6. IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6#p62>
7. Основні методології обстеження організацій. Стандарт IDEF0 [Електронний ресурс] - Режим доступу до ресурсу: <https://www.cfin.ru/vernikov/idef/idef0.shtml>
8. Діаграма варіантів використання (use case diagram) [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12_2
9. Поняття ER-моделі [Електронний ресурс]– Режим доступу до ресурсу: <https://www.bestprog.net/uk/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ua/>

10. Mobile Tech Trends 2021: Era of Immersive Technologies [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://becominghuman.ai/mobile-tech-trends-2021-era-of-immersive-technologies-2b92ea65ee3e>.

11. Get started with the Navigation component [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.android.com/guide/navigation/navigation-getting-started>.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку інформаційної системи
«Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку
та оподаткування ННІ БіЕМ СумДУ »

ПОГОДЖЕНО:

Доцент кафедри комп'ютерних наук

_____ Марченко А.В.

Студент групи ІТ-71

_____ Сич Є.В.

Суми 2021

1. Призначення й мета створення мобільного додатку

1.1 Призначення мобільного додатку

Мобільний додаток має надавати інформацію абітурієнтам про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ, рекламувати її та надавати контактні дані.

1.2 Мета створення мобільного додатку

Збільшення кількості абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ за рахунок використання мобільного додатку.

1.3 Цільова аудиторія

До цільової аудиторії додатку можна віднести абітурієнтів вищих навчальних закладів України віком від 17-ти до 23-х років.

2 Вимоги до мобільного додатку

2.1 Вимоги до мобільного додатку в цілому

2.1.1 Вимоги до структури й функціонування мобільного додатку

Мобільний додаток має бути доступним в мережі в онлайн-магазині додатків “Google Play” під назвою “Бет Бухгалтер”. Мобільний додаток повинен складатися з 8-ми сторінок, кожна з яких буде рівнем у грі.

2.1.2 Вимоги до персоналу

Від персоналу не вимагається особливих технічних навичок для експлуатації мобільного додатку, для підтримки мобільного додатку потрібні навички роботи з IDE Android Studio, знання мов програмування Kotlin, Java.

2.1.3 Вимоги до збереження інформації

Уся інформація надана у мобільному додатку буде зберігатися у внутрішніх фалах та ресурсах додатку.

2.1.4 Вимоги до розмежування доступу

Розроблюваний мобільний додаток має бути загальнодоступним.

У даному мобільному додатку відсутнє розмежування за рівнем доступу до інформації, усі користувачі мають однаковий доступ.

2.2 Структура мобільного додатку

2.2.1 Загальна інформація про структуру мобільного додатку

Структура мобільного додатку являє собою набір екранів з інформацією, екранів з рівнями гри та допоміжних діалогових вікон.

Такими екранами є:

Початковий – перша сторінка при відкритті додатку, містить кнопку старт.

Правила гри – загальна інформація про правила гри.

Перший рівень – завдання вирішити систему лінійних рівнянь.

Другий рівень – завдання вирішити систему лінійних рівнянь.

Третій рівень – завдання знайти директора в одному з чотирьох місць.

Четвертий рівень – завдання підібрати код до сейфу.

П'ятий рівень – завдання здати баланс у потрібне місце.

Шостий рівень – завдання вибрати три правдивих твердження.

Сьомий рівень – завдання відгадати слово-пароль.

Восьмий рівень – загальна знайти кабінет для здачі балансу.

Результат – час за який пройдено гру, контактні дані.

2.2.2 Навігація

Навігація у додатку відбувається шляхом переходу на наступний екран у певній початково заданій послідовності.

2.2.3 Дизайн та структура додатку

Стиль мобільного додатку має бути сучасним, приємним для сприйняття, у якості основних кольорів було запропоновано використати сині та білі відтінки.

Прототип дизайну екранів додатку показано на рисунках .



Рисунок 2.1 – Початковий екран

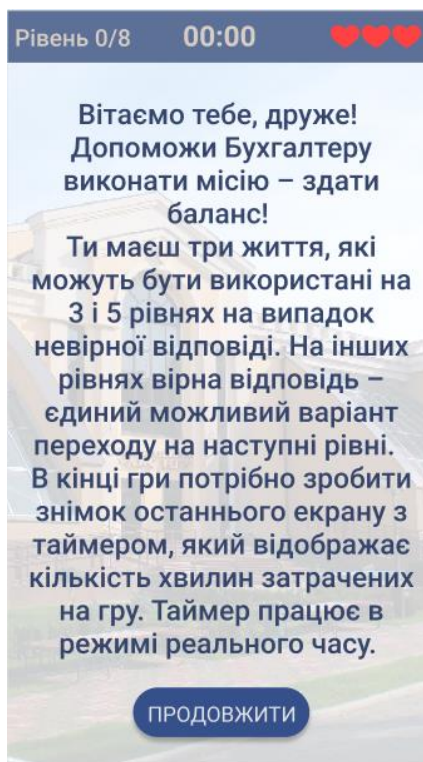


Рисунок 2.2 – Правила

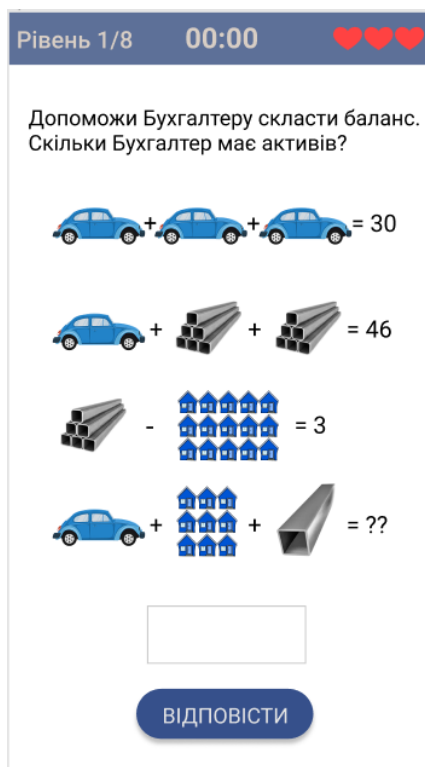







Рисунок 2.3 – Перший рівень

Рівень 2/8 00:00 

Допоможи Бухгалтеру скласти баланс. Скільки БетБухгалтер має пасивів?

 = 48
 = 40
 = 8
 = ??

ВІДПОВІСТИ

Рисунок 2.4 – Другий рівень

Рівень 3/8 00:00 

 САУНА
 МОРЕ
 БАНК
 ДІЛОВА ЗУСТРІЧ

Рисунок 2.5 – Третій рівень



Рисунок 2.6 – Четвертий рівень

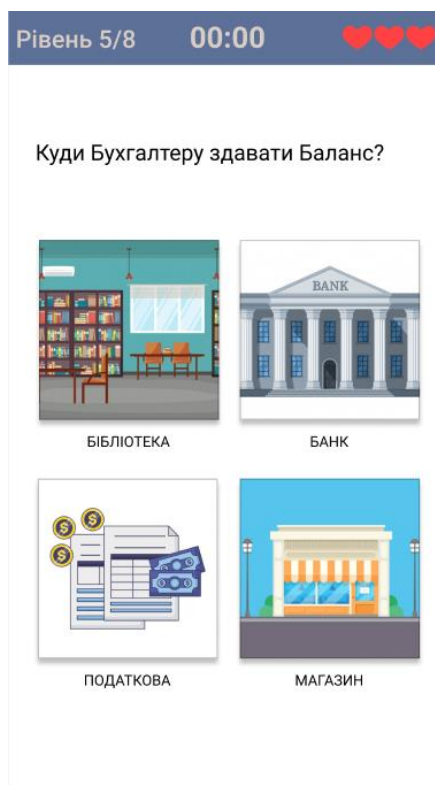


Рисунок 2.7 – П'ятий рівень

Рівень 6/8 00:00 



> 2х помилок



одна помилка




без помилок

Цікаві факти про гроші: правда чи ні?
Обирай три правильні відповіді:

- Факт 1
- Факт 2
- Факт 3
- Факт 4
- Факт 5

ВІДПОВІСТИ

Рисунок 2.8 – Шостий рівень

Рівень 7/8 00:00 

Скажи слово-пароль черговому на вході до податкової.

Слово-пароль складається з перших букв термінів які необхідно дібрати до визначень.

- ... – документ, що засвідчує права власності на винахід чи технологію.
- ... – ресурси підприємства, які використовуються для отримання доходів.
- ... – залишок на рахунку.
- ... – джерела утворення активів.
- ... – володіння й користування майном на договірних засадах.
- ... – це перевірка діяльності підприємства для встановлення законності, доцільності й ефективності дій.
- ... – економічна діяльність, суб'єкти якої уникають офіційного обліку і сплати податків.

ВІДПОВІСТИ

Рисунок 2.9 – Сьомий рівень



Рисунок 2.10 – Восьмий рівень

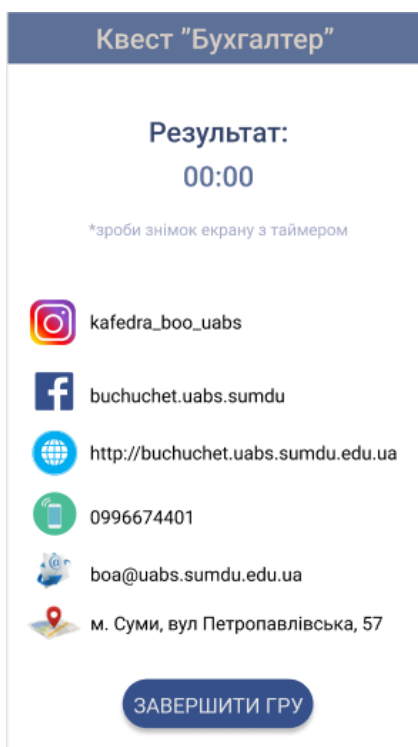


Рисунок 2.11 – Результат та контактні дані

2.3 Вимоги до функціонування системи

На основі потреб замовника розроблений додаток має задовольняти такі функціональні вимоги:

- поступове почергове проходження рівнів;
- втрата життів при певних не вірних діях;
- відслідковування часу проходження;
- відображення результату проходження завдань гри;
- відображення контактної інформації;

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація мобільного додатку відбувається з використанням:

- Android Studio
- Kotlin
- XML

2.4.2 Вимоги до лінгвістичного забезпечення

Мобільний додаток повинен бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Android Marshmallow або вище.

3 Склад і зміст робіт зі створення мобільного додатку

Докладний опис етапів роботи зі створення мобільного додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення мобільного додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	1 день
2	Складання технічного завдання	3 дні
3	Підготовка прототипу	5 днів
4	Створення макету дизайну мобільного додатку	2 дні
5	Розробка	20 днів
6	Робота над адаптивністю дизайну	2 дні
7	Тестування	1 день
8	Підготовка до випуску	1 день
9	Завантаження додатку у магазин “Google Play”	1 день
10	Завершення роботи	1 день
	Загальна тривалість робіт	37 днів

4 Вимоги до складу й змісту робіт із введення мобільного додатку в експлуатацію

Для того, щоб додаток можна було завантажити та користуватися, необхідно розмістити його у магазин “Google Play”, тому необхідно створити Google Developer аккаунт. Для завантаження додатку у магазин “Google Play” необхідно завантажити підписаний та сертифікований файл додатку та пройти валідацію Google Console.

ДОДАТОК Б

Планування робіт

Деталізація мети проекту методом SMART. Продуктом дипломного проекту є програма для визначення величини зерна металу, призначена для кафедр з дослідження металу у вищих навчальних закладах.

Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити програму для визначення величини зерна металу.
Measurable (вимірювана)	Результатом роботи проекту є оцінка замовника.
Achievable (досяжна)	Реалізація додатку здійснюється за допомогою середовища розробки Visual Studio за технологією Windows Forms з використанням мови програмування C#.
Relevant (реалістична)	У наявності є всі необхідні програмні засоби: Visual Studio. Розробник достатньо кваліфікований для виконання поставлених задач.
Time-framed (обмежена у часі)	Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником проекту. Проект повинен бути виконаний згідно з календарним планом.

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить WBS(Work Break Structure) - представлення проекту, виконане у вигляді ієрархічної структури робіт, що досягається за допомогою послідовної декомпозиції. Інструмент спрямований на детальне планування, оцінку

вартості, визначення та розподіл персональної відповідальності виконавців та інші - тобто, на основні роботи і результати, що визначають зміст проекту.

Як правило, на верхньому рівні вказується сам проект, під ним (на першому рівні) - основні результати, кожен з яких, в свою чергу, деталізується, тобто наступний рівень завжди менше попереднього за обсягом робіт і, як правило, включає 2 і більше пакетів робіт. При цьому в різних гілках WBS може бути різна кількість рівнів в залежності від потрібного ступеня деталізації.

В ході виконання даного проекту перший рівень структури називається «Віртуальна лабораторна робота з визначення величини зерна», який в свою чергу декомпозиується на чотири рівні: ініціалізація, планування, реалізація та завершення.

Ініціалізація додатку складається з таких етапів: ознайомлення з предметною областю, визначення в потребі додатку та ідентифікація ідеї проекту.

Наступний рівень – планування, який розбивається на два рівні:

1. аналіз документації:
2. визначення вимог:
 - визначення інструментарію;
 - планування WBS;
 - планування OBS;
 - складання календарного плану;
 - визначення бюджету;
 - визначення ризиків;

Третій рівень діаграми – реалізація, який розбивається на чотири рівні:

1. моделювання роботи додатку;
 2. розробка макету додатку;
 3. розробка функціоналу додатку;
- тестування:
- тестування розробником;
 - тестування незалежною особою;

Останній етап створення проекту є завершення, що містить в собі такі процеси: оформлення документації, архівація та експлуатація продукту.

На рисунку Б.2 приведена WBS-структура даного проекту.

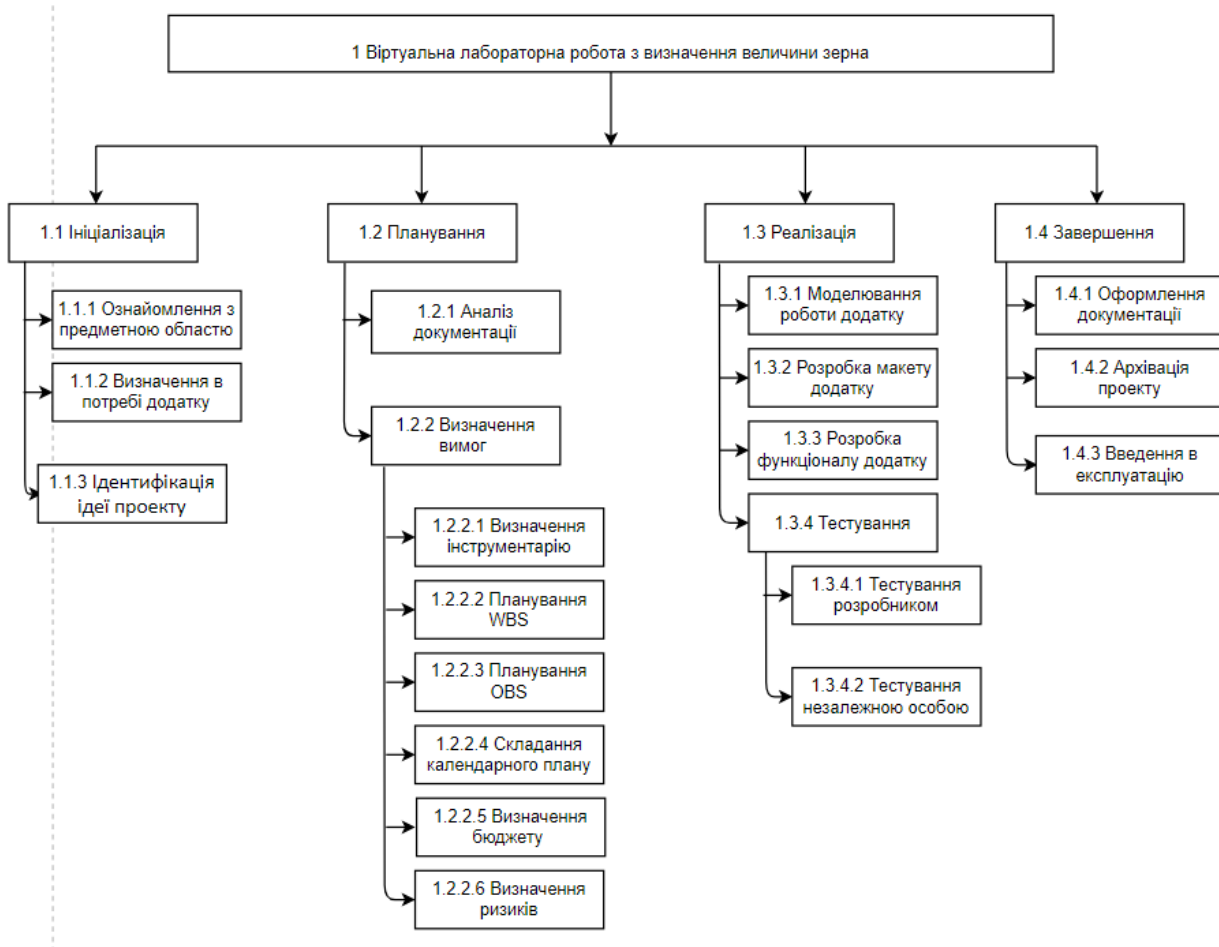


Рисунок Б.2 – WBS-структура проекту

Планування структури організації, для впровадження готового проекту (OBS). Після того, як була побудована WBS структура проекту наступним етапом є розроблення OBS (Organization Break structure) - склад, підпорядкованість, взаємодія і розподіл робіт по підрозділах і органам управління, між якими встановлюються певні відносини з приводу реалізації владних повноважень, потоків команд і інформації. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Список виконавців, що функціонують в проекті представлений в таблиці Б.2. Організаційна структура проекту зображена на рисунку Б.3.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Сич Є. В.	Виконує розробку основного функціоналу проекту.
Менеджер проекту	Марченко А.В.	Відповідає за виконання термінів, виконує збір та аналіз даних.
Тестувальник	Замовник	Відповідає за тестування функціоналу проекту

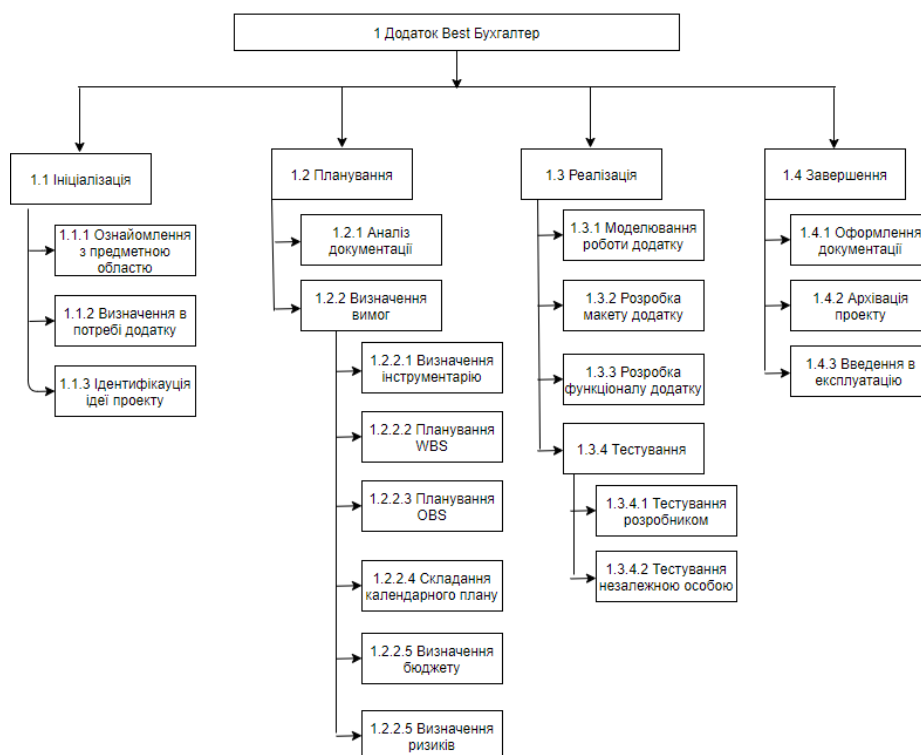


Рисунок Б.3 – OBS – структура проекту

Діаграма Ганта. Далі побудуємо календарний план виконання дипломного проекту. Найпоширеніший формат графіка в будь-якій галузі — діаграма Ганта. Управління проектами з діаграмами Ганта засноване на форматі гістограм. Це

допомагає відслідковувати відсоток робіт, виконаних по кожному завданню. Керівникам проектів дуже важливо правильно розподілити завдання і бути впевненими в тому, що проект буде завершений вчасно. Основна увага діаграм Ганта зосереджено на процентному завершенні кожного завдання. Крім того, діаграми Ганта краще для проектів з невеликою кількістю взаємопов'язаних завдань. Завдяки засобам програмного продукту MS Project була розроблена діаграма Ганта, яка у вигляді гістограми відображає тривалість кожного процесу, що був визначений на етапі формування WBS. Діаграма Ганта представлена на рисунку Б.4.

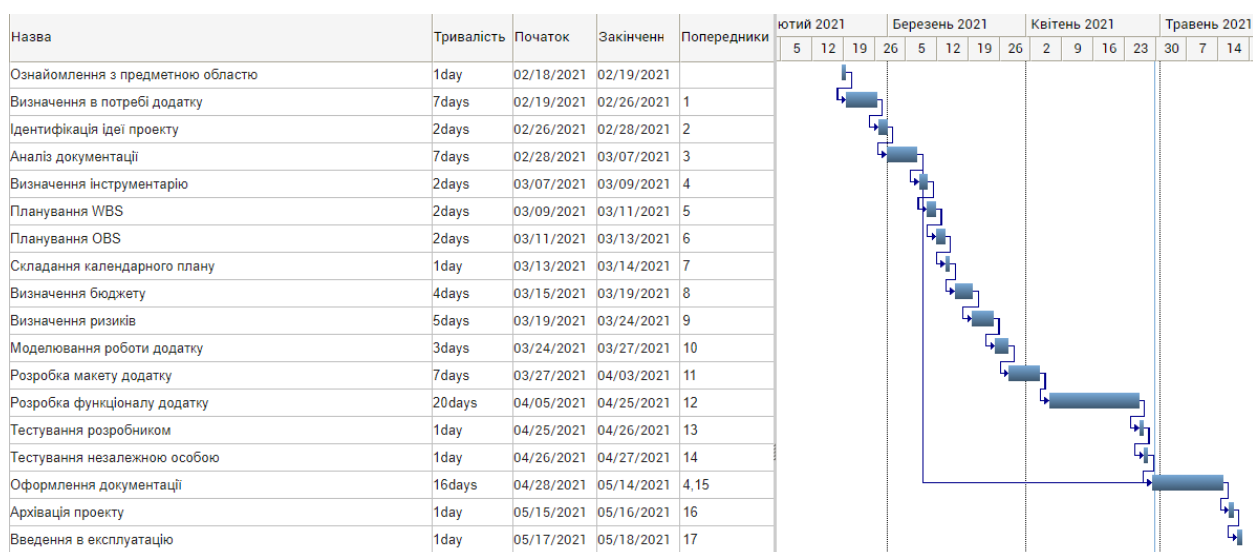


Рисунок Б.4 – Діаграма Ганта проекту

Назва	Тривалість	Початок	Закінченн	Попередники
Ознайомлення з предметною областю	1day	02/18/2021	02/19/2021	
Визначення в потребі додатку	7days	02/19/2021	02/26/2021	1
Ідентифікація ідеї проекту	2days	02/26/2021	02/28/2021	2
Аналіз документації	7days	02/28/2021	03/07/2021	3
Визначення інструментарію	2days	03/07/2021	03/09/2021	4
Планування WBS	2days	03/09/2021	03/11/2021	5
Планування OBS	2days	03/11/2021	03/13/2021	6
Складання календарного плану	1day	03/13/2021	03/14/2021	7
Визначення бюджету	4days	03/15/2021	03/19/2021	8
Визначення ризиків	5days	03/19/2021	03/24/2021	9
Моделювання роботи додатку	3days	03/24/2021	03/27/2021	10
Розробка макету додатку	7days	03/27/2021	04/03/2021	11
Розробка функціоналу додатку	20days	04/05/2021	04/25/2021	12
Тестування розробником	1day	04/25/2021	04/26/2021	13
Тестування незалежною особою	1day	04/26/2021	04/27/2021	14
Оформлення документації	16days	04/28/2021	05/14/2021	4,15
Архівація проекту	1day	05/15/2021	05/16/2021	16
Введення в експлуатацію	1day	05/17/2021	05/18/2021	17

Рисунок Б.5 – Список робіт для побудови діаграми Ганта

Аналіз ризиків. Ризик – ймовірнісна подія, яка може позитивно чи негативно вплинути на проект. Причиною виникнення ризиків є невизначеності, існуючі в кожному проекті. Ризики можуть бути «відомі» - ті, які визначені, оцінені, для яких можливе планування. Ризики «невідомі» - ті, які не ідентифіковані і не можуть бути прогнозовані. Хоча специфічні ризики і умови їх виникнення не визначені, але більшу частину ризиків можна передбачити.

Ідентифікація ризиків - визначення ризиків, здатних вплинути на проект, і документування їх характеристик.

Ідентифікація ризиків визначає, які ризики здатні вплинути на проект, і документує характеристики цих ризиків. Ідентифікація ризиків не буде ефективною, якщо вона не буде проводитися регулярно протягом реалізації проекту.

Ідентифікація ризиків повинна залучати якомога більше учасників: менеджерів проекту, користувачів, незалежних фахівців.

Класифікація ризиків:

1. За ймовірністю виникнення:

- слабо ймовірнісні;
- мало ймовірнісні;
- ймовірні;
- досить ймовірні;
- майже ймовірні.

2. За величиною втрат:

- мінімальна;
- низька;
- середня;
- висока;
- максимальна.

На основі цих даних була проведена класифікація ризиків для даного проекту, що наведена в таблиці Б.3.

Таблиця Б.3 – Класифікація ризиків

№	Назва ризику	Ймовірність	Величина втрат
1	Некоректно складене ТЗ	2	4
2	Недотримання календарного плану	1	3
3	Некоректна робота програмного забезпечення	4	4
4	Некоректна робота апаратного забезпечення	4	4
5	Хвороба розробника	2	2
6	Некоректне тестування	2	1

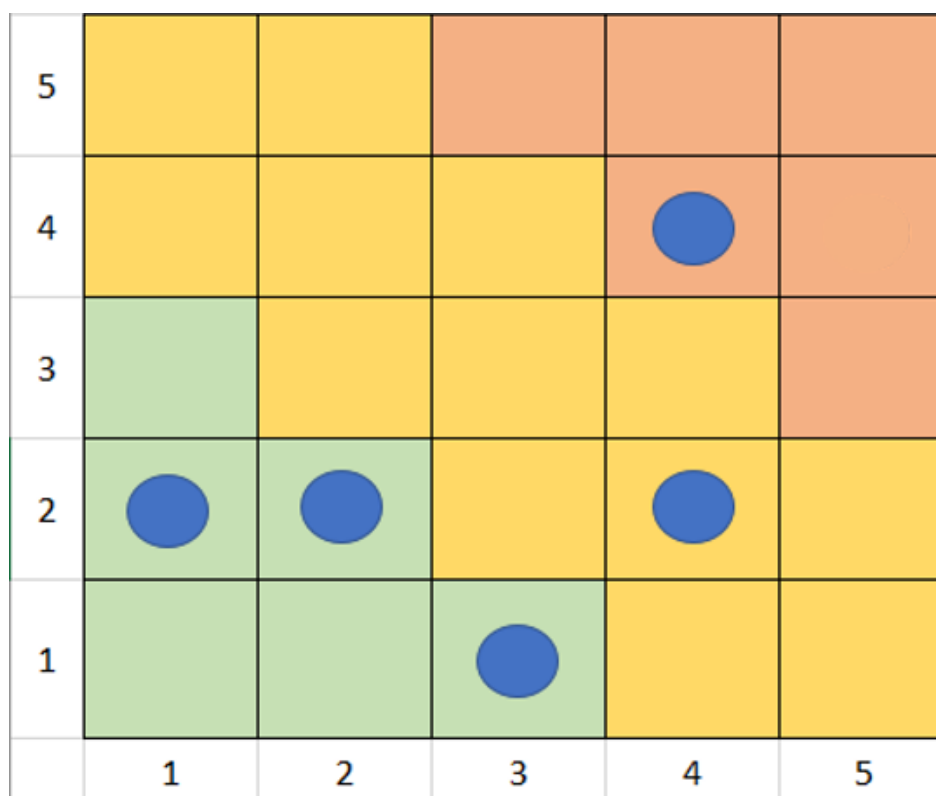


Рисунок Б.6 – Матриця ризиків

Далі визначаємо рівні ризиків та ступінь їх дії.

Рівні можуть бути:

- допустимі $1 < R < 4$;
- оправдані $5 < R < 10$;
- недопустимі $13 < R < 25$.

Ступінь дії ризиків:

- ті, що можна проігнорувати $1 < R < 4$;
- незначні $5 < R < 8$;
- помірні $9 < R < 10$;
- істотні $11 < R < 16$;
- критичні $17 < R < 25$.

На основі цих даних була виконана оцінка ступенів та рівнів для кожного ризику в проекті. Результати роботи представлені в таблиці Б.3.

Таблиця Б.4 – Визначення ступенів та рівнів ризиків

№	Назва ризику	Ймовірність ризику	Величина втрат	Рівень ризику	Ступінь дії
1	Некоректно складене ТЗ	2	4	Оправданий	Незначний
2	Недотримання календарного плану	1	3	Допустимий	Проігнорувати
3	Некоректна робота програмного забезпечення	4	4	Недопустимий	Істотний

Продовження таблиці Б.4

№	Назва ризику	Ймовірність ризику	Величина втрат	Рівень ризику	Ступінь дії
4	Некоректна робота апаратного забезпечення	4	4	Недопустимий	Істотний
5	Хвороба розробника	2	2	Допустимий	Проігнорувати
6	Некоректне тестування	2	1	Допустимий	Проігнорувати

Після виконання прогнозування виникнення ризиків та їх ступеню впливу на результат реалізації проекту, були розроблені варіанти запобігання та реакції на кожний із них. Результати даного етапу представлені в таблиці Б.4.

Таблиця Б.5 – Варіанти запобігання та реакції на ризики

Ризики проекту	План запобігання ризику	План реакції на ризик
Некоректно складене ТЗ	Замовник повинен скласти детальне ТЗ та озвучити терміни. Замовник та розробник повинні обговорити та затвердити його.	Уважно та чітко окреслити те, що було виконано невірно (після розмови із замовником) та зробити правки.

Продовження таблиці Б.5

Ризики проекту	План запобігання ризику	План реакції на ризик
Недотримання календарного плану	Затвердження зазначених термінів із замовником. Командна робота над планом виконання всіх термінів.	1. Обговорення варіантів внесення правок до термінів реалізації із керівником та замовником. 2. Домовитися про умови зміни термінів із замовником. Якщо це недопустимо, тоді переорганізувати роботу таким чином, щоб в результаті терміни виконувалися.
Некоректна робота програмного забезпечення	Встановлення ліцензійного програмного забезпечення з перевірених джерел перед початком роботи. Забезпечити наявність антивірусного програмного забезпечення.	Перезапуск або переустановлення програми, яка дала збій.
Некоректна робота апаратного забезпечення	Раз на 4-6 місяців виконувати перевірку працездатності апаратного забезпечення	Виконати ремонт апаратного забезпечення, якщо терміни виконання завдань не дозволяють чекати, знайти тимчасову заміну.

Продовження таблиці Б.5

Хвороба розробника	Виконувати певну частину роботи в команді для того, щоб члени проекту змогли замінити один одного при необхідності. При плануванні термінів залишити декілька резервних днів для таких випадків.	Передати повноваження робітника іншому члену команди, якщо цього вимагають терміни виконання.
Некоректне тестування	Виконати пошук кваліфікованого тестувальника в даній предметній області.	Передати проект на додаткове тестування кваліфікованому спеціалісту.

ДОДАТОК В

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ

Сич Є.В., студент; Марченко А.В., доцент
Сумський державний університет, м. Суми, Україна

На даний момент складно уявити життя людей без спеціальних пристроїв, механізмів та технологій використання яких спрощує та підвищує якість життя. Вже на протязі більше ніж десяти років мобільні додатки займають далеко не останнє місце у списку речей, які значно допомагають у повсякденному житті. В наші дні смартфон є одним з найзручніших пристроїв для навчання, розваг, покупок. Саме через все вищесказане можна вважати, що мобільні додатки являються достатньо актуальними.

На даний час існує достатньо велика конкуренція між підприємствами та установами майже у всіх сферах послуг. Однією з таких сфер є освіта, де успіх перш за все залежить від кількості залучених студентів. На скільки нам відомо найбільший відсоток студентів в Україні це особи від 17 до 23 років, саме ця цільова група проводить найбільше часу у мережі за допомогою смартфонів. Тобто можна передбачити, що для донесення інформації до більшої кількості абітурієнтів буде доцільно використовувати мобільний додаток.

Отже метою проєкту є розробка мобільного ігрового додатку для потенційних абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ на базі операційної системи Android.

Призначення додатку полягає у донесенні інформації про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ у ігровому форматі за допомогою мобільного додатку.

Гра буде складатися з 8-ми рівнів на яких будуть розташовані завдання, переможцем стає той учасник, що швидше за всіх подолає вісім рівнів.

Використання даного додатку допоможе більш обширно та цікаво доносити інформацію до потенційних абітурієнтів про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ.

ДОДАТОК Г. КОД ПРОГРАМИ

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.betaccountant.view.ToolbarView
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple_200"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="0dp"
        app:max_lives_amount="3"
        app:remaining_lives_amount="3"
        app:remaining_life_img="@drawable/ic_remaining_life_heart"
        app:used_life_img="@drawable/ic_used_life_heart"/>

    <fragment
        android:id="@+id/fragment_container"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@id/toolbar"
        app:navGraph="@navigation/navigation" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

bool_question_dialog_layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/dialog_background"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">

    <TextView

```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/choose_right_answer"
android:textColor="@color/black"
android:textSize="20sp" />

```

```

<TextView
    android:id="@+id/questionDialogQuestionTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@color/black"
    android:textSize="20sp" />

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">

```

```

<TextView
    android:id="@+id/questionDialogYesBtn"
    style="@style/baseButtonStyle"
    android:layout_gravity="center"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/yes"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```

<TextView
    android:id="@+id/questionDialogNoBtn"
    style="@style/baseButtonStyle"
    android:layout_gravity="center"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/no"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.9"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```

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

```

fragment_eighth_level.xml

```

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

```

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor"
    android:fillViewport="true"
    tools:context=".fragment.EighthLevel">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="@dimen/sp_20">

        <TextView
            android:id="@+id/textView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="@string/eighth_level_question"
            android:textColor="@color/black"
            android:textSize="@dimen/sp_20"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="@id/guideline" />

        <GridLayout
            android:id="@+id/imageGrid"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp"
            android:columnCount="3"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/textView3" />

        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/guideline"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            app:layout_constraintGuide_percent="0.1" />
    </androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>

```

fragment_fifth_level.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

```



```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/mainBackgroundColor"
tools:context=".fragment.ThirdLevel">

```

```

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="20dp"
    android:layout_marginTop="30dp"
    android:gravity="center"
    android:text="@string/fifth_level_task"
    android:textColor="@color/black"
    android:textSize="@dimen/sp_20"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```

<GridLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:columnCount="2"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView6"
    tools:ignore="UselessParent">

```

```

<LinearLayout
    android:id="@+id/administrationContainer"
    style="@style/thirdFifthLevelItemsStyle">

```

```

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/administration" />

```

```

    <TextView
        style="@style/thirdFifthLevelItemTextStyle"
        android:text="@string/administration" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:id="@+id/bankBalanceContainer"
    style="@style/thirdFifthLevelItemsStyle">

```

```

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/bank" />

```

```

<TextView
    style="@style/thirdFifthLevelItemTextStyle"
    android:text="@string/bank" />

</LinearLayout>

<LinearLayout
    android:id="@+id/taxContainer"
    style="@style/thirdFifthLevelItemsStyle">

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/tax" />

    <TextView
        style="@style/thirdFifthLevelItemTextStyle"
        android:text="@string/tax" />

</LinearLayout>

<LinearLayout
    android:id="@+id/shopContainer"
    style="@style/thirdFifthLevelItemsStyle">

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/fond" />

    <TextView
        style="@style/thirdFifthLevelItemTextStyle"
        android:text="@string/fond" />

</LinearLayout>

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

```

fragment_fourth_level.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor"
    tools:context=".fragment.FourthLevel">

    <TextView
        android:id="@+id/thirdQuestionTxt"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

```

```

android:text="@string/fourth_level_question"
android:textColor="@color/black"
android:textSize="20sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.023"
app:layout_constraintWidth_percent="0.9" />

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/constraintLayout"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHeight_percent="0.6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/thirdQuestionTxt"
    app:layout_constraintVertical_bias="0.109"
    app:layout_constraintWidth_percent="0.9">

```

```

<ImageView
    android:id="@+id/thirdQuestionImg"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/safe" />

```

```

<EditText
    android:id="@+id/inputCode"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="@android:color/transparent"
    android:backgroundTint="@color/blue_36"
    android:gravity="center"
    android:importantForAutofill="no"
    android:inputType="number"
    android:maxLength="7"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.617"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/thirdQuestionImg"
    app:layout_constraintVertical_bias="0.631"
    app:layout_constraintWidth_percent="0.4" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<TextView
    android:id="@+id/finishGameButton"
    style="@style/baseButtonStyle"
    android:background="@drawable/answer_btn_bg"

```

```

android:text="@string/answer"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/constraintLayout"
app:layout_constraintVertical_bias="0.6" />

```

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

fragment_image_question_level.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor"
    tools:context="fragment.ImageQuestionLevelFragment"
    android:fillViewport="true">

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

<TextView
    android:id="@+id/questionTxt"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="@string/first_level_question"
    android:textSize="@dimen/sp_20"
    android:textColor="@color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.023"
    app:layout_constraintWidth_percent="0.9" />

```

```

<ImageView
    android:id="@+id/questionImg"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:src="@color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHeight_percent="0.6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/questionTxt"
    app:layout_constraintVertical_bias="0.109"
    app:layout_constraintWidth_percent="0.9" />

```

```

<EditText
    android:id="@+id/answerInput"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/input_answer"
    android:importantForAutofill="no"
    android:inputType="number"
    android:textSize="@dimen/sp_20"
    android:backgroundTint="@color/blue_36"
    app:layout_constraintBottom_toTopOf="@+id/answerBtn"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/questionImg" />

```

```

<TextView
    android:id="@+id/answerBtn"
    style="@style/baseButtonStyle"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/answer"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/questionImg"
    app:layout_constraintVertical_bias="0.6" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

</ScrollView>

```

fragment_result.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor"
    android:fillViewport="true"
    tools:context="fragment.ResultFragment">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="20dp">

        <TextView
            android:id="@+id/resultHint"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/make_screenshot"
            android:visibility="gone"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/guideline3" />

```

```

<TextView
    android:id="@+id/resultTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/result"
    android:textColor="@color/blue_36"
    android:textSize="26sp"
    android:visibility="gone"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/resultHint" />

```

```

<TextView
    android:id="@+id/resultTimeTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="00:00"
    android:textColor="@color/blue_36"
    android:textColorLink="@color/black"
    android:textSize="32sp"
    android:textStyle="bold"
    android:visibility="gone"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/resultTxt" />

```

```

<TextView
    android:id="@+id/instagramLink"
    style="@style/contactItemStyle"
    android:layout_marginTop="20dp"
    android:text="@string/sumdu_kafedra_boo"
    app:drawableStartCompat="@drawable/ic_instagram"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/resultTimeTxt" />

```

```

<TextView
    android:id="@+id/facebookLink"
    style="@style/contactItemStyle"
    android:text="@string/facebook_link"
    app:drawableStartCompat="@drawable/ic_facebook"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/instagramLink" />

```

```

<TextView
    android:id="@+id/siteLink"
    style="@style/contactItemStyle"
    android:text="@string/site_link"

```

```

app:drawableStartCompat="@drawable/ic_internet"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/facebookLink" />

```

```

<TextView
    android:id="@+id/mobilePhoneLink"
    style="@style/contactItemStyle"
    android:text="@string/mobile_phone_number"
    app:drawableStartCompat="@drawable/ic_phone"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/siteLink" />

```

```

<TextView
    android:id="@+id/staticPhoneLink"
    style="@style/contactItemStyle"
    android:text="@string/static_phone_number"
    app:layout_constraintBottom_toBottomOf="@+id/mobilePhoneLink"
    app:layout_constraintStart_toEndOf="@id/mobilePhoneLink"
    app:layout_constraintTop_toBottomOf="@+id/siteLink" />

```

```

<TextView
    android:id="@+id/emailLink"
    style="@style/contactItemStyle"
    android:text="@string/email_link"
    app:drawableStartCompat="@drawable/ic_email"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/mobilePhoneLink" />

```

```

<TextView
    android:id="@+id/locationLink"
    style="@style/contactItemStyle"
    android:text="@string/location_link"
    app:drawableStartCompat="@drawable/ic_location"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/emailLink" />

```

```

<TextView
    android:id="@+id/textView7"
    style="@style/contactItemStyle"
    android:gravity="center"
    android:layout_marginTop="40dp"
    android:text="@string/quest_developer"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/locationLink" />

```

```

<TextView
    android:id="@+id/finishGameButton"
    style="@style/baseButtonStyle"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/finish_game"

```

```

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView7"
app:layout_constraintVertical_bias="1" />

```

```

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.02" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

```
</ScrollView>
```

fragment_rules.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="fragment.RulesFragment"
    android:fillViewport="true"
    android:background="@drawable/library_bg"
    android:padding="20dp">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:gravity="center">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="26sp"
        android:textColor="@color/blue_36"
        android:gravity="center"
        android:text="@string/rules"/>

```

```

<TextView
    android:id="@+id/continueRulesBtn"
    style="@style/baseButtonStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/continue" />

```



```

</LinearLayout>
</ScrollView>

```

fragment_seventh_level.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor"
    android:fillViewport="true">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="20dp"
        tools:context=".fragment.SeventhLevel">

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/say_password_for_duty"
            android:textColor="@color/black"
            android:textSize="20sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <LinearLayout
            android:id="@+id/linearLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:background="@drawable/transparent_background_with_border"
            android:orientation="vertical"
            android:padding="10dp"
            app:layout_constraintTop_toBottomOf="@+id/textView2"
            tools:layout_editor_absoluteX="20dp">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="@string/pass_word_contains"
                android:textColor="@color/black"
                android:textSize="20sp" />

            <TextView
                android:layout_marginTop="30dp"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/seventh_level_task"
        android:textSize="18sp" />
</LinearLayout>

<EditText
    android:id="@+id/seventhLvlAnswerInput"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:backgroundTint="@color/blue_36"
    android:hint="@string/input_answer"
    android:importantForAutofill="no"
    android:inputType="text"
    android:textSize="@dimen/sp_20"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />

<TextView
    android:id="@+id/seventhLvlAnswerBtn"
    style="@style/baseButtonStyle"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/answer"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/seventhLvlAnswerInput" />

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

```

fragment_sixth_level.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="20dp"
        tools:context=".fragment.SixthLevel">

        <ImageView
            android:id="@+id/imageView"

```

```

    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:src="@drawable/transport"
    app:layout_constraintTop_toTopOf="parent" />

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/facts_about_money_question"
    android:textColor="@color/black"
    android:textSize="@dimen/sp_20"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />

```

```

<LinearLayout
    android:id="@+id/factsLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    tools:layout_editor_absoluteX="20dp" />

```

```

<TextView
    android:id="@+id/sixLVLAnswerBtn"
    style="@style/baseButtonStyle"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/answer"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/factsLayout"
    app:layout_constraintVertical_bias="1" />

```

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

```
</ScrollView>
```

fragment_start.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="fragment.StartFragment"
    android:fillViewport="true">

    <FrameLayout
        android:layout_width="match_parent"

```

```
android:layout_height="wrap_content">
```

```
<ImageView
```

```
    android:id="@+id/imageView2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/start_screen_background"
    android:scaleType="centerCrop"
    android:src="@drawable/start_screen_background" />
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingHorizontal="20dp">
```

```
<TextView
```

```
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:gravity="center"
    android:text="@string/start_title"
    android:textColor="@color/black"
    android:textSize="@dimen/sp_26"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<ImageView
```

```
    android:id="@+id/imageView3"
    android:layout_width="@dimen/start_image_size"
    android:layout_height="@dimen/start_image_size"
    android:src="@drawable/accountant"
    app:layout_constraintBottom_toTopOf="@+id/textView4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView8" />
```

```
<TextView
```

```
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/rules"
    android:textColor="@color/black"
    android:textSize="@dimen/sp_20"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```

<TextView
    android:id="@+id/startBtn"
    android:layout_width="@dimen/start_button_size"
    android:layout_height="@dimen/start_button_size"
    android:background="@drawable/circle_button_bg"
    android:gravity="center"
    android:text="Crapt"
    android:textAllCaps="true"
    android:textColor="@color/white"
    android:textSize="@dimen/start_button_text_size"
    android:textStyle="bold"
    android:layout_marginTop="30dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4" />

```

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

```
</FrameLayout>
```

```
</ScrollView>
```

fragment_third_level.xml

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

```

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/mainBackgroundColor"
    tools:context=".fragment.ThirdLevel">

```

```
<TextView
```

```

    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="20dp"
    android:layout_marginTop="30dp"
    android:gravity="center"
    android:text="@string/third_level_task"
    android:textColor="@color/black"
    android:textSize="@dimen/sp_20"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```
<GridLayout
```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:columnCount="2"
    app:layout_constraintBottom_toBottomOf="parent"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView5"
tools:ignore="UselessParent">

```

```

<LinearLayout
    android:id="@+id/bathContainer"
    style="@style/thirdFifthLevelItemsStyle">

```

```

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/bath" />

```

```

    <TextView
        style="@style/thirdFifthLevelItemTextStyle"
        android:text="@string/bath" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:id="@+id/seaContainer"
    style="@style/thirdFifthLevelItemsStyle">

```

```

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/sea" />

```

```

    <TextView
        style="@style/thirdFifthLevelItemTextStyle"
        android:text="@string/sea" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:id="@+id/bankContainer"
    style="@style/thirdFifthLevelItemsStyle">

```

```

    <ImageView
        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/bank" />

```

```

    <TextView
        style="@style/thirdFifthLevelItemTextStyle"
        android:text="@string/bank" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:id="@+id/businessMeetingContainer"
    style="@style/thirdFifthLevelItemsStyle">

```

```

    <ImageView

```

```

        style="@style/thirdFifthLevelItemImageStyle"
        android:src="@drawable/business_meeting" />

```

```

<TextView
    style="@style/thirdFifthLevelItemTextStyle"
    android:text="@string/business_meeting" />

```

```
</LinearLayout>
```

```
</GridLayout>
```

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

game_over_dialog.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/dialog_background"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">

    <TextView
        android:id="@+id/storyDialogTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="@dimen/sp_20"
        android:textColor="@color/black"
        android:text="Гру завершено!\nНа жаль ти втратив всі життя"/>

    <TextView
        android:id="@+id/completeBtn"
        style="@style/baseButtonStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="20dp"
        android:background="@drawable/answer_btn_bg"
        android:text="@string/complete" />
</LinearLayout>

```

guess_term_dialog_layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/dialog_background"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">

```

```

<TextView
    android:id="@+id/guessTermDialogText"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textColor="@color/black"
    android:textSize="20sp" />

<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/guessTermDialogTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|end"
        android:layout_marginEnd="20dp"
        android:gravity="center"
        android:letterSpacing="0.2"
        android:textColor="@color/black"
        android:textSize="22sp"
        android:translationZ="1dp"
        android:textAllCaps="true"/>

    <ImageView
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:layout_gravity="center"
        android:src="@drawable/key" />
</FrameLayout>

<EditText
    android:id="@+id/answerTermInput"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/input_answer"
    android:importantForAutofill="no"
    android:inputType="text"
    android:backgroundTint="@color/blue_36"
    android:textSize="@dimen/sp_20"/>

<TextView
    android:id="@+id/guessTermAnswerBtn"
    style="@style/baseButtonStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/answer" />

```



```
</LinearLayout>
```

layout_toolbar_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="@color/blue_36"
    android:orientation="horizontal"
    android:paddingHorizontal="@dimen/toolbar_horizontal_padding"
    android:paddingVertical="@dimen/toolbar_vertical_padding">

    <TextView
        android:id="@+id/toolbarLabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="start"
        android:textColor="@color/white"
        android:textSize="@dimen/toolbar_text_size"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <LinearLayout
        android:id="@+id/toolbarLivesCounter"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:orientation="horizontal"
        android:paddingStart="10dp"
        android:paddingEnd="15dp"
        app:layout_constraintStart_toEndOf="@id/toolbarLabel"
        app:layout_constraintEnd_toStartOf="@id/toolbarTimeCounter"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>

    <Chronometer
        android:id="@+id/toolbarTimeCounter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="end"
        android:textColor="@color/white"
        android:textSize="@dimen/toolbar_text_size"
        android:textStyle="bold"
        android:visibility="gone"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>
```

```

<ImageView
    android:id="@+id/infoButton"
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:src="@drawable/ic_info_24"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

navigation.xml

```

<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/navigation"
    app:startDestination="@id/startFragment">

    <fragment
        android:id="@+id/startFragment"
        android:name="com.betaccountant.fragment.StartFragment"
        android:label="fragment_start"
        tools:layout="@layout/fragment_start" />

    <fragment
        android:id="@+id/firstLevelFragment"
        android:name="com.betaccountant.fragment.ImageQuestionLevelFragment"
        android:label="fragment_first_level"
        tools:layout="@layout/fragment_image_question_level" />

    <fragment
        android:id="@+id/thirdLevelFragment"
        android:name="com.betaccountant.fragment.ThirdLevel"
        android:label="fragment_third_level"
        tools:layout="@layout/fragment_third_level" />

    <fragment
        android:id="@+id/fourthLevelFragment"
        android:name="com.betaccountant.fragment.FourthLevel"
        android:label="fragment_fourth_level"
        tools:layout="@layout/fragment_fourth_level" />

    <fragment
        android:id="@+id/fivesLevelFragment"
        android:name="com.betaccountant.fragment.FifthLevel"
        android:label="fragment_fives_level"
        tools:layout="@layout/fragment_fifth_level" />

    <fragment
        android:id="@+id/sixthLevelFragment"
        android:name="com.betaccountant.fragment.SixthLevel"

```

```

    android:label="fragment_sixth_level"
    tools:layout="@layout/fragment_sixth_level" />

```

```

<fragment
    android:id="@+id/seventhLevelFragment"
    android:name="com.betaccountant.fragment.SeventhLevel"
    android:label="fragment_seventh_level"
    tools:layout="@layout/fragment_seventh_level" />

```

```

<fragment
    android:id="@+id/eighthLevelFragment"
    android:name="com.betaccountant.fragment.EighthLevel"
    android:label="fragment_eighth_level"
    tools:layout="@layout/fragment_eighth_level" />

```

```

<fragment
    android:id="@+id/resultFragment"
    android:name="com.betaccountant.fragment.ResultFragment"
    android:label="fragment_result"
    tools:layout="@layout/fragment_result" />

```

```

<fragment
    android:id="@+id/rulesFragment"
    android:name="com.betaccountant.fragment.RulesFragment"
    android:label="fragment_rules"
    tools:layout="@layout/fragment_rules" />

```

```

</navigation>

```

one_wrong_statement_task_dialog_layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/dialog_background"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="20dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/one_wrong_statement_task_text"
            android:textSize="20sp"
            android:gravity="center"
            android:textColor="@color/black"

```

```

        android:layout_marginBottom="20dp"/>
<RadioGroup
    android:id="@+id/oneWrongStatementTaskRadioGroup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<TextView
    android:id="@+id/answerBtn"
    style="@style/baseButtonStyle"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/answer" />

</LinearLayout>

</ScrollView>

```

story_dialog_layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/dialog_background"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">

<TextView
    android:id="@+id/storyDialogTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="@dimen/sp_20"
    android:textColor="@color/black"/>

<TextView
    android:id="@+id/continueStoryBtn"
    style="@style/baseButtonStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/continue" />
</LinearLayout>

```

story_dialog_layout_with_background_image.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"

```

```

android:layout_height="wrap_content"
android:background="@drawable/dialog_background"
android:gravity="center"
android:orientation="vertical"
android:padding="20dp">

```

```

<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

```

```

<TextView
    android:id="@+id/storyDialogTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:textSize="26sp"
    android:textColor="@color/black" />

```

```

<ImageView
    android:id="@+id/storyBackgroundImage"
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:layout_gravity="center"
    android:alpha="0.2" />

```

```

</FrameLayout>

```

```

<TextView
    android:id="@+id/continueStoryBtn"
    style="@style/baseButtonStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/continue" />

```

```

</LinearLayout>

```

story_dialog_layout_with_center_image.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/dialog_background"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="20dp">

    <TextView
        android:id="@+id/storyDialogTxt"
        android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:gravity="center"
    android:textColor="@color/black"
    android:textSize="@dimen/sp_20" />

```

```

<ImageView
    android:id="@+id/storyCenterImage"
    android:layout_width="300dp"
    android:layout_height="300dp" />

```

```

<TextView
    android:id="@+id/leftTimeTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@color/blue_1c"
    android:textStyle="bold"
    android:textSize="40sp"/>

```

```

<TextView
    android:id="@+id/continueStoryBtn"
    style="@style/baseButtonStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/answer_btn_bg"
    android:text="@string/continue" />

```

```
</LinearLayout>
```

video_dialog_layout.xml

```

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

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center">
    </VideoView>
</FrameLayout>

```

AccountantDB.kt

```

package com.betaccountant.db

import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

```

```

import com.betaccountant.db.dao.FactDao
import com.betaccountant.db.dao.ImageDao
import com.betaccountant.db.dao.StoryDao
import com.betaccountant.db.dao.TaskDao
import com.betaccountant.db.model.Fact
import com.betaccountant.db.model.Image
import com.betaccountant.db.model.Story
import com.betaccountant.db.model.Task

@Database(
    entities = [Task::class, Image::class, Story::class, Fact::class],
    version = 2
)
abstract class AccountantDB : RoomDatabase() {
    abstract fun taskDao(): TaskDao
    abstract fun imageDao(): ImageDao
    abstract fun storyDao(): StoryDao
    abstract fun factDao(): FactDao

    companion object {
        private const val DATABASE = "Accountant"

        // For Singleton instantiation
        @Volatile
        private var instance: AccountantDB? = null

        fun getInstance(context: Context): AccountantDB {
            return instance ?: synchronized(this) {
                instance
                ?: buildDatabase(context).also { instance = it }
            }
        }

        private fun buildDatabase(context: Context): AccountantDB {
            return Room.databaseBuilder(context, AccountantDB::class.java, DATABASE)
                .fallbackToDestructiveMigration()
                .build()
        }
    }
}

```

BoolQuestionDialog.kt

```

package com.betaccountant.dialog

import android.app.Dialog
import android.content.Context
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.os.Bundle
import android.view.View

```

```

import com.betaccountant.R
import kotlinx.android.synthetic.main.bool_question_dialog_layout.view.*
import kotlinx.android.synthetic.main.story_dialog_layout.view.*

class BoolQuestionDialog(
    context: Context,
    private val question: String?,
    private val answer: Boolean,
    private val answerListener: (isRightAnswer: Boolean) -> Unit
) : Dialog(context) {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val dialogView = layoutInflater.inflate(R.layout.bool_question_dialog_layout, null)
        dialogView.apply {
            questionDialogQuestionTxt.text = question
            questionDialogYesBtn.setOnClickListener(this@BoolQuestionDialog::handleClick)
            questionDialogNoBtn.setOnClickListener(this@BoolQuestionDialog::handleClick)
        }
        setContentView(dialogView)
        setCancelable(false)
        window?.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
    }

    private fun handleClick(view: View?) {
        when (view?.id) {
            R.id.questionDialogYesBtn -> {
                answerListener(answer)
            }
            R.id.questionDialogNoBtn -> {
                answerListener(!answer)
            }
        }
        this@BoolQuestionDialog.dismiss()
    }
}

```

EighthLevel.kt

```

package com.betaccountant.fragment

import android.content.res.Resources
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.graphics.drawable.Drawable
import android.media.MediaPlayer
import android.os.Bundle
import android.util.DisplayMetrics
import android.view.Gravity
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

```



```

import android.widget.ImageView
import android.widget.FrameLayout
import android.widget.TextView
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.dialog.GuessTermDialog
import com.betaccountant.dialog.StoryDialog
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.fragment_eighth_level.*
import kotlin.random.Random

class EighthLevel : Fragment() {

    private var imageList: ArrayList<FrameLayout>? = null
    private val captionsList = listOf(
        "начальник відділу:\nПекло О.О.",
        "головний бухгалтер:\nЗагребло Д.А.",
        "секретар:\nПомагайло Н.І.",
        "фахівець 1-ї категорії:\nХапайло О.Й.",
        "консультант:\nВдячний О.Т.",
        "головний економіст:\nЩедрий А.Д."
    )
    private val messagesList = listOf(
        "Цей спеціаліст зайнятий, спробуй інші двері.",
        "Обідня перерва.",
        "Сьогодні не приймальний день.",
        "Цей працівник у відпустці.",
        "Цей працівник на лікарняному.",
    ).toMutableList()
    private var doorCount = 6

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? = inflater.inflate(R.layout.fragment_eighth_level, container, false)

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        val termList = resources.getStringArray(R.array.terms).toMutableList()
        imageList = getImageList()
        imageList?.forEach { imageView ->
            imageGrid.addView(imageView)
            imageView.setOnClickListener {
                it.visibility = View.INVISIBLE
                if (isRightDoor()) {
                    MediaPlayer.create(context, R.raw.win_sound).start()
                    (activity as MainActivity).toolbar?.pauseTimeCounter()
                    StoryDialog(requireContext(), getString(R.string.mission_complete), {
                        (activity as MainActivity).showPromoVideoAndNavigateToLevel()
                    })
                }
            }
        }
    }
}

```

```

        (activity as MainActivity).navController?.navigate(R.id.resultFragment)
    }).show()
    return@setOnClickListener
} else {
    imageUrl?.remove(it)
    val randomTerm = termList.random()
    termList.remove(randomTerm)
    val randomText = messagesList.random()
    messagesList.remove(randomText)
    GuessTermDialog(requireContext(),
        "$randomText\nЩоб спробувати ще раз, відгадай бухгалтерський термін.",randomTerm) {}.show()
    }
    doorCount--
}
}
}

```

```
private fun isRightDoor() = Random.nextInt(1, doorCount) == 1
```

```
private fun getImageList(): ArrayList<FrameLayout> {
    val imageUrl = ArrayList<FrameLayout>()
    for (i in captionsList.indices) {
        imageUrl.add(
            createImageWithBottomCaption(
                ContextCompat.getDrawable(
                    requireContext(),
                    R.drawable.door
                ),
                captionsList[i]
            )
        )
    }
    return imageUrl
}

```

```
private fun createImageWithBottomCaption(image: Drawable?, text: String): FrameLayout {
    val displayMetrics = DisplayMetrics()
    activity?.windowManager?.defaultDisplay?.getMetrics(displayMetrics)
    val screenWidth = displayMetrics.widthPixels
    val width = screenWidth * 0.25
    val horizontalMargin = width * 0.1
    val height = width * 2.5
    return FrameLayout(requireContext()).apply {
        layoutParams = FrameLayout.LayoutParams(
            width.toInt(),
            height.toInt()
        ).apply {
            addView(createImage(image))
            addView(createCaption(text))
            setMargins(horizontalMargin.toInt(), 0, horizontalMargin.toInt(), 0)
        }
    }
}

```

```

}

private fun createImage(drawable: Drawable?): ImageView {
    return ImageView(context).apply {
        layoutParams = FrameLayout.LayoutParams(
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.MATCH_PARENT
        )
        setImageDrawable(drawable)
    }
}

private fun createCaption(text: String): TextView {
    val displayMetrics = DisplayMetrics()
    activity?.windowManager?.defaultDisplay?.getMetrics(displayMetrics)
    val screenWidth = displayMetrics.widthPixels
    val marginBottom = screenWidth * 0.05
    return TextView(context).apply {
        gravity = Gravity.CENTER
        layoutParams = FrameLayout.LayoutParams(
            ViewGroup.LayoutParams.WRAP_CONTENT,
            ViewGroup.LayoutParams.WRAP_CONTENT
        ).apply {
            gravity = Gravity.CENTER
            textSize = 6f
            setText(text)
            setBackgroundColor(resources.getColor(R.color.golden))
            setMargins(20, 0, 20, marginBottom.toInt())
            setPadding(20, 10, 20, 10)
            setTextColor(resources.getColor(R.color.brown))
        }
    }
}
}
}

```

Fact.kt

```

package com.betaccountant.db.model

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity
data class Fact(
    @PrimaryKey(autoGenerate = true)
    val factId: Int? = null,
    val levelNumber: Int,
    val text: String,
    val isTrue: Boolean,
    val groupId: Int = 0
)

```

FactDao.kt

```

package com.betaccountant.db.dao

import androidx.room.Dao
import androidx.room.Insert
import androidx.room.OnConflictStrategy
import androidx.room.Query
import com.betaccountant.db.model.Fact

@Dao
interface FactDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertAll(facts: List<Fact>)

    @Query("SELECT * FROM Fact WHERE levelNumber = :levelNumber")
    fun getFactsByLevel(levelNumber: Int): List<Fact>?
}

```

FifthLevel.kt

```

package com.betaccountant.fragment

import android.media.MediaPlayer
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.db.AccountantDB
import com.betaccountant.dialog.GameOverDialog
import com.betaccountant.dialog.OneWrongStatementTaskDialog
import com.betaccountant.dialog.StoryDialog
import com.betaccountant.enums.Level
import com.betaccountant.enums.Locations
import com.betaccountant.db.model.Fact
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.fragment_fifth_level.*
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

class FifthLevel : Fragment() {

    private var allFactsList: ArrayList<Fact>? = null
    private var balanceLocation: Locations? = null
    private var isBalanceLocation: Boolean = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
}

```

```

GlobalScope.launch {
    allFactsList = AccountantDB.getInstance(requireContext()).factDao().getFactsByLevel(5) as ArrayList<Fact>?
}
}

override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View? = inflater.inflate(R.layout.fragment_fifth_level, container, false)

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    StoryDialog(
        requireContext(),
        getString(R.string.fifth_level_start_story), {},
        ContextCompat.getDrawable(
            requireContext(),
            R.drawable.man_with_calculator
        ),
        true
    ).show()
    init()
}

private fun init() {
    balanceLocation = Locations.TAX_OFFICE
    administrationContainer.setOnClickListener(this::handleItemClick)
    bankBalanceContainer.setOnClickListener(this::handleItemClick)
    taxContainer.setOnClickListener {
        isBalanceLocation = balanceLocation == Locations.TAX_OFFICE
        handleItemClick(it)
    }
    shopContainer.setOnClickListener(this::handleItemClick)
}

private fun handleItemClick(view: View?) {
    if(allFactsList != null) {
        val randomGroupId = allFactsList!!.random().groupId
        val oneTaskFactsList = allFactsList!!.filter { it.groupId == randomGroupId }
        allFactsList = allFactsList!!.filterNot { it.groupId == randomGroupId } as ArrayList<Fact>
        OneWrongStatementTaskDialog(
            requireContext(),
            oneTaskFactsList,
        ) {
            handleAnswer(it)
            view?.visibility = View.INVISIBLE
        }.show()
    }
}

private fun handleAnswer(isRightAnswer: Boolean) {

```

```

    if (isRightAnswer) {
//      Toast.makeText(context, "right", Toast.LENGTH_LONG).show()
    } else {
        activity?.toolbar?.subtractOneLife()
        if(activity?.toolbar?.getRemainingLivesAmount() == 0){
            GameOverDialog(requireContext()){
                activity?.finish()
            }.show()
        }
    }
    showBalanceDialog()
}

private fun showBalanceDialog() {
    StoryDialog(
        requireContext(),
        getString(if (isBalanceLocation) R.string.help_betaccountant_reach_tax_office else R.string.balance_not_submitted),
        {
            if (isBalanceLocation) {
                MediaPlayer.create(context, R.raw.win_sound).start()
                (activity as MainActivity).navigateToLevel(Level.SIXTH)
            }
        }, if (isBalanceLocation) {
            ContextCompat.getDrawable(
                requireContext(),
                R.drawable.man_with_calculator
            )
        } else null
    ).show()
}
}

```

FourthLevel.kt

```

package com.betaccountant.fragment

import android.media.MediaPlayer
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.dialog.StoryDialog
import com.betaccountant.enums.Level
import kotlinx.android.synthetic.main.fragment_fourth_level.*

class FourthLevel : Fragment() {
    companion object{
        private const val ANSWER_CODE = "35240"
    }
}

```

```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? = inflater.inflate(R.layout.fragment_fourth_level, container, false)

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    finishGameButton.setOnClickListener {
        if (inputCode.text.toString() == ANSWER_CODE) {
            MediaPlayer.create(context, R.raw.win_sound).start()
            (activity as MainActivity).showPromoVideoAndNavigateToLevel(Level.FIFTH)
        } else {
            showWrongAnswerDialog()
        }
    }
}

private fun showWrongAnswerDialog() {
    StoryDialog(
        requireContext(),
        getString(R.string.third_level_wrong_answer), {}
    ).show()
}
}

```

GameOverDialog.kt

```

package com.betaccountant.dialog

import android.app.Dialog
import android.content.Context
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.os.Bundle
import com.betaccountant.R
import kotlinx.android.synthetic.main.game_over_dialog.view.*
import kotlinx.android.synthetic.main.guess_term_dialog_layout.view.*
import java.util.*
import kotlin.random.Random

class GameOverDialog (context: Context,
    private val onComplete: () -> Unit
): Dialog(context) {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val dialogView =
            inflater.inflate(R.layout.game_over_dialog, null)
        dialogView.apply {
            completeBtn.setOnClickListener {
                onComplete()
            }
        }
    }
}

```

```

    }
}
setContentView(dialogView)
setCancelable(false)
window?.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
}
}

```

GuessTermDialog.kt

```
package com.betaccountant.dialog
```

```

import android.app.Dialog
import android.content.Context
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.os.Bundle
import android.text.InputFilter
import com.betaccountant.R
import kotlinx.android.synthetic.main.guess_term_dialog_layout.view.*
import java.util.*
import kotlin.random.Random

```

```

class GuessTermDialog(
    context: Context,
    private val text: String,
    private val term: String,
    private val answerClick: () -> Unit
): Dialog(context) {

    private var wrongAnswerCount = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val dialogView =
            inflater.inflate(R.layout.guess_term_dialog_layout, null)
        dialogView.apply {
            guessTermDialogText.text = text
            guessTermDialogTxt.text = mixSymbols(term)
            answerTermInput.filters = arrayOf<InputFilter>(InputFilter.LengthFilter(term.length))
            guessTermAnswerBtn.setOnClickListener {
                val answer = answerTermInput.text.toString().toLowerCase(Locale.ROOT)
                val isRightAnswer = answer == term.toLowerCase(Locale.ROOT)
                if(isRightAnswer) {
                    answerClick()
                    this@GuessTermDialog.dismiss()
                } else {
                    wrongAnswerCount++
                    if(wrongAnswerCount <= 1){
                        showWrongAnswerDialog()
                    } else {
                        showRightAnswer()
                    }
                }
            }
        }
    }
}

```



```

        }
    }
}
setContentView(dialogView)
setCancelable(false)
window?.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
}

private fun mixSymbols(string: String): String {
    val mixedStr = StringBuilder(string)
    repeat(20) {
        val i1 = Random.nextInt(0, mixedStr.lastIndex)
        val i2 = Random.nextInt(0, mixedStr.lastIndex)
        val s1 = mixedStr[i1]
        val s2 = mixedStr[i2]
        mixedStr.setCharAt(i1, s2)
        mixedStr.setCharAt(i2, s1)
    }
    return mixedStr.toString()
}

private fun showWrongAnswerDialog() {
    StoryDialog(
        context,
        context.getString(R.string.wrong_answer), {}
    ).show()
}

private fun showRightAnswer() {
    StoryDialog(
        context,
        term, {}
    ).show()
}
}
}

```

Image.kt

```

package com.betaccountant.db.model

import androidx.room.Entity
import androidx.room.ForeignKey
import androidx.room.PrimaryKey

@Entity
data class Image(
    @PrimaryKey(autoGenerate = true)
    val imageId: Int,
    @ForeignKey(entity = Task::class, parentColumns = ["levelNumber"], childColumns = ["levelNumber"])
    val taskId: Int,
    val url: String,

```

```
)
```

ImageDao.kt

```
package com.betaccountant.db.dao

import androidx.room.Dao
import androidx.room.Insert
import androidx.room.OnConflictStrategy
import com.betaccountant.db.model.Image

@Dao
interface ImageDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertAll(images: List<Image>)
}
```

ImageQuestionLevelFragment.kt

```
package com.betaccountant.fragment

import android.media.MediaPlayer
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.MainActivity.Companion.FRAGMENT_LEVEL
import com.betaccountant.R
import com.betaccountant.db.AccountantDB
import com.betaccountant.dialog.StoryDialog
import com.betaccountant.enums.Level
import kotlinx.android.synthetic.main.fragment_image_question_level.*
import kotlinx.android.synthetic.main.fragment_sixth_level.*
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

class ImageQuestionLevelFragment : Fragment() {

    private var tryCount = 0
    private var currentLevel: Level? = null
    private var answer: String? = null

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_image_question_level, container, false)
    }
}
```

```

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    init()
}

private fun init() {
    currentLevel = (arguments?.getSerializable(FRAGMENT_LEVEL) as Level)
    when (currentLevel) {
        Level.FIRST -> setFirstFragment()
        Level.SECOND -> setSecondFragment()
        else -> Toast.makeText(context, getString(R.string.unexpected_error), Toast.LENGTH_LONG)
            .show()
    }
    answerBtn.setOnClickListener {
        if (answerInput.text.toString() == answer && currentLevel != null) {
            handleRightAnswer()
        } else {
            handleWrongAnswer()
        }
    }
}

private fun showWrongAnswerDialog(text: String) {
    StoryDialog(
        requireContext(),
        text, {}
    ).show()
}

private fun setFirstFragment() = GlobalScope.launch {
    val task = AccountantDB.getInstance(requireContext()).taskDao().getTaskByLevel(1)
    activity?.runOnUiThread {
        questionTxt.text = "${task?.taskDesc} ${task?.question}"
        answer = task?.rightAnswer
        questionImg.setImageResource(R.drawable.first_level_task)
    }
}

private fun setSecondFragment() = GlobalScope.launch {
    val task = AccountantDB.getInstance(requireContext()).taskDao().getTaskByLevel(2)
    activity?.runOnUiThread {
        questionTxt.text = "${task?.taskDesc} ${task?.question}"
        answer = task?.rightAnswer
        questionImg.setImageResource(R.drawable.second_level_task)
    }
}

private fun handleRightAnswer() {
    MediaPlayer.create(context, R.raw.win_sound).start()
    val nextLevel =
        Level.values()[currentLevel!!.value] // current value because indices starts from 0
    when (currentLevel) {

```

```

Level.FIRST -> {
    (activity as MainActivity).navigateToLevel(nextLevel)
}
Level.SECOND -> {
    handleSecondLevelRightAnswer(nextLevel)
}
else -> Toast.makeText(
    context,
    getString(R.string.unexpected_error),
    Toast.LENGTH_LONG
).show()
}
}

private fun handleSecondLevelRightAnswer(nextLevel: Level) {
    StoryDialog(
        requireContext(),
        getString(R.string.second_to_third_level_story),
        {
            (activity as MainActivity).showPromoVideoAndNavigateToLevel(
                nextLevel
            )
        },
        ContextCompat.getDrawable(
            requireContext(),
            R.drawable.man_with_calculator
        ),
        true
    ).show()
}

private fun handleWrongAnswer() {
    tryCount++
    when (tryCount) {
        3 -> showWrongAnswerDialog(
            getString(
                when (currentLevel) {
                    Level.FIRST -> R.string.continue_numbers_line
                    Level.SECOND -> R.string.remember_actives
                    else -> R.string.remember_actives
                }
            )
        )
    }
    5 -> showWrongAnswerDialog(
        getString(
            when (currentLevel) {
                Level.FIRST -> R.string.first_level_right_answer
                Level.SECOND -> R.string.second_level_right_answer
                else -> R.string.second_level_right_answer
            }
        )
    )
}
}

```

```

        else -> showWrongAnswerDialog(getString(R.string.wrong_answer))
    }
}

```

Level.kt

```
package com.betaccountant.enums
```

```

enum class Level(val value: Int) {
    FIRST(1),
    SECOND(2),
    THIRD(3),
    FOURTH(4),
    FIFTH(5),
    SIXTH(6),
    SEVENTH(7),
    EIGHTH(8)
}

```

Locations.kt

```
package com.betaccountant.enums
```

```
import kotlin.random.Random
```

```

enum class Locations(val value: Int) {
    BATH(1),
    SEA(2),
    BANK(3),
    BUSINESS_MEETING(4),
    LIBRARY(5),
    BANK_BALANCE(6),
    TAX_OFFICE(7),
    SHOP(8);

    companion object {
        fun getRandomLocation(level: Level): Locations? {
            return when (level) {
                Level.THIRD -> {
                    val value = Random.nextInt(BATH.value, BUSINESS_MEETING.value)
                    getByValue(value)
                }
                Level.FIFTH -> {
                    val value = Random.nextInt(LIBRARY.value, SHOP.value)
                    getByValue(value)
                }
                else -> {
                    val value = Random.nextInt(values().first().value, values().size)
                    getByValue(value)
                }
            }
        }
    }
}

```

```

    }

    private fun getByValue(value: Int): Locations? = values().firstOrNull { it.value == value }
}
}

```

MainActivity.kt

```

package com.betaccountant

import android.app.AlertDialog
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.app.AppCompatActivity
import androidx.navigation.NavController
import androidx.navigation.Navigation
import com.betaccountant.db.AccountantDB
import com.betaccountant.db.getFactsList
import com.betaccountant.db.getTasksFromStorage
import com.betaccountant.dialog.PromoVideoDialog
import com.betaccountant.enums.Level
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

class MainActivity : AppCompatActivity() {

    companion object {
        const val FRAGMENT_LEVEL = "FRAGMENT_LEVEL"
    }

    var navigationController: NavController? = null
    private set

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO)
        fillDB()
        setContentView(R.layout.activity_main)
        navigationController = Navigation.findNavController(this, R.id.fragment_container)
    }

    override fun onBackPressed() {
        if (navigationController?.currentDestination?.id == R.id.startFragment) {
            super.onBackPressed()
        } else {
            showExitDialog()
        }
    }

    internal fun navigateToLevel(nextLevel: Level) {
        when (nextLevel) {

```

```

Level.FIRST, Level.SECOND -> {
    val bundle = Bundle()
    bundle.putSerializable(FRAGMENT_LEVEL, nextLevel)
    navController?.navigate(R.id.firstLevelFragment, bundle)
}
Level.THIRD -> navController?.navigate(R.id.thirdLevelFragment)
Level.FOURTH -> navController?.navigate(R.id.fourthLevelFragment)
Level.FIFTH -> navController?.navigate(R.id.fivesLevelFragment)
Level.SIXTH -> navController?.navigate(R.id.sixthLevelFragment)
Level.SEVENTH -> navController?.navigate(R.id.seventhLevelFragment)
Level.EIGHTH -> navController?.navigate(R.id.eighthLevelFragment)
}
toolbar?.setLabel(getString(R.string.completed_levels, nextLevel.value))
}

private fun showExitDialog() {
    val alertBuilder = AlertDialog.Builder(this)
    alertBuilder.apply {
        setMessage(R.string.exit_dialog_message)
        setCancelable(true)
        setPositiveButton(R.string.yes) { _, _ -> finish() }
        setNegativeButton(R.string.no) { _, _ -> }
    }
    val alert = alertBuilder.create()
    alert.apply {
        setTitle(R.string.exit_dialog_title)
        show()
    }
}

internal fun showPromoVideoAndNavigateToLevel(nextLevel: Level? = null) {
    val promoVideoUrl = "android.resource://" + this.packageName + "/" +
        when (nextLevel) {
            Level.THIRD -> R.raw.promo_part_1
            Level.FIFTH -> R.raw.promo_part_2
            Level.SEVENTH -> R.raw.promo_part_3
            else -> R.raw.promo_part_4
        }
    val promoDialog = PromoVideoDialog(this, promoVideoUrl)
    promoDialog.show()
    toolbar?.pauseTimeCounter()
    promoDialog.start {
        toolbar?.resumeTimeCounter()
        if (nextLevel != null) {
            navigateToLevel(nextLevel)
        }
    }
}

private fun fillDB() {
    val db = AccountantDB.getInstance(this)
    GlobalScope.launch {

```

```

        db.clearAllTables()
        db.taskDao().insertAll(getTasksFromStorage())
        db.factDao().insertAll(getFactsList())
    }
}
}

```

OneWrongStatementTaskDialog.kt

```

package com.betaccountant.dialog

import android.app.Dialog
import android.content.Context
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.os.Bundle
import android.view.View
import android.view.ViewGroup
import android.widget.LinearLayout
import android.widget.RadioButton
import androidx.core.content.ContextCompat
import com.betaccountant.R
import com.betaccountant.db.model.Fact
import kotlinx.android.synthetic.main.one_wrong_statement_task_dialog_layout.*
import kotlinx.android.synthetic.main.one_wrong_statement_task_dialog_layout.view.*

class OneWrongStatementTaskDialog(
    context: Context,
    private val facts: List<Fact>,
    private val answerListener: (isRightAnswer: Boolean) -> Unit
) : Dialog(context) {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val dialogView = layoutInflater.inflate(R.layout.one_wrong_statement_task_dialog_layout, null)
        dialogView.apply {
            facts.forEach { oneWrongStatementTaskRadioGroup.addView(createRadioButtonView(it)) }
            answerBtn.setOnClickListener(this@OneWrongStatementTaskDialog::handleClick)
        }
        setContentView(dialogView)
        setCancelable(false)
        window?.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
    }

    private fun handleClick(view: View?) {
        val radioButton: RadioButton? =
            findViewById(oneWrongStatementTaskRadioGroup.checkedRadioButtonId)
        if (radioButton != null) {
            val answerFactNumber = oneWrongStatementTaskRadioGroup.indexOfChild(radioButton)
            answerListener(!facts[answerFactNumber].isTrue)
            this@OneWrongStatementTaskDialog.dismiss()
        }
    }
}

```



```

    }
}

private fun createRadioButtonView(fact: Fact): RadioButton {
    val radioButton = RadioButton(context)
    val params = LinearLayout.LayoutParams(
        ViewGroup.LayoutParams.MATCH_PARENT,
        ViewGroup.LayoutParams.WRAP_CONTENT
    )
    params.topMargin = 20
    return radioButton.apply {
        layoutParams = params
        text = fact.text
        background =
            ContextCompat.getDrawable(context, R.drawable.transparent_background_with_border)
        buttonTintList = ContextCompat.getColorStateList(context, R.color.border_grey)
        minHeight = 150
    }
}
}
}

```

PromoVideoDialog.kt

```

package com.betaccountant.dialog

import android.app.Dialog
import android.content.Context
import android.media.MediaPlayer.OnPreparedListener
import android.net.Uri
import android.os.Bundle
import android.view.Gravity
import android.view.WindowManager
import com.betaccountant.R
import kotlinx.android.synthetic.main.video_dialog_layout.*
import kotlinx.android.synthetic.main.video_dialog_layout.view.*

class PromoVideoDialog(context: Context, private val promoVideoUrl: String?) : Dialog(context) {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val dialogView = layoutInflater.inflate(R.layout.video_dialog_layout, null)
        val uri = Uri.parse(promoVideoUrl)
        dialogView.videoView.setVideoURI(uri)
        setContentView(dialogView)
        setCancelable(false)
        window?.setLayout(
            WindowManager.LayoutParams.MATCH_PARENT,
            WindowManager.LayoutParams.WRAP_CONTENT
        )
        window?.setGravity(Gravity.CENTER)
    }
}

```

```

        videoView.setOnPreparedListener { mp ->
            mp.setOnSeekCompleteListener {
                videoView.start()
            }
        }
    }
}

fun start(completeListener: () -> Unit) {
    videoView.start()
    videoView.setOnCompleteListener {
        completeListener.invoke()
        this@PromoVideoDialog.dismiss()
    }
}
}

```

ResultFragment.kt

```
package com.betaccountant.fragment
```

```

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.fragment_result.*

```

```
class ResultFragment : Fragment() {
```

```
    companion object {
```

```
        private const val INSTAGRAM_LINK =
```

```
            "https://www.instagram.com/sumdu_kafedra_boo/"
```

```
        private const val FACEBOOK_LINK = "https://www.facebook.com/groups/buchuchet.uabs.sumdu"
```

```
        private const val SITE_LINK = "https://op.biem.sumdu.edu.ua"
```

```
        private const val EMAIL_LINK = "mailto:n.ovcharova@uabs.sumdu.edu.ua"
```

```
        private const val LOCATION_LINK =
```

```

"https://www.google.com/maps/place/%D0%B2%D1%83%D0%BB%D0%B8%D1%86%D1%8F+%D0%9F%D0%B5%D1%82%D1%80%D0%BE%D0%B
F%D0%B0%D0%B2%D0%BB%D1%96%D0%B2%D1%81%D1%8C%D0%BA%D0%B0,+57,+%D0%A1%D1%83%D0%BC%D0%B8,+%D0%A1%D1
%83%D0%BC%D1%81%D1%8C%D0%BA%D0%B0+%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C,+40000/@50.9051338,34.793
3402,17z/data=!3m1!4m5!3m4!1s0x4129018c0bbb6515:0x6530719c9fe77b62!8m2!3d50.9051338!4d34.7955289"

```

```
        const val IS_FROM_START = "IS_FROM_START"
```

```
        private const val MOBILE_PHONE = "0996674401"
```

```
        private const val STATIC_PHONE = "0542665092"
```

```
    }
```

```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? = inflater.inflate(R.layout.fragment_result, container, false)

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    val isFromStart = arguments?.getBoolean(IS_FROM_START) ?: false
    if(!isFromStart) {
        (activity as MainActivity).toolbar?.setLabel(getString(R.string.inside_app_name))
        (activity as MainActivity).toolbar?.hideLives()
        (activity as MainActivity).toolbar?.hideTimer()
        val resultTime = (activity as MainActivity).toolbar?.getFormattedTime()
        resultTxt.visibility = View.VISIBLE
        resultTimeTxt.visibility = View.VISIBLE
        resultHint.visibility = View.VISIBLE
        resultTimeTxt.text = resultTime
    } else {
        resultTxt.visibility = View.GONE
        resultTimeTxt.visibility = View.GONE
        resultHint.visibility = View.GONE
        finishGameButton.visibility = View.GONE
    }

    finishGameButton.setOnClickListener {
        activity?.finish()
    }
    instagramLink.setOnClickListener {
        openUrl(INSTAGRAM_LINK)
    }
    facebookLink.setOnClickListener {
        openUrl(FACEBOOK_LINK)
    }
    siteLink.setOnClickListener {
        openUrl(SITE_LINK)
    }
    mobilePhoneLink.setOnClickListener {
        callIntent(MOBILE_PHONE)
    }
    staticPhoneLink.setOnClickListener {
        callIntent(STATIC_PHONE)
    }
    emailLink.setOnClickListener {
        openUrl(EMAIL_LINK)
    }
    locationLink.setOnClickListener {
        openUrl(LOCATION_LINK)
    }
}

private fun openUrl(url: String) {
    val intent = Intent(Intent.ACTION_VIEW)

```

```

        intent.data = Uri.parse(url)
        startActivity(intent)
    }

    private fun callIntent(phone: String) {
        val phoneNumber = "tel:$phone"
        val intent = Intent(Intent.ACTION_DIAL)
        intent.data = Uri.parse(phoneNumber)
        startActivity(intent)
    }
}

```

RulesFragment.kt

```

package com.betaccountant.fragment

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.enums.Level
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.fragment_rules.*

class RulesFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? = inflater.inflate(R.layout.fragment_rules, container, false)

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        continueRulesBtn.setOnClickListener {
            activity?.toolbar?.timeCounterEnabled(true)
            activity?.toolbar?.livesCounterEnabled(true)
            activity?.toolbar?.resetTimeCounter()
            activity?.toolbar?.startTimeCounter()
            (activity as MainActivity).navigateToLevel(Level.SIXTH)
        }
    }
}

```

SeventhLevel.kt

```

package com.betaccountant.fragment

import android.media.MediaPlayer

```

```

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.dialog.StoryDialog
import com.betaccountant.enums.Level
import kotlinx.android.synthetic.main.fragment_seventh_level.*
import java.util.*

class SeventhLevel : Fragment() {

    companion object {
        private const val LEVEL_ANSWER = "звіт"
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? = inflater.inflate(R.layout.fragment_seventh_level, container, false)

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        seventhLvlAnswerBtn.setOnClickListener {
            if (isRightAnswer()) {
                StoryDialog(
                    requireContext(),
                    "Вітаємо!\nБаланс потрапив до податкової!",
                    {
                        MediaPlayer.create(context, R.raw.win_sound).start()
                        (activity as MainActivity).navigateToLevel(Level.EIGHTH)
                    },
                    ContextCompat.getDrawable(requireContext(), R.drawable.docs_and_coins),
                    true
                ).show()
            } else {
                showWrongAnswerDialog()
            }
        }
    }

    private fun showWrongAnswerDialog() {
        StoryDialog(
            requireContext(),
            getString(R.string.wrong_answer), {}
        ).show()
    }

    private fun isRightAnswer() =

```

```

    seventhLvlAnswerInput.text.toString().toLowerCase(Locale.ROOT) == LEVEL_ANSWER
}

```

SixthLevel.kt

```

package com.betaccountant.fragment

import android.media.MediaPlayer
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.CheckBox
import android.widget.LinearLayout
import android.widget.Toast
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.db.AccountantDB
import com.betaccountant.db.model.Fact
import com.betaccountant.dialog.StoryDialog
import com.betaccountant.enums.Level
import kotlinx.android.synthetic.main.fragment_sixth_level.*
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

class SixthLevel : Fragment() {

    private var factsList: List<Fact>? = null
    private var checkBoxFactsList: ArrayList<CheckBox>? = null
    private val BUS_ADDITIONAL_TIME = 7L
    private val BIKE_ADDITIONAL_TIME = 5L
    private val CAR_ADDITIONAL_TIME = 3L

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? = inflater.inflate(R.layout.fragment_sixth_level, container, false)

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        sixLVLAnswerBtn.setOnClickListener {
            handleAnswerClick()
        }
        GlobalScope.launch {
            factsList = AccountantDB.getInstance(requireContext()).factDao()
                .getFactsByLevel(6) as ArrayList<Fact>?
            activity?.runOnUiThread {
                checkBoxFactsList = ArrayList()
                factsList?.forEach { checkBoxFactsList?.add(createCheckBox(it)) }
            }
        }
    }
}

```

```

        checkBoxFactsList?.forEach { factsLayout.addView(it) }
    }
}

private fun handleAnswerClick() {
    if (getCheckedItemsCount() != 3) {
        Toast.makeText(context, getString(R.string.choose_only_three_facts), Toast.LENGTH_LONG)
            .show()
    } else {
        MediaPlayer.create(context, R.raw.win_sound).start()
        val dialog = StoryDialog(
            requireContext(), getString(R.string.go_to_tax), {},
            ContextCompat.getDrawable(
                requireContext(),
                when (getRightAnswersCount()) {
                    2 -> R.drawable.bike
                    3 -> R.drawable.car
                    else -> R.drawable.tralleybus
                }
            )
        )
        dialog.show()
        dialog.setContinueButtonVisibility(false)
        dialog.dismissAfter(
            requireActivity(),
            when (getRightAnswersCount()) {
                2 -> BIKE_ADDITIONAL_TIME
                3 -> CAR_ADDITIONAL_TIME
                else -> BUS_ADDITIONAL_TIME
            }
        )
        dialog.setOnDismissListener {
            (activity as MainActivity).showPromoVideoAndNavigateToLevel(Level.SEVENTH)
        }
    }
}

private fun getRightAnswersCount(): Int {
    var rightAnswersCount = 0
    checkBoxFactsList?.forEach { checkBox ->
        if (checkBox.isChecked) {
            factsList?.forEach { fact ->
                if (fact.text == checkBox.text && fact.isTrue) {
                    rightAnswersCount++
                }
            }
        }
    }
    return rightAnswersCount
}

```

```

private fun getCheckedItemsCount() = checkBoxFactsList?.filter { it.isChecked }?.size ?: 0

private fun createCheckBox(fact: Fact) = CheckBox(context)
    .apply {
        layoutParams = LinearLayout.LayoutParams(
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.WRAP_CONTENT
        ).apply {
            topMargin = 20
            setPadding(5, 5, 5, 5)
        }
        text = fact.text
        background =
            ContextCompat.getDrawable(
                context,
                R.drawable.transparent_background_with_border
            )
        buttonTintList = ContextCompat.getColorStateList(context, R.color.border_grey)
    }
}

```

StartFragment.kt

```

package com.betaccountant.fragment

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.enums.Level
import com.betaccountant.fragment.ResultFragment.Companion.IS_FROM_START
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.fragment_start.*

class StartFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_start, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        startBtn.setOnClickListener {
            activity?.toolbar?.timeCounterEnabled(true)
            activity?.toolbar?.livesCounterEnabled(true)
        }
    }
}

```



```

        activity?.toolbar?.resetTimeCounter()
        activity?.toolbar?.startTimeCounter()
        activity?.toolbar?.infoButton?.visibility = View.GONE
        (activity as MainActivity).navigateToLevel(Level.FIRST)
    }
    activity?.toolbar?.setLabel(R.string.inside_app_name)
    activity?.toolbar?.infoButton?.setOnClickListener {
        val bundle = Bundle()
        bundle.putBoolean(IS_FROM_START, true)
        (activity as MainActivity).navController?.navigate(R.id.resultFragment, bundle)
    }
}
}
}

```

Story.kt

```

package com.betaccountant.db.model

import androidx.room.Entity
import androidx.room.ForeignKey
import androidx.room.PrimaryKey

@Entity
data class Story(
    @PrimaryKey(autoGenerate = true)
    val storyId: Int,
    @ForeignKey(entity = Task::class, parentColumns = ["levelNumber"], childColumns = ["levelNumber"])
    val taskId: Int,
    val storyTxt: String,
    val rightAnswer: String?,
)

```

StoryDao.kt

```

package com.betaccountant.db.dao

import androidx.room.Dao
import androidx.room.Insert
import androidx.room.OnConflictStrategy
import com.betaccountant.db.model.Story

@Dao
interface StoryDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertAll(stories: List<Story>)

}

```

StoryDialog.kt

```

package com.betaccountant.dialog

```

```

import android.app.Activity
import android.app.Dialog
import android.content.Context
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.graphics.drawable.Drawable
import android.os.Bundle
import android.view.View
import com.betaccountant.R
import kotlinx.android.synthetic.main.story_dialog_layout.view.*
import kotlinx.android.synthetic.main.story_dialog_layout.view.continueStoryBtn
import kotlinx.android.synthetic.main.story_dialog_layout.view.storyDialogTxt
import kotlinx.android.synthetic.main.story_dialog_layout_with_background_image.view.*
import kotlinx.android.synthetic.main.story_dialog_layout_with_center_image.*
import kotlinx.android.synthetic.main.story_dialog_layout_with_center_image.view.*
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import java.util.*
import java.util.concurrent.TimeUnit

class StoryDialog(
    context: Context,
    private val message: String,
    private val continueClick: () -> Unit,
    private val image: Drawable? = null,
    private val isBackgroundImage: Boolean = false
): Dialog(context) {

    private lateinit var dialogView: View

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        dialogView = LayoutInflater.inflate(
            when {
                isBackgroundImage -> {
                    R.layout.story_dialog_layout_with_background_image
                }
                image != null -> {
                    R.layout.story_dialog_layout_with_center_image
                }
                else -> {
                    R.layout.story_dialog_layout
                }
            }, null
        )
        dialogView.apply {
            storyDialogTxt.text = message
            if (isBackgroundImage) {
                storyBackgroundImage.setImageDrawable(image)
            } else if (image != null) {

```

```

        storyCenterImage.setImageDrawable(image)
    }
    continueStoryBtn.setOnClickListener {
        continueClick.invoke()
        this@StoryDialog.dismiss()
    }
}
setContentView(dialogView)
setCancelable(false)
window?.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
}

fun setContinueButtonVisibility(isVisible: Boolean) {
    dialogView.continueStoryBtn.visibility = if (isVisible) View.VISIBLE else View.GONE
}

fun dismissAfter(activity: Activity, seconds: Long) {
    var leftTime = seconds
    GlobalScope.launch {
        while (leftTime != 0L) {
            activity.runOnUiThread {
                dialogView.leftTimeTxt.text = leftTime.toString()
            }
            delay(1000)
            leftTime--
        }
        activity.runOnUiThread {
            this@StoryDialog.dismiss()
        }
    }
}
}
}
}

```

Task.kt

```

package com.betaccountant.db.model

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity
data class Task(
    @PrimaryKey(autoGenerate = true)
    val taskId: Int = 0,
    val levelNumber: Int,
    val question: String?,
    val rightAnswer: String? = null,
    val taskDesc: String? = null,
    val isAddTask: Boolean? = false
)

```

TaskDao.kt

```

package com.betaccountant.db.dao

import androidx.room.Dao
import androidx.room.Insert
import androidx.room.OnConflictStrategy
import androidx.room.Query
import com.betaccountant.db.model.Fact
import com.betaccountant.db.model.Task

@Dao
interface TaskDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertAll(tasks: List<Task>)

    @Query("SELECT * FROM Task WHERE levelNumber = :levelNumber AND isAddTask = 0")
    fun getTaskByLevel(levelNumber: Int): Task?

    @Query("SELECT * FROM Task WHERE levelNumber = :levelNumber AND isAddTask = 1")
    fun getAdditionalTaskByLevel(levelNumber: Int): Task?
}

```

ThirdLevel.kt

```

package com.betaccountant.fragment

import android.media.MediaPlayer
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.core.content.ContextCompat
import androidx.fragment.app.Fragment
import com.betaccountant.MainActivity
import com.betaccountant.R
import com.betaccountant.dialog.BoolQuestionDialog
import com.betaccountant.dialog.GameOverDialog
import com.betaccountant.dialog.StoryDialog
import com.betaccountant.enums.Level
import com.betaccountant.enums.Locations
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.fragment_third_level.*

class ThirdLevel : Fragment() {

    private lateinit var questionsWithAnswers: MutableMap<String, String>
    private var directorLocation: Locations? = null
    private var isDirectorLocation: Boolean = false

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,

```

```

    savedInstanceState: Bundle?
): View? = inflater.inflate(R.layout.fragment_third_level, container, false)

override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    super.onCreateView(view, savedInstanceState)
    StoryDialog(
        requireContext(),
        getString(R.string.third_level_start_story), {},
        ContextCompat.getDrawable(
            requireContext(),
            R.drawable.man_with_calculator
        ),
        true
    ).show()
    init()
}

private fun init() {
    directorLocation = Locations.getRandomLocation(Level.THIRD)
    val questions = resources.getStringArray(R.array.third_level_questions)
    val answers = resources.getStringArray(R.array.third_level_answers)
    questionsWithAnswers = questions.zip(answers).toMap().toMutableMap()
    bathContainer.setOnClickListener {
        isDirectorLocation = directorLocation == Locations.BATH
        handleItemClick(it)
    }
    seaContainer.setOnClickListener {
        isDirectorLocation = directorLocation == Locations.SEA
        handleItemClick(it)
    }
    bankContainer.setOnClickListener {
        isDirectorLocation = directorLocation == Locations.BANK
        handleItemClick(it)
    }
    businessMeetingContainer.setOnClickListener {
        isDirectorLocation = directorLocation == Locations.BUSINESS_MEETING
        handleItemClick(it)
    }
}

private fun handleItemClick(view: View?) {
    val question = questionsWithAnswers.keys.random()
    val answer = getBoolAnswerByQuestion(question)
    questionsWithAnswers.remove(question)
    BoolQuestionDialog(
        requireContext(),
        question,
        answer
    ){
        handleAnswer(it)
        view?.visibility = View.INVISIBLE
    }.show()
}

```

```

}

private fun handleAnswer(isRightAnswer: Boolean) {
    if (isRightAnswer) {
//      Toast.makeText(context, "right", Toast.LENGTH_LONG).show()
    } else {
        activity?.toolbar?.subtractOneLife()
        if(activity?.toolbar?.getRemainingLivesAmount() == 0){
            GameOverDialog(requireContext()){
                activity?.finish()
            }.show()
        }
    }
}

showDirectorDialog()
}

private fun showDirectorDialog() {
    StoryDialog(
        requireContext(),
        getString(if (isDirectorLocation) R.string.director_found else R.string.director_not_found),
        {
            if (isDirectorLocation) {
                MediaPlayer.create(context, R.raw.win_sound).start()
                (activity as MainActivity).navigateToLevel(Level.FOURTH)
            }
        }
    ).show()
}

private fun getBoolAnswerByQuestion(question: String): Boolean =
    questionsWithAnswers[question] == getString(R.string.yes)
}

```

ToolbarView.kt

```

package com.betaccountant.view

import android.content.Context
import android.graphics.drawable.Drawable
import android.os.SystemClock
import android.util.AttributeSet
import android.view.View
import android.widget.Chronometer
import android.widget.ImageView
import android.widget.LinearLayout
import android.widget.TextView
import com.betaccountant.R
import kotlinx.android.synthetic.main.layout_toolbar_view.view.*
import java.text.DateFormat
import java.text.SimpleDateFormat
import java.util.*

```

```

class ToolbarView @JvmOverloads constructor(
    context: Context,
    attrs: AttributeSet,
    defStyleAttr: Int = 0
): LinearLayout(context, attrs, defStyleAttr) {

    private var attrs: AttributeSet? = attrs
    private var label: TextView? = null
    private var timeCounter: Chronometer? = null
    private var livesCounter: LinearLayout? = null
    private var maxLivesAmount: Int? = 0
    private var remainingLivesAmount: Int? = 0
    private var remainingLifeImg: Drawable? = null
    private var usedLifeImg: Drawable? = null
    private var timeCounterStoppedTime: Long = 0
    val infoButton: ImageView

    init {
        val view = View.inflate(context, R.layout.layout_toolbar_view, this)
        label = view.toolbarLabel
        timeCounter = view.toolbarTimeCounter
        livesCounter = view.toolbarLivesCounter
        infoButton = view.infoButton
        readParams(context)
    }

    fun startTimeCounter() {
        timeCounter?.base = SystemClock.elapsedRealtime() - timeCounterStoppedTime
        timeCounter?.start()
    }

    fun pauseTimeCounter() {
        timeCounterStoppedTime = SystemClock.elapsedRealtime() - (timeCounter?.base ?: 0)
        timeCounter?.stop()
    }

    fun resumeTimeCounter() {
        startTimeCounter()
    }

    fun resetTimeCounter() {
        timeCounter?.base = SystemClock.elapsedRealtime()
        timeCounterStoppedTime = 0
    }

    fun timeCounterEnabled(enabled: Boolean) {
        timeCounter?.visibility = if (enabled) View.VISIBLE else View.GONE
    }

    fun getTime(): Long = timeCounterStoppedTime

    fun getFormattedTime(): String {

```

```

    val simple: DateFormat = SimpleDateFormat("mm:ss")
    val result = Date(getTime())
    return simple.format(result)
}

fun setLabel(labelTxt: String?) {
    label?.text = labelTxt
}

fun setLabel(stringId: Int) {
    label?.text = resources.getString(stringId)
}

fun getLabel(): String {
    return label?.text.toString()
}

fun livesCounterEnabled(enabled: Boolean) {
    livesCounter?.visibility = if (enabled) View.VISIBLE else View.GONE
}

fun getMaxLivesAmount(): Int {
    return maxLivesAmount!!
}

fun getRemainingLivesAmount(): Int {
    return remainingLivesAmount!!
}

fun subtractOneLife() {
    setRemainingLivesAmount(getRemainingLivesAmount().dec())
}

private fun setRemainingLivesAmount(remainingLivesAmount: Int?) {
    this.remainingLivesAmount = remainingLivesAmount
    toolbarLivesCounter.removeAllViewsInLayout()
    for (i in 1..maxLivesAmount!!) {
        toolbarLivesCounter.addView(createLifeView(if (i <= remainingLivesAmount!!) remainingLifeImg else usedLifeImg))
    }
}

fun hideLives() {
    toolbarLivesCounter.visibility = View.GONE
}

fun hideTimer(){
    toolbarTimeCounter.visibility = View.GONE
}

private fun readParams(context: Context) {
    val attributes = context.theme.obtainStyledAttributes(
        attrs,

```



```

R.styleable.ToolbarView,
0, 0
)
try {
setLabel(attributes.getString(R.styleable.ToolbarView_label))
timeCounterEnabled(
attributes.getBoolean(
R.styleable.ToolbarView_time_counter_enabled,
false
)
)
livesCounterEnabled(
attributes.getBoolean(
R.styleable.ToolbarView_lives_counter_enabled,
false
)
)
maxLivesAmount = attributes.getInteger(R.styleable.ToolbarView_max_lives_amount, 0)
remainingLivesAmount =
attributes.getInteger(R.styleable.ToolbarView_remaining_lives_amount, 0)
remainingLifeImg = attributes.getDrawable(R.styleable.ToolbarView_remaining_life_img)
usedLifeImg = attributes.getDrawable(R.styleable.ToolbarView_used_life_img)
setRemainingLivesAmount(remainingLivesAmount!!)
} finally {
attributes.recycle()
}
}

private fun createLifeView(lifeImage: Drawable?): ImageView {
val lifeView = ImageView(context)
val params = LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT, 1F)
params.marginStart =
context.resources.getDimensionPixelSize(R.dimen.toolbar_life_views_margin)
lifeView.layoutParams = params
lifeView.setImageDrawable(lifeImage)
return lifeView
}
}

```

ДОДАТОК Д

АКТ ВПРОВАДЖЕННЯ

ЗАТВЕРДЖУЮ

Завідувачка кафедри бухгалтерського
обліку та оподаткування
ННІ БіЕМ
Сумського державного університету
Серпенінова Ю.С.
«*24*» *травня* 2021 р.

Акт


Впровадження результатів дипломного проекту
студента Сумського державного університету
Сича Євгенія Владиславовича

Даним актом підтверджується, що результати роботи студента Сича Є.В. на тему «Мобільний ігровий додаток для абітурієнтів кафедри бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ» впроваджено в профорієнтаційну роботу кафедри бухгалтерського обліку та оподаткування.

Даний мобільний додаток містить інформацію про кафедру бухгалтерського обліку та оподаткування ННІ БіЕМ СумДУ у ігровому форматі. Гра складається з 8-ми рівнів на яких розташовані завдання, переможцем стає той учасник, що швидше за всіх подолає всі рівні. Між рівнями додатку є вбудована реклама щодо переваг навчання на спеціальності «Облік і оподаткування».

Впровадження мобільного додатку в профорієнтаційні заходи кафедри допомагає більш обширно та цікаво доносити інформацію до потенційних абітурієнтів, залучати їх до соціальних мереж кафедри.

Завідувачка кафедри бухгалтерського
обліку та оподаткування
ННІ БіЕМ
Сумського державного університету



Серпенінова Ю.С.