

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Android додаток для полегшення навчання та сприяння активного
залучення студентів»**

Виконав студент групи ІПЗм-22-1
НІКОЛАЙЧУК Максим Юрійович

Керівник роботи:
КРАВЧЕНКО Світлана Миколаївна

Рецензент:
ВЛАСЕНКО Олег Васильович

Житомир – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«18» вересня 2023 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **НІКОЛАЙЧУК Максим Юрійович**

Керівник роботи: **КРАВЧЕНКО Світлана Миколаївна**

Тема роботи: **«Android додаток для полегшення навчання та сприяння активного залучення студентів»**,

затверджена Наказом закладу вищої освіти від **«18» вересня 2023 р. № 479/с**

Вихідні дані для роботи: Предметом дослідження є методи, засоби та теорії, що полегшують навчання та заохочують студентів. Об'єктом дослідження є додаток для полегшення навчання та сприяння активного залучення студентів.

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Кравченко С.М.	22.09.2023р.	22.09.2023р.
2	Кравченко С.М.	22.09.2023р.	22.09.2023р.
3	Кравченко С.М.	22.09.2023р.	22.09.2023р.

РЕФЕРАТ

Кваліфікаційна робота присвячена розробці Android-додатку для полегшення навчання та сприяння активного залучення студентів.

Кваліфікаційна робота складається з прототипу програмного комплексу, якій побудований використовуючи архітектурний стиль MVVM та пояснювальної записки з детальним теоритичним дослідженням проблематики платформи Android та навчання.

Пояснювальна записка складається з вступу, трьох розділів, висновків та списку використаної літератури, що містить 20 найменувань. Також в роботі наведено 41 рисунок/ілюстрація та 3 додатки.

Кваліфікаційна робота складається з трьох розділів.

В першому розділі було розкрито предметну область використання мобільних технологій, а саме платформи Android у навчальному процесі. Проведено аналіз можливих рішень проблеми та постановлено задачі дослідження.

В другому розділі було детально проаналізовано існуючі аналоги мобільних засобів, які полегшують навчання, було порівняно їх переваги й недоліки. Було обрано необхідні засоби та технології, що полегшать та прискорять процес розробки

В третьому розділі було розгорнуто та протестовано модель у вигляді Android-додатку. У ході тестування всі виявлені недоліки було усунуто та внесено необхідні правки. Також було успішно оцінено та проаналізовано ефективність роботи додатку.

Висновок містить в собі підсумки проведеного дослідження.

Ключові слова: ANDROID, MVVM, РОЗРОБКА, ДОСЛІДЖЕННЯ, МЕТОДИКА, ЗАСОБИ, ТЕХНОЛОГІЇ, АНАЛОГИ, ЕФЕКТИВНІСТЬ.

ABSTRACT

The qualification work is dedicated to the development of an Android application aimed at facilitating learning and promoting active student engagement. The qualification work consists of a prototype of a software complex built using the MVVM architectural style and an explanatory note with a detailed theoretical investigation of the issues related to the Android platform and education.

The explanatory note includes an introduction, three chapters, conclusions, and a list of references containing 20 entries. The work also includes 38 figures/illustrations and 3 appendices. The qualification work is divided into three chapters.

In the first chapter, the subject area of mobile technology usage in the educational process, specifically the Android platform, was disclosed. An analysis of possible solutions to the problem was conducted, and research tasks were formulated.

The second chapter provided a detailed analysis of existing analogs of mobile tools that facilitate learning, comparing their advantages and disadvantages. The necessary tools and technologies were selected to facilitate and expedite the development process.

The third chapter deployed and tested the model in the form of an Android application. During testing, all identified shortcomings were addressed, and necessary modifications were made. The efficiency of the application was successfully evaluated and analyzed.

The conclusion contains the summarization of the conducted research.

Keywords: ANDROID, MVVM, DEVELOPMENT, RESEARCH, METHODOLOGY, TOOLS, TECHNOLOGIES, ANALOGS, EFFICIENCY.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

MVVM – Model View ViewModel

MVP – Model View Presenter

MVC – Model View Controller

API – Application Programming Interface

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1. Визначення напрямків предметної області.....	11
1.2. Аналіз існуючих рішень	15
1.3. Основні моделі, методи і підходи гейміфікації	24
1.4. Постановка задачі дослідження	31
Висновки до першого розділу	32
РОЗДІЛ 2. ПРОЕКТУВАННЯ ANDROID-ДОДАТКУ ДЛЯ ПОЛЕГШЕННЯ НАВЧАННЯ ТА СПРИЯННЯ АКТИВНОГО ЗАЛУЧЕННЯ СТУДЕНТІВ.....	33
2.1. Аналіз аналогів додатку	33
2.2. Вибір засобів та технологій розробки	41
2.3. Проектування внутрішньої структури додатку	48
Висновки до другого розділу.....	53
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ.....	55
3.1. Імплементация гейміфікації у додатку	55
3.2. Розгортання та тестування моделі у вигляді Android-додатку	56
3.3. Оцінка та аналіз ефективності роботи додатку	67
Висновки до третього розділу	68
ВИСНОВОК	69
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71
ДОДАТКИ.....	73

ВСТУП

Актуальність теми. Сучасний світ переживає стрімкий розвиток інформаційних технологій, який необхідно враховувати в усіх сферах життя, зокрема в системі вищої освіти. Освіта стає все більш інтерактивною та орієнтованою на індивідуальні потреби студентів. У цьому контексті, розробка Android-додатка для полегшення навчання та активного залучення студентів стає надзвичайно актуальною і важливою задачею.

1. Розвиток Мобільних Технологій: Зростання кількості користувачів мобільних пристроїв, таких як смартфони та планшети, створює потребу в ефективних навчальних інструментах, що відповідають сучасним стандартам та вимогам.

2. Зміна Підходів до Навчання: Сучасні підходи до навчання надають акцент на інтерактивність та індивідуалізацію. Android-додаток, орієнтований на полегшення навчання, може стати потужним інструментом для впровадження цих змін.

3. Гейміфікація в Освіті: Використання гейміфікації у навчальних процесах стає все більш популярним. Розробка додатка, який інтегрує гейміфікаційні елементи, може стимулювати активність та інтерес студентів до навчання.

4. Потреба в Інноваційних Засобах Навчання: Сучасні студенти очікують від університетів та вищих навчальних закладів інноваційних підходів до навчання. Розробка Android-додатка відповідає цій потребі та дозволяє навчальним закладам залишатися конкурентоспроможними.

5. Пандемія, Дистанційне Навчання та Військовий Стан: Глобальні виклики, такі як пандемія COVID-19 та військова агресія РФ, підкреслили важливість та необхідність технологічних засобів для дистанційного навчання. Android-додаток може стати зручним і ефективним інструментом для віддалених навчальних процесів.

6. Залучення та Мотивація Студентів: Студенти стають більш зацікавленими в навчанні, якщо їхнє навчання є захопливим та інтерактивним.

Android-додаток, який поєднує в собі елементи гри та навчання, може забезпечити стимулюючий досвід для користувачів.

Усі ці фактори свідчать про важливість розробки та впровадження Android-додатка для полегшення навчання та активного залучення студентів як ефективного інструменту у сучасному вищому навчальному процесі.

Протягом дослідження у рамках наукової роботи буде розглянуто основні наукові та соціальні способи спрощення навчання та підвищення залучення студентів до освітнього процесу з врахуванням сучасних тенденцій та реалій. Аналіз отриманих результатів та порівняння їх з іншими порталами дозволить оцінити ефективність їх застосування у навчанні студентів.

Дана магістерська робота присвячена розробці та аналізу способів розробки додатку на платформу Android, що розрахований на полегшення процесів навчання та активного заохочення студентів до участі в навчальному процесі.

Ця наукова робота має на меті полегшити навчання у 21-му віці технологій. У майбутньому, отриманий додаток може стати фундаментом для розробки більш комплексної системи онлайн навчальних закладів, що винесуть процеси навчання на новий технологічний рівень.

Четверта промислова революція. У нашому світі настала так звана четверта промислова революція, що призвела до стрімкого розвитку технологій по всьому світу, а з цим і кардинально почала змінювати буденне життя людей по всьому земному шару. Наприклад, наразі майже кожна людина у світі має під рукою телефон, що дозволяє їй бути завжди на зв'язку, навчатись на ходу або навіть виконувати якусь роботу. Отже з приходом нових технологій і з'являються нові зручні інструменти, такі як додатки для навчання та розваг.

Метою даної роботи є розробка та аналіз Android додатку для полегшення навчання та сприяння активного залучення студентів з метою знайти та проаналізувати засоби для спрощення та популяризації процесів навчання.

Для досягнення цієї мети поставлено такі завдання:

1. Дослідити наявні додатки для навчання, які використовують технології або методики для полегшення та популяризації навчання.

2. Оцінити популярність та взаємодію з користувачами наявних аналогів

3. Розробити прототип Android-додатку, використовуючи обрані методики та засоби полегшення та популяризації навчального процесу.

4. Визначити ефективність методик та засобів

5. Розробити рекомендації для вдосконалення додатку на основі отриманих результатів.

6. Визначити потенційні можливості та виклики для подальшого вдосконалення застосування методик та засобів для полегшення та популяризації навчання у Android-додатках.

Об'єктом дослідження є методи та засоби проектування додатку для полегшення навчання та сприяння активного залучення студентів

Предметом дослідження є процес розробки додатку для полегшення навчання та сприяння активного залучення студентів.

Наукова новизна. Наукова новизна даної магістерської роботи проявляється у вперше використаному підході до поєднання теорії ігор та гейміфікації в розробці Android-додатка для навчання. Робота не лише розглядає теоретичні аспекти використання різних засобів та методик у навчанні, але і впроваджує ці принципи у конкретний мобільний додаток, що сприяє покращенню якості навчання та активізації студентської участі.

Робота вперше розглядає конкретний вплив Android-додатка на мотивацію студентів та їх результативність у навчанні. Це дозволяє отримати нові знання щодо ефективності застосування технологій в освіті, без обов'язкового використання гейміфікації.

Практичне значення полягає у внесенні новаторського підходу до використання мобільних технологій у вищій освіті та навчальних процесах.

Розробка Android-додатка спрямована на використання сучасних технологій для удосконалення процесу навчання. Застосування мобільних

платформ в навчанні стає важливим етапом у вдосконаленні освітніх стандартів та забезпеченні взаємодії студентів із змістом навчання.

Мобільний додаток розроблено з урахуванням сучасного способу життя студентів, які активно користуються мобільними пристроями. Це дозволяє підтримувати високий рівень доступності та залучення студентів у навчальний процес.

Значущість полягає в оптимізації навчання через розробку ефективного інструменту, який дозволяє студентам активно взаємодіяти з навчальними матеріалами та сприяє покращенню їх розуміння та освоєнню знань.

Публікації. Матеріали дослідження з тематики кваліфікаційної роботи доповідалися на XVI Всеукраїнській науково-практичній WEB конференції аспірантів, студентів та молодих вчених 21-23 березня 2023 року.[12, 13]

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Визначення напрямків предметної області

В розділі 1.1. "Визначення напрямків предметної області" цієї магістерської роботи глибше розглядаються ключові аспекти, які визначають сутність та обґрунтування розробки Android додатку, спрямованого на полегшення процесу навчання та активізацію участі студентів. Цей розділ відіграє важливу роль у встановленні фундаментальних принципів та напрямків, які визначають цілі та завдання дослідження. У ході розгляду було винесено такі напрямки предметної області:

Сучасний стан освіти та перспективи його поліпшення через мобільні технології.

Освітній ландшафт переживає неабиякі трансформації. Аналізуючи його сучасний стан, ми розглядаємо складність доступу до навчальних ресурсів, виклики, пов'язані зі зниженням мотивації студентів та їхньою пасивністю у навчальному процесі. Одним із ключових моментів є віддаленість навчання, що породжує проблеми із доступом до матеріалів та взаємодії учасників освітнього процесу. Мобільні технології можуть стати катализатором покращення, забезпечуючи доступ до навчальних ресурсів будь-де та будь-коли. При цьому вони можуть активізувати інтерактивність та ефективність у навчальному процесі.

Перш за все, сучасні студенти є активними користувачами мобільних пристроїв, таких як смартфони та планшети. Вони вже звикли до використання додатків для розваг, комунікації та різних корисних функцій. Створення Android-дodatка для навчання стане зручним інструментом для студентів, оскільки вони можуть легко і швидко отримувати доступ до необхідних матеріалів та ресурсів зі свого мобільного пристрою.

Застосування мобільних технологій у навчальному процесі відкриває двері до нових методів викладання та навчання. Вони можуть сприяти створенню інтерактивних навчальних інструментів, від яких студенти зможуть отримати не

лише інформацію, а й можливість активного взаємодії з навчальним матеріалом. Це стимулює зацікавленість та підвищує рівень засвоєння знань.

Останні події у світі та Україні підкреслюють необхідність адаптації освітнього процесу до віддалених форм. Мобільні технології стали важливим засобом забезпечення неперервності навчання під час кризових ситуацій. Вони дозволяють створити віртуальні класи, сприяючи дистанційному спілкуванню та вивченню. Однак викликом залишається забезпечення рівного доступу до технологій для всіх учнів та забезпечення якісної освіти у віддаленому форматі.

Важливість та переваги Android додатків у навчанні

Android платформа відкриває широкі можливості для розробки освітніх додатків. Вона є масово поширеною та дозволяє розробникам створювати інноваційні, персоналізовані продукти. Android має гнучкість у впровадженні функцій та інтерфейсів, що робить його привабливим для створення додатків, спрямованих на навчання. Врахування цієї платформи виправдовується її великою аудиторією та технічними можливостями, які можна ефективно використати для розвитку освітніх додатків.[8]

Майже кожен в Україні вже має смартфон і переважна кількість користувачів обрали саме платформу Android для щоденного користування. Телефони на з цією операційною системою є найдоступніші на ринку, що виправдовує їх популярність. З розвитком технологій

Однією з головних переваг Android є його гнучкість. Ця платформа дозволяє розробникам створювати різноманітні додатки з різноманітними функціями та інтерфейсами. Від ігор до освітніх додатків, від медіа до продуктивних інструментів - Android може втілювати широкий спектр можливостей для навчання.

Спільнота розробників Android постійно зростає та активно співпрацює, обмінюється досвідом та ідеями. Це створює сприятливе середовище для розвитку інновацій та вдосконалення освітніх додатків. Можливість обговорення, спільної роботи та використання відкритих ресурсів сприяє постійному покращенню функціоналу та якості програм.

Платформа Android пропонує багатий набір функцій, що дозволяє створювати додатки з різноманітними функціональними можливостями. Це включає в себе можливості використання різних мультимедійних форматів, інтерактивних елементів, а також інтеграцію з різними сервісами та пристроями. Завдяки цьому розробники можуть створювати персоналізовані додатки, що відповідають унікальним потребам користувачів.[5]

Орієнтація на підвищення ефективності та залучення студентів.

Щоб стимулювати активну участь, додаток має бути інтерактивним та цікавим. Персоналізовані завдання, гейміфікація та можливості співпраці між студентами та викладачами стануть основою для створення залучаючого середовища. Застосування інноваційних методик, таких як штучний інтелект для індивідуального підходу до студентів, та розширена реальність для покращення здатностей сприйняття матеріалу, стануть ключовими елементами для створення ефективного навчального додатку.

Додаток також може полегшити організацію навчального процесу. Він може включати календар подій, розклад занять, нагадування про важливі події та завдання. Такий функціонал дозволить студентам краще планувати свій час та уникати забутливості щодо найважливіших дедлайнів та подій.

Створення інтерактивного навчального досвіду є важливим аспектом, який сприяє активній участі студентів у процесі навчання. Мобільні додатки можуть пропонувати інтерактивні завдання, тести, вправи або можливість обговорення та обміну ідеями. Це сприяє залученню студентів та створює комунікаційну платформу між ними та викладачами.

Одним із ключових аспектів ефективного навчання є здатність навчального середовища адаптуватися до потреб кожного студента. Додатки можуть надати можливість налаштування контенту під індивідуальні особистість та освітні потреби кожного учня. Це дозволяє створювати персоналізовані шляхи навчання, що підвищує ефективність засвоєння матеріалу.

Використання гейміфікаційних методик може стимулювати активність та підвищувати мотивацію студентів до навчання. Відзначення досягнень, створення рейтингів, використання елементів змагання та нагородження за успіхи можуть стати інструментами, що заохочують учнів активно вивчати матеріал та брати участь у навчальному процесі.

Включення інноваційних технологій, таких як штучний інтелект та розширена реальність, може додати нові розміри до ефективності додатку. Використання індивідуалізованих рекомендацій, вивчення через взаємодію з віртуальними об'єктами або доповненням реальності може значно збільшити поглиблення вивчення та зацікавленість студентів.

Встановлення основних завдань дослідження.

Основною метою роботи є створення додатку, який не лише спростить процес навчання через мобільність, а й стимулюватиме активність та інтерес студентів до знань. Дослідження та реалізація цих пунктів визначають напрямки для подальшої розробки додатку.

Одним із основних завдань є розробка унікального контенту та функціоналу, які будуть відповідати потребам сучасних студентів. Це включає в себе розробку цікавого, високоякісного контенту та інструментів, які сприятимуть активному навчанню та стимулюванню інтересу до отримання знань.

Необхідно провести тестування прототипу додатку серед різних груп користувачів для оцінки ефективності та прийняттям учасниками навчального процесу. Збір зворотного зв'язку та аналіз його допоможуть виявити сильні та слабкі сторони додатку, що дозволить підвищити його ефективність та корисність для користувачів.

Одним з ключових завдань є створення моделі взаємодії користувачів з додатком. Це включає в себе аналіз користувацьких потреб, зручності та ефективності використання інтерфейсу, щоб забезпечити максимально зручний та доступний інструмент для навчання.

Після запуску додатку, важливо провести оцінку його ефективності на основі зібраної статистики та відгуків користувачів. Це дозволить виявити можливості для подальшого вдосконалення, враховуючи реальні потреби користувачів та їхній досвід використання додатку.

З точки зору ефективності, Android-додаток для полегшення навчання має значний потенціал зменшити бар'єри до доступу до освіти. Завдяки мобільності та доступності, студенти можуть навчатися в будь-який час та в будь-якому місці, що робить освіту більш гнучкою та доступною для всіх шарів суспільства.

Також, враховуючи теперішні реалії, такі як тотальна епідемія вірусу COVID-19(NCOV-19), що застала нас несподівано в кінці 19 року та призвела до тотального карантину, що в свою чергу стало однією з причин активного розвитку технологій дистанційного навчання. Несподівана збройна агресія стала новим випробуванням для освітньої системи, яка почала оправлятися після карантину і знову технології дистанційного навчання стали потрібними, актуальними та життєво важливими для освіти та науки в Україні. Я вважаю, що розробка мобільного додатку, який дозволить отримувати якісну освіту та полегшить навчальні процеси стане великим кроком у розвитку освітньої системи України та незабаром усього світу.

1.2. Аналіз існуючих рішень

Із методик, які стануть у нагоди при розробці додатку для спрощення освітнього процесу можна перелічити такі:

- Адаптивне навчання
- Гейміфікація
- Активне навчання
- Методи мультимедіа
- Теорія конструктивізму та когнітивного навчання

Адаптивне навчання

Адаптивне навчання - це методика освіти, яка фокусується на індивідуалізації та персоналізації процесу навчання для кожного учня. Основна

ідея полягає в тому, щоб створити навчальне середовище, яке враховує потреби, здібності, темп та стиль навчання кожного конкретного учня.

Цей підхід враховує індивідуальні особливості кожного учня, дозволяючи їм навчатися власним темпом та зосереджуватися на конкретних аспектах матеріалу. Основні принципи адаптивного навчання базуються на використанні технологій, що дають змогу автоматично адаптувати матеріали та завдання до рівня знань та індивідуальних потреб учнів.

Центральним елементом адаптивного навчання є персоналізація. Замість традиційної однакової програми для всіх учнів, адаптивне навчання використовує дані про успішність, прогрес та особистість кожного учня, щоб створити індивідуальні навчальні плани. Це може включати різноманітні завдання, матеріали, рівень складності та темп навчання, що оптимізовані під кожного користувача.[20]

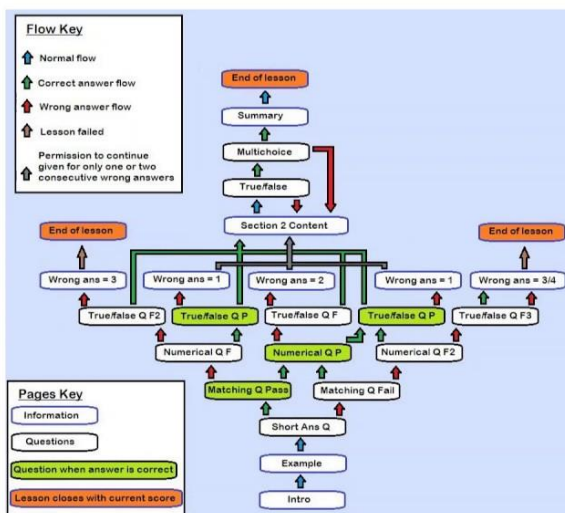


Рис. 1.1. – Схема роботи адаптивного навчання

Однією з переваг адаптивного навчання є здатність підтримувати мотивацію учнів та забезпечувати ефективніший процес навчання. Завдяки індивідуальному підходу, учні відчувають більшу відповідальність за своє навчання та можуть досягати кращих результатів, отримуючи особисто зрозумілі завдання та матеріали.

Перевагами таких систем навчання є надання споживачам їх ширших можливостей щодо вільного вибору власної освітньої траєкторії й рівня оволодіння компетентністю, регулювання темпу навчання і строків засвоєння навчального матеріалу, планування самостійної та індивідуальної роботи над тією чи іншою темою. Крім того, в них передбачено диференціацію навчання залежно від рівня знань і вмінь учня та його пізнавального досвіду, індивідуальних особливостей сприйняття навчального матеріалу, здібностей до розв'язування завдань різної складності тощо. За таких умов відбувається перехід від найпростішого алгоритму вивчення одного й того ж навчального матеріалу всіма учнями до складного алгоритму генерування змісту й методів навчання, коли різні учні досягають запланованих результатів різними способами організації освітнього процесу, у різному темпі й на основі різних варіацій подавання навчального матеріалу.[1]

Таким чином, адаптивне навчання – це така дидактична система набуття компетентності, яка враховує індивідуальні особливості в організації навчальної діяльності учнів, що надає їм можливість вибудувати власну освітню траєкторію, зважаючи на індивідуальний темп навчання, глибину освоєння змісту й освітні потреби. Ефективність такого навчання значно підвищується за умов використання інформаційно-комунікативних технологій, що надає можливості для розширення індивідуалізації й диференціації освітнього процесу відповідно до когнітивних особливостей учнів.[1]

Гейміфікація

Гейміфікація - це стратегія, яка використовує принципи та елементи ігор у нетрадиційних контекстах, зокрема в освіті, бізнесі, здоров'ї та інших сферах, для стимулювання мотивації, залучення учасників та підвищення їхньої участі та продуктивності.

Основна мета гейміфікації полягає в тому, щоб перетворити будь-яку діяльність на щось подібне до гри, з використанням ігрових елементів, таких як точки, рівні, бейджі, лідерборди, винагороди та завдання, щоб стимулювати учасників до досягнення цілей.

У контексті освіти, гейміфікація може бути використана для залучення студентів та покращення їхнього навчального процесу. Вона може включати ігрові завдання для вивчення нового матеріалу, виконання тестів у формі гри, створення конкурсів та співробітництва між учнями, що стимулює активність та покращує засвоєння інформації.

Головна перевага гейміфікації полягає в тому, що вона може стимулювати інтерес до навчання або виконання завдань, надихаючи учасників до участі через ігрові елементи, які зазвичай сприймаються як веселі та захоплюючі. Це може допомогти уникнути монотонності та збільшити мотивацію до досягнення цілей.

Принципи гейміфікації можуть бути застосовані в різних контекстах, від навчання до розваг та робочого оточення. Вона використовується для створення збалансованого середовища, що сприяє зростанню вмінь, мотивації та досягнень учасників.

Незважаючи на свої переваги, гейміфікація вимагає обережного планування та розробки, оскільки не всі ігрові елементи можуть ефективно заохочувати чи зацікавлювати всіх учасників. Однак, правильно впроваджена гейміфікація може значно поліпшити участь та результати у відповідних сферах.

Як не дивно, але в класичній освіті забувають про дуже просту, але безмежно значущу річ — про те, що те, що ми робимо має приносити нам радість. Це робить ігри такими захоплюючими. І цього так не вистачає в процесі навчання. Звичайна академічна програма влаштована досить нудно: лекції — семестри — іспити — лекції — семестр — іспит. Але чому б не організувати це як у грі, з подоланням рівнів складності і поступовим перетворенням новачка в магістра? Монотонна дорога стає рельєфною, з'являється мотив переходити від одного рівня на іншій. Ви просуваєтесь до своєї мети шляхом гравця. Виникнення галузі навчальних ігор для дітей починається у 1980-х роках. Гейміфікація або ігрофікація проникає в усі сфери життя. Починаючи від професійної діяльності, закінчуючи системою освіти, наприклад, навчальна гра на основі покеру «Minecraft» навчає концепціям інженерно-будівельного мистецтва в школах Швеції, «Lumosity» — наче тренажерний зал для мозку. Все

частіше ми зустрічаємося з інфографікою (графічне візуальне подання інформації, даних або знань, призначених для швидкого та чіткого відображення комплексної інформації).[3]

Вважаємо, що гейміфікація не повинна відводити слухача вбік від запланованого результату – вона повинна допомагати досягти його. Часто використовуючи гейміфікацію розробники захоплюються самою механікою гри, забуваючи про поставлені цілі навчального 5 курсу. Для запобігання такій ситуації важливо завжди пам'ятати, чого ми хочемо від користувача, який проходить наш курс. Додаючи ігрові елементи необхідно відразу перевіряти, чи допомагають вони нашій аудиторії поліпшити розуміння та подальше застосування отриманої інформації. Нам потрібно усвідомити, що головна мета застосування гейміфікації – зміна поведінки людини або аудиторії в цілому, тобто гейміфікація потрібна не для розваг, а для підвищення ефективності навчального процесу.[14]

Активне навчання

Активне навчання - це педагогічний підхід, що акцентує на активній участі студентів у навчальному процесі через взаємодію, співпрацю та самостійну діяльність замість пасивного приймання інформації від вчителя. Цей підхід спрямований на створення навчального середовища, де учні активно залучені до процесу навчання, сприяючи їхньому зростанню як особистостей та учнів.

Основні принципи активного навчання включають співпрацю, обмін думками, розв'язання проблем, аналіз та синтез інформації. Замість традиційної лекційної форми, активне навчання використовує різноманітні методи, такі як групова робота, дискусії, проекти, вправи для розвитку критичного мислення та співпраці.

Цей підхід стимулює учнів до активної участі в навчанні та сприяє розвитку різних навичок, таких як критичне мислення, комунікація, співпраця та розв'язання проблем. Він також сприяє зростанню мотивації, оскільки учні відчувають більшу відповідальність за власне навчання та більшу зацікавленість у процесі.

Серед інструментів активного навчання можна виділити "метод кишень", коли студенти розв'язують певні завдання в малих групах, обговорюючи результати з усією аудиторією. Також до цього підходу відносяться методи поглибленого аналізу, проекти з реальними ситуаціями, рольові ігри та співпраця в парах для вирішення завдань.

Активне навчання допомагає створити стимулююче, інтерактивне середовище, де кожен студент може відчувати себе активним учасником навчального процесу, що сприяє кращому засвоєнню матеріалу та розвитку широкого спектру навичок, необхідних для успішного життя.

Активне навчання (active learning) - це процес, в центрі якого знаходиться навчання. Воно зосереджується на тому, як учні вчаться, а не лише на тому, що вони вивчають. Під час такого навчання учнів заохочують "напружувати зивини" ('think hard'), а не тільки пасивно отримувати інформацію від викладача. Активне навчання базується на теорії конструктивізму. Конструктивізм підкреслює той факт, що учні конструюють або будують своє власне розуміння. Конструктивісти стверджують, що навчання - це процес "створення сенсу". Учні розвивають свої наявні знання, щоб досягти глибшого рівня розуміння. Це означає, що учні стають більш здатними аналізувати, оцінювати та синтезувати ідеї (таким чином досягаючи навичок вищого порядку за таксономією Блума). Активне навчання (active learning) - це процес, в центрі якого знаходиться навчання. Воно зосереджується на тому, як учні вчаться, а не лише на тому, що вони вивчають. Під час такого навчання учнів заохочують "напружувати зивини" ('think hard'), а не тільки пасивно отримувати інформацію від викладача. Активне навчання базується на теорії конструктивізму. Конструктивізм підкреслює той факт, що учні конструюють або будують своє власне розуміння. Конструктивісти стверджують, що навчання - це процес "створення сенсу". Учні розвивають свої наявні знання, щоб досягти глибшого рівня розуміння. Це означає, що учні стають більш здатними аналізувати, оцінювати та синтезувати ідеї (таким чином досягаючи навичок вищого порядку за таксономією Блума).[17]

Засоби мультимедіа

Методи мультимедіа - це підхід до навчання, який використовує різні форми медіа, такі як відео, аудіо, графіка та інтерактивні елементи, для передачі інформації та сприяння засвоєнню знань учнями. Цей підхід створює більш різноманітне, привабливе та ефективне навчальне середовище.

Одним з основних елементів методів мультимедіа є використання візуальних матеріалів, таких як відеоролики, ілюстрації, схеми та графіка. Вони допомагають візуалізувати концепції, що полегшує засвоєння матеріалу, особливо для візуально налаштованих учнів.

Додатково, аудіоелементи включають у себе записи лекцій, аудіокниги та інші аудіоматеріали, що дозволяють учням слухати та розуміти матеріал у формі звукових файлів.

Також методи мультимедіа можуть використовувати інтерактивні елементи, такі як веб-сайти, вправи, ігри та програми, які дозволяють учням активно взаємодіяти з навчальним матеріалом. Це створює більш залучаючий та динамічний процес навчання.

Важливим аспектом методів мультимедіа є їхня гнучкість та доступність. Завдяки цим методам, учні можуть мати доступ до навчального матеріалу в будь-який час та з будь-якого пристрою, що сприяє зручності та глибшому засвоєнню інформації.

Комбінація різноманітних медіа формує багатоаспектний навчальний досвід, сприяючи кращому розумінню та запам'ятовуванню матеріалу. Використання методів мультимедіа в освіті дозволяє забезпечити більш ефективний та цікавий спосіб навчання, відповідаючи потребам сучасного цифрового світу.

Попри безапеляційний факт широкого застосування мультимедійних навчальних систем в освітньому процесі на всіх етапах і в різних формах, відчувається дефіцит доступних освітніх мультимедійних засобів і програм. Застосування мультимедійних технологій може позитивно позначитися на відкритому і дистанційному навчальних процесах, зокрема, стимулювати

когнітивні аспекти навчання (сприйняття та усвідомлення інформації); посилити мотивацію студентів; допомогти у розвитку навичок спільної роботи й колективного пізнання; розвинути фундаментальний підхід до навчання.[16]

Однією з основних сфер застосування ММС є освіта, у широкому розумінні слова, включаючи як навчальні програми і курси, так і відеоенциклопедії, довідники, словники, інтерактивні путівники, тренажери, ситуаційно-рольові ігри тощо. Розв'язання проблеми поєднання потоків інформації різної модальності (звук, текст, графіка, відео) робить комп'ютер універсальним інструментом у практично будь-якій галузі знання і людської діяльності. Тенденція до швидкого зниження вартості апаратного і програмного забезпечення ММС, поліпшення їхніх технічних і споживчих характеристик, розвиток оригінальних авторських інструментальних систем стали причиною до того, що навчальні мультимедіа-продукти створюються великою кількістю фірм, навчальних і наукових установ, приватних осіб. Це значно наповнює ринок навчальних продуктів із різних галузей і професій.

Отже, можна сподіватися, що застосування технології мультимедіа здатне значно підвищити ефективність навчання. Простий і дружній інтерфейс робить мультимедіа-курс незамінним під час самостійної підготовки учня. Тому такі курси знайдуть все більш широке застосування в системі дистанційної освіти. Зрозуміло, значною мірою успіх залежить і від правильно організованого навчального процесу. Зокрема, якщо метою є освоєння практичних прийомів і навичок, то впровадження навчання з використанням комп'ютерів і інтерактивних програм мультимедіа чекає безсумнівний успіх. [4]

Теорія конструктивізму та когнітивного навчання

Теорія конструктивізму та когнітивного навчання представляють собою ключові підходи до психології навчання, що визнають активну участь учня у процесі засвоєння знань та покладаються на його конструктивну діяльність та мислення. Ці теорії підкреслюють важливість процесу побудови знань, розуміння та внутрішнього перетворення інформації в мозкову активність.

Теорія конструктивізму, вперше сформульована в середині 20 століття, стверджує, що людина активно конструює свої знання шляхом взаємодії з оточуючим середовищем. Згідно з цією теорією, навчання відбувається через власний досвід, а не просто прийняття інформації. Учні будують нові знання на основі своїх попередніх знань, переконань та досвіду. Важливою є взаємодія з іншими учнями та оточуючим середовищем для розвитку своїх розумових здібностей.[15]

Конструктивізм підтримує ідею, що навчання повинно сприяти активному процесу конструювання знань, а не лише передачі інформації вчителем учневі. Цей підхід ставить учня у центр навчального процесу, дозволяючи йому вирішувати проблеми, ставити запитання та досліджувати новий матеріал.

Когнітивне навчання, з іншого боку, фокусується на процесах мислення та сприйняття, які відбуваються у мозку під час навчання. Ця теорія акцентує на уявленні про те, як люди опрацьовують інформацію, вирішують проблеми та засвоюють нові знання. Вона вивчає психологічні процеси, такі як сприйняття, увага, пам'ять, мислення та мовлення, які допомагають у розумінні та засвоєнні інформації.

Когнітивна теорія навчання відмічає, що людина активно переробляє інформацію, адаптується до нових знань та створює власні ментальні моделі світу. Основна ідея полягає в тому, що навчання відбувається через розумову діяльність, тобто через сприйняття, уявлення, розуміння та застосування інформації у різних ситуаціях. [19]

Обидві теорії підкреслюють важливість активності та власного конструктивного мислення учня в процесі навчання. Вони вказують на те, що навчання повинно бути спрямоване на стимулювання учнів до пошуку, розв'язання проблем та активного використання знань, щоб сприяти їхньому особистісному розвитку та засвоєнню нового матеріалу.

З погляду на сьогоденні задачі навчання і особливо його конкретне спрямування на особистість, конструктивістський підхід є найбільш придатним підходом до системного формування знань і проектування діяльності згідно з

кінцевим результатом. Зберігання фрагментів сприйняття пам'яттю у вигляді певних «конструкцій», формування знаннєвих систем на когнітивному ґрунті і неформальне відношення до самого процесу опрацювання знань як до особистої потреби «виробляти» і «створювати», використовуючи форми співробітництва («взаємопідтримка» за П. К. Анохіним), – це і є сучасний підхід до навчальної діяльності.[9]

1.3. Основні моделі, методи і підходи гейміфікації

Гейміфікація (від англ. Gamification – геймізація, геймерізація, ігрофікація) – один із сучасних напрямків засвоєння інформації та навчання. Ігрофікація – за визначенням Оксфордського словника, – застосування типових елементів ігрового процесу (підрахунок балів, змагання між гравцями, наявність правил гри) в інших галузях діяльності з метою заохочення до взаємодії. Більш ніж півмільярда людей у світі грає в онлайн ігри мінімум годину на день. 97 % хлопців та 94 % дівчат до 18 років грають регулярно.

Існують найрізноманітніші моделі гейміфікації, які використовуються для проектування та реалізації гейміфікованих досвідів. Основними з них вважають:

Бейджі – це нагороди, які надаються користувачеві за виконання певних завдань або досягнення результатів. Ці нагороди можуть відображатись у профілі користувача, показуючи досягнення їх власника всім, хто заходить на сторінку.

Рейтинги – це системи, які оцінюють успішність користувачів в різних аспектах, таких як рівень активності, рівень знань або статуси проходження курсів чи лекцій. Такі рейтинги надихають користувачів до процесу змагання, що в свою чергу на пряму впливає на ефективність навчання у студентів.

Прогрес-бари – це інтерактивні елементи, які відображають прогрес користувача у певному завданні, курсі або будь-якій іншій сфері, прогрес якої можна вимірювати. Вони мотивують користувачів до досягнення цілей та створюють почуття задоволеності від успіху.

Методи гейміфікації на пряму направлені на те, щоб залучити природні людські інстинкти такі як: конкуренція, досягнення, статус, самовираження,

альтруїзм та розв'язання задач. В основі стратегії гейміфікації лежить винагородження за виконані завдання.

Методи гейміфікації показують шляхи застосування ігрових механік та елементів гри для збільшення ефективності, зручності та актуальності певного процесу, наприклад навчання. До таких шляхів можуть належати бейджі, рівні, приналежність до кланів, рейтинги, винагороди за пророблену роботу тощо. Методи гейміфікації використовуються для підвищення мотивації, зниження відчуття монотонності та нудності.

Основними методами гейміфікації в освіті вважають:

Гра: Використання ігрових елементів для викладання нових знань та навичок. Наприклад, використання ігрових симуляцій для вивчення науки, історії або математики.

Бали та досягнення: Система нагородження учнів за виконання певних дій або досягнення цілей. Наприклад, створення системи балів за виконання домашніх завдань, відповіді на питання або виконання додаткових завдань.

Ранги та рейтинги: система порівняння учнів за результатами їхньої діяльності. Наприклад, створення рейтингу класу за кількістю балів, отриманих за домашні завдання, відповіді на питання та інші дії.

Місії та завдання: система виконання певних дій для отримання бонусів або досягнення цілей. Наприклад, створення завдань на кожен урок, виконання яких допоможе учням закріпити отримані знання та навички.

Персонажі та історії: Система створення персонажів або історій для залучення учнів до навчального процесу. Наприклад, створення історії про винахідника, яка допоможе учням краще зрозуміти історію розвитку науки.

Звертаючись до дослідження, проведеного компанією IBM, близько 90% співробітників, що прийняли участь у пограмі навчання, у якій були застосовані елементи гейміфікації заявили, що такий підхід значно допоміг їм краще освоїти матеріал та підвищити рівень роботи.[3]

Також, звертаючись до звіту, наданого компанією TalentLMS, використання гейміфікації у навчальних процесах збільшує залучення студентів

приблизно на 40%, полегшує освоєння матеріалу на 55% та покращує шанси успішно завершити курси на 50%.[3]

Метод гейміфікації «Гра»

Метод гейміфікації "Гра" включає використання ігрових елементів у процесі навчання, що дозволяє зробити процес більш захопливим та привабливим для учнів.

У цьому методі навчання можуть брати участь різні ігрові елементи, такі як графіка, звук, механіка гри, анімація, відео та інші. Наприклад, можна використовувати ігрові симулятори для вивчення наукових понять, створювати ігрові завдання для закріплення вивченого матеріалу або використовувати ігрову механіку для вирішення навчальних завдань.



Рис 1.1 – Складові концепта метода гри

При використанні методу гейміфікації "Гра" учні стають активними учасниками навчального процесу, бо вони займаються не просто слуханням лекцій або читанням матеріалу, а активно взаємодіють з ігровими елементами, вирішують ігрові завдання та отримують миттєву зворотну інформацію про свої успіхи.

Чотири концепти, в основі яких лежить ідея гри: гейміфікація, серйозні ігри, забавки та ігровий дизайн. Відмінності між ними лежать у двох вимірах:

- ігри/забавки (gaming/playing) – свідчить про спрямування та регульованість діяльності;
- повністю/частково (whole/parts) – вказує на міру інтеграції ігрових елементів у процес.

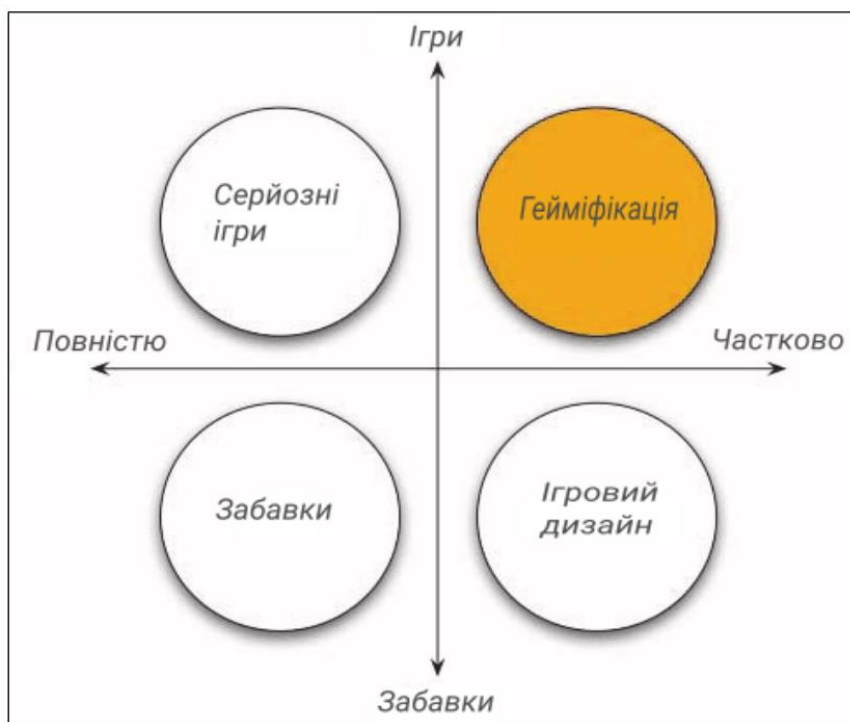


Рис. 1.2 – Концепти, в основі яких лежить ідея гри

Серйозні ігри (serious games) мають конкретну мету, націлену на розв’язання реальних життєвих ситуацій.

Забавки (toys) – це ігри, що не мають чітких правил, а також не націлені на конкретний результат чи мету; вони орієнтовані лише на переживання позитивних емоцій або невимушене дослідження.

Ігровий дизайн (playful design) теж не має конкретної мети, як підтримується правилами; його використовують, щоби зробити процес більш людським, приємним та легким для сприйняття.

Гейміфікація використовує елементи гри, але основа процесу залишається незмінною. Втім, межі між цими концептами є радше теоретичними, натомість, на практиці вони можуть успішно поєднуватись.

Метод гейміфікації «Бали та досягнення»

Метод гейміфікації "Бали та досягнення" полягає у використанні системи нагород та досягнень для стимулювання та мотивації користувачів до активного навчання та досягнення поставлених цілей. Зазвичай користувач отримує бали за успішне виконання завдань та проходження різних рівнів складності. За накопичені бали користувач може отримати досягнення та піднятися на наступний рівень.

Отримання досягнень може стати додатковим стимулом для користувачів. Наприклад, користувач може отримати досягнення за успішне виконання завдань певного рівня складності, за проходження курсу, за виконання додаткових завдань тощо. Кожне досягнення може мати свій власний опис та зображення, що зробить навчання більш захопливим та інтерактивним. На рисунку 1.3 наведено приклади таких досягнень.

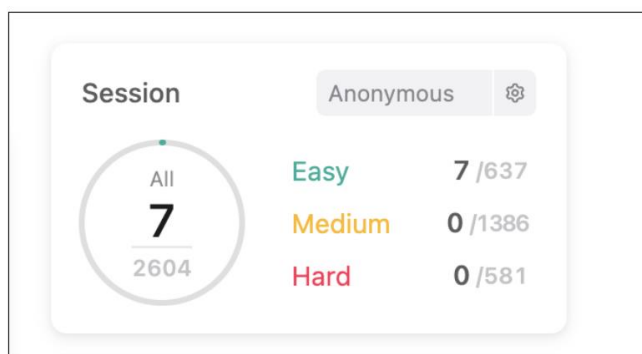


Рис. 1.3 – Приклад відображення досягнень

Крім того, додаток може містити список досягнень, які студент може отримати при досягненні певних мет. Наприклад, "Майстер програмування" - за успішне виконання великої кількості завдань різної складності, або "Експерт" - за успішне виконання всіх завдань на певному рівні складності.

Отже, метод гейміфікації «Бали та досягнення» є ефективним способом у процесі створення додатку для полегшення навчального процесу. Використання системи нагород стимулює користувачів до більш активного та цілеспрямованого навчання.

Метод гейміфікації «Ранги та рейтинги»

Метод гейміфікації "Ранги та рейтинги" полягає у використанні системи позиційного ранжування користувачів за їхнім досягненням та успіхами в процесі навчання.

Цей метод може бути реалізований шляхом введення рейтингу користувачів, який буде відображати їхній рівень знань та успішність у вирішенні завдань. Користувачі, які більше успішно пройшли курс, отримують вищі позиції у рейтингу.

Рейтинг можна відображати як загальний, так і в рамках конкретної категорії (наприклад, мови програмування, рівня складності завдань тощо). Також можна ввести систему рангів, які будуть надавати користувачам певні привілеї та можливості.

Крім того, введення рейтингу може сприяти взаємодії та конкуренції між користувачами. Вони зможуть порівнювати свій рівень знань та успішність з іншими користувачами, стимулювати один одного до активного навчання та досягнення більш високих результатів.

Отже, цей метод є ефективним та корисним для підвищення мотивації та зацікавленості студентів у процесах навчання. Використання системи рангів та рейтингів створює конкурентне середовище, що стимулює користувачів до отримання кращих результатів та підвищення рівня знань.

Метод гейміфікації «Місії та завдання»

Метод гейміфікації "Місії та завдання" використовується для створення відчуття гри та пригоди в процесі навчання. Основна ідея полягає в тому, щоб перетворити навчання на пригоду з набором завдань, які необхідно виконати, щоб здобути нові знання та навички.

Цей метод може бути реалізований через створення різноманітних місій та завдань для користувачів. Наприклад, користувач може отримувати завдання на написання певного блоку коду, вирішення складної задачі, створення власного проекту тощо. За кожне успішно виконане завдання користувач отримує бали, які додаються до його рахунку.

Для того, щоб користувач відчував себе у грі, можна використовувати сюжетну лінію, яка відображатиме його прогрес у навчанні. Наприклад, кожне нове завдання може представляти наступний етап пригоди, в якій герой (користувач) отримує нові знання та навички для подолання наступних викликів.

Крім того, в додатку можна використовувати бонусні завдання, які дають додаткові бали та навички, а також можуть мати певні призи за їх успішне виконання.

Таким чином, метод гейміфікації "Місії та завдання" може стати відмінним інструментом для підвищення мотивації та зацікавленості користувачів у навчанні. Створення пригодницької сюжетної лінії та різноманітних завдань може допомогти користувачам забути про монотонність навчання та зануритися у світ гри, де їхні успіхи будуть визнані та винагороджені. Крім того, ранги та рейтинги можуть бути також використані для стимулювання користувачів до досягнення нових цілей та досягнень у навчанні.

Отже, метод гейміфікації "Місії та завдання" може бути дуже ефективним інструментом для навчання. Він допоможе зробити навчання більш захоплюючим та мотивуючим для користувачів, що дозволить їм досягти кращих результатів у навчанні.

Метод гейміфікації «Соціальна інтеракція»

Цей метод гейміфікації включає в себе використання елементів ігор та змагань, що спрямовані на взаємодію між користувачами. Це може включати в себе використання рейтингів, змагань та співпраці між користувачами.

У контексті навчання, метод гейміфікації "Соціальна інтеракція" може бути використаний, наприклад, для створення спільноти програмістів, де користувачі можуть обговорювати технічні проблеми та рішення, а також допомагати один одному у вирішенні завдань. Така спільнота може включати в себе рейтинги та систему нагород, що дозволить користувачам отримувати бали та досягнення за активну участь у спільноті.

Крім того, метод гейміфікації "Соціальна інтеракція" може бути використаний для проведення змагань між користувачами з різних країн або

навіть між різними університетами. Наприклад, можна провести змагання зі створення найкращого програмного продукту, або виконання певних завдань. Користувачі можуть конкурувати між собою за місця у рейтингу та отримувати нагороди за перемогу.

Застосування методу гейміфікації "Соціальна інтеракція" у навчанні може допомогти збільшити мотивацію та зацікавленість користувачів до вивчення цієї галузі, а також дозволить залучити до навчання більше людей та створити активну спільноту програмістів.

Наприклад, у додатку можна створити форум, де користувачі зможуть обговорювати свої успіхи та проблеми, запитувати поради від інших користувачів. Також можна розробити систему "наставників", де досвідчені програмісти можуть допомагати новачкам у вирішенні складних завдань.

Цей метод також можна поєднати з попередніми методами гейміфікації, наприклад, створити гру, де користувачі можуть спільно виконувати завдання та змагатися між собою.

В цілому, застосування методів гейміфікації у додатку для навчання може значно збільшити мотивацію користувачів до навчання, зробити процес навчання більш захоплюючим та ефективним.

1.4 Постановка задачі дослідження

До цієї магістерської роботи, з метою дослідити проблему, було поставлено такі задачі:

- дослідити наявні додатки для навчання, які використовують технології або методики для полегшення та популяризації навчання.
- оцінити популярність та взаємодію з користувачами наявних аналогів
- розробити прототип Android-додатку, використовуючи обрані методики, засоби полегшення та популяризації навчального процесу.
- визначити ефективність методик та засобів
- розробити рекомендації для вдосконалення додатку на основі отриманих результатів.
- застосувати основні моделі, методи та засоби застосування гейміфікації.

- визначити потенційні можливості та виклики для подальшого вдосконалення застосування методик та засобів для полегшення та популяризації навчання у Android-додатках.

Висновки до першого розділу.

У розділі 1 було постановлено завдання роботи та розглянуто різні методи та засоби, що сприяють полегшенню навчання та сприяння активного залучення студентів

У підрозділі 1.1 було проведено постановку завдання роботи та доведено актуальність розробки додатку у сучасних реаліях.

У підрозділі 1.2 було перераховано популярні методи полегшення навчання для студентів та способи для сприяння активного залучення студентів у навчальному процесі

У підрозділі 1.3 було досліджено та проаналізовано моделі, методи та підходи гейміфікації у навчальних додатках.

У підрозділі 1.4 було постановлено задачі дослідження, яких буде дотримано протягом усієї роботи.

Метою цього розділу було провести огляд предметної області та обґрунтувати актуальність теми роботи. У ході огляду предметної області було вивчено основні методи та способи для полегшення навчання та сприяння активного залучення студентів.

Висновки з цього розділу будуть корисні при виборі методів, які будуть застосовані при розробці додатку для полегшення навчання та сприяння активного залучення студентів, а також подальшому розширенні проекту.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ANDROID-ДОДАТКУ ДЛЯ ПОЛЕГШЕННЯ НАВЧАННЯ ТА СПРИЯННЯ АКТИВНОГО ЗАЛУЧЕННЯ СТУДЕНТІВ

Сучасна освіта постійно розвивається, шукаючи нові технології та інструменти для покращення навчального процесу. У світі, де мобільність та доступність до інформації стали ключовими, розробка мобільних додатків стає невід'ємною складовою сучасної освіти. Не дивно, що ентузіасти та великі компанії одразу взялись за розробку платформ та додатків для полегшення отримання освіти у світі.

2.1. Аналіз аналогів додатку

Duolingo

Duolingo — електронна платформа вивчення мови і краудсорсингового перекладу тексту, що має платний та безкоштовний види акаунтів. Сервіс розроблений так, що під час уроків користувачі допомагають перекладати вебсайти, статті та інші документи.



Рис. 2.1 – Логотип додатку Duolingo

«Дуолінго» пропонує численні письмові уроки й диктанти, проте розмовним навичкам приділяється менше уваги. Програма веде ігрове дерево

навичок, за яким просуваються користувачі, і словниковий розділ, де можна практикувати вже вивчені слова. Учні отримують «бали досвіду» (монети) в міру вивчення мови, наприклад, після проходження уроку. Навички вважаються вивченими, коли учасники виконують всі пов'язані з ними уроки. За один урок можна заробити 10—15 балів. Також можна повторювати вже пройдені теми та заробити 10 балів, а також можна проходити «Легендарні уроки», де можна заробити 40—45 балів. За один курс користувач може вивчити до 2000 слів.

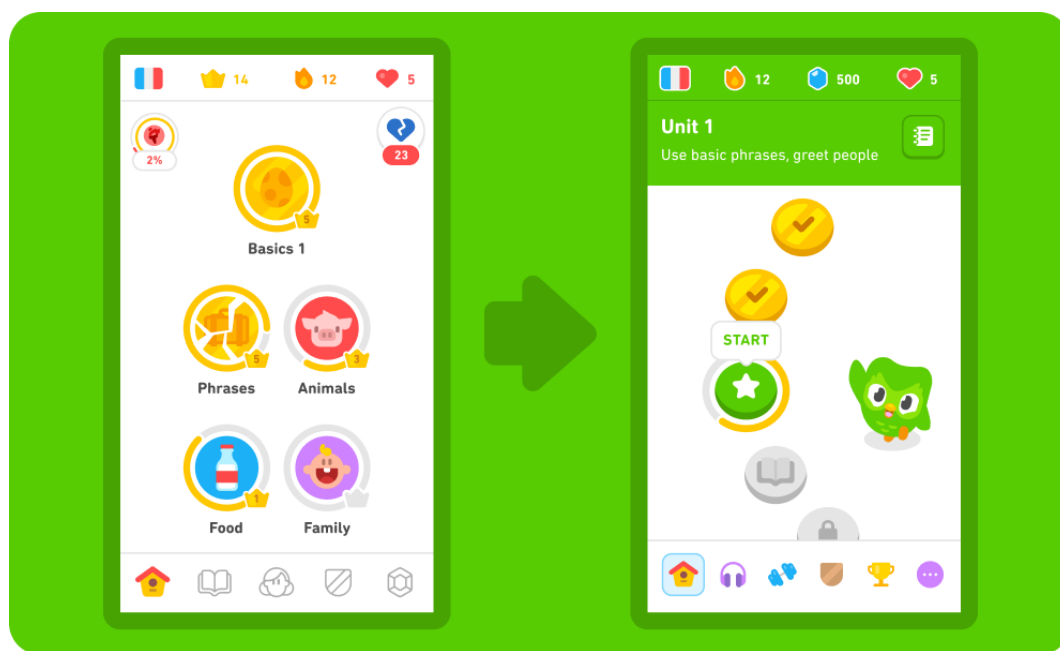


Рис. 2.2 – Меню додатку Duolingo

«Дуолінго» використовує для навчання підхід, заснований на аналізі великої кількості статистичних даних. На кожному етапі система запам'ятовує, які питання викликали в користувачів труднощі і які помилки були виявлені. Потім вона агрегує ці дані й використовує для машинного навчання. Таким чином формуються індивідуальні уроки.

Kahoot!

Kahoot! — ігрова навчальна платформа, що використовується в класі в школах та інших навчальних закладах. На сайті надано каталог ігор — «Kahoots» — кожна з яких є вікториною, що містить питання з декількома варіантами відповідей.



Рис. 2.3 Логотип платформи Kahoot!

Сайтом можна користуватися в веб-браузері, а з 2017 року також і в додатках для мобільних пристроїв (доступно на меншій кількості мов). Kahoot! може використовуватися для перевірки знань учнів, або, як перерва в класних заняттях(рис 2.4).

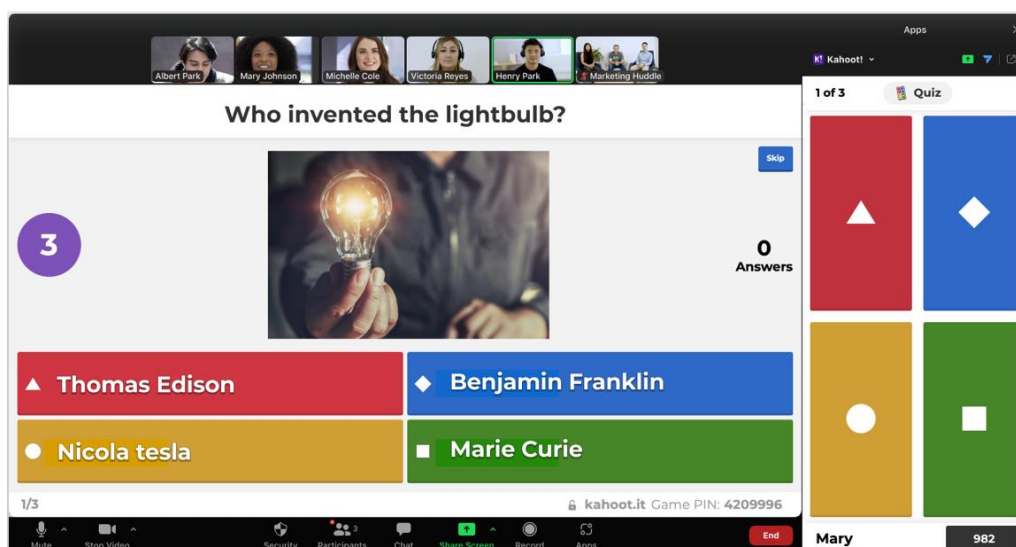


Рис. 2.4. Навчальний процес на платформі Kahoot!

Вона дозволяє викладачам створювати питання, а студенти відповідають на них за допомогою своїх пристроїв. Гравці отримують бали за правильні відповіді та змагаються за перемогу(рис 2.5).



Рис. 2.5. Приклад змагань студентів на платформі Kahoot!

ClassCraft

Classcraft — це веб-додаток, створений канадцем Шоном Янгом, який дозволяє вчителям керувати рольовою грою, у якій їхні учні втілюють різних персонажів.¹²



Рис 2.6. Логотип платформи Classcraft

Classcraft використовувався як доповнення до навчання та змінив спосіб викладання.⁴ Він був розроблений, щоб заохочувати командну роботу, підвищувати мотивацію учасників і сприяти кращій поведінці в класі.

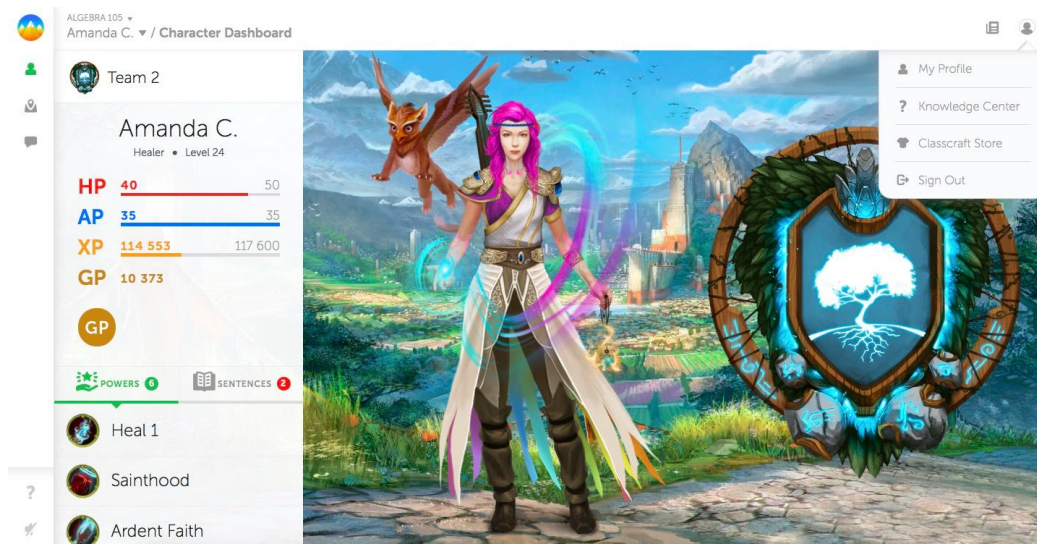


Рис. 2.7. Процес роботи на платформі Classcraft

Використовуючи різні умовності сучасних відеоігор, учні досягають вищих рівнів, працюють у команді та отримують здібності, які справді впливають на їх життя. Це приклад гейміфікації, включеної в будь-яку навчальну програму, де гра змінює те, як учні навчаються в класі протягом року.



Рис 2.8 Classcraft у навчальному процесі

Swift Playground

Swift Playgrounds - це інноваційне середовище розробки, створене компанією Apple для вивчення програмування мовою Swift. Це програмне забезпечення спеціально призначене для початківців у програмуванні, але також корисне для досвідчених розробників для вивчення нових концепцій та експериментів з кодом.

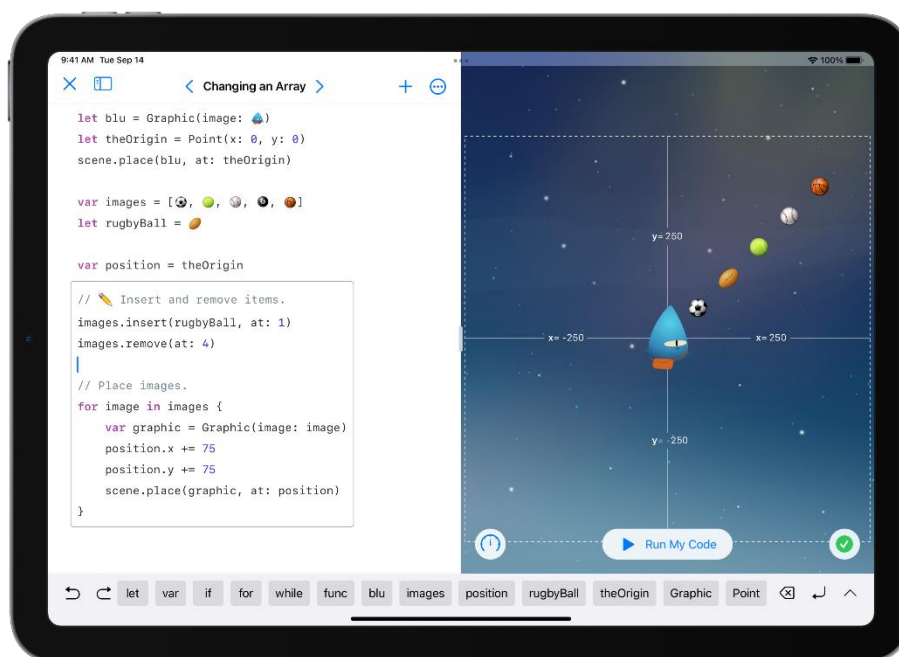


Рис. 2.9 – Навчальний процес у додатку Swift Playgrounds

Swift Playgrounds - це інноваційне середовище розробки, створене компанією Apple для вивчення програмування мовою Swift. Це програмне забезпечення спеціально призначене для початківців у програмуванні, але також корисне для досвідчених розробників для вивчення нових концепцій та експериментів з кодом.

Swift Playgrounds надає інтерактивне середовище, де користувачі можуть писати код, отримувати миттєвий зворотний зв'язок щодо правильності коду, а також бачити результати своєї роботи в реальному часі. Це середовище включає в себе різноманітні інструменти, від ігрових завдань до складніших проектів, які допомагають засвоювати основи програмування.

Одним із ключових аспектів Swift Playgrounds є можливість навчання програмуванню за допомогою інтерактивних графічних ігор та інструкцій. Він створений таким чином, щоб зробити навчання цікавим та залучаючим, зокрема для молодших користувачів або для тих, хто тільки починає вивчати програмування.

Swift Playgrounds може бути використаний для створення різних проектів, від простих до більш складних. Користувачі можуть експериментувати з кодом,

побачити його вплив на роботу програми, а також отримати відмінну можливість вивчати нові концепції програмування у навчальних цілях.

Однією з основних переваг Swift Playgrounds є те, що воно доступне безкоштовно для використання на пристроях від Apple, що дозволяє широкому колу користувачів вивчати програмування, використовуючи доступні технології.

Загалом, Swift Playgrounds відкриває шлях до зручного та захоплюючого вивчення програмування, надаючи унікальну можливість вчитися, експериментувати та розвивати навички у цьому захоплюючому та швидкозмінному світі технологій.

Було розглянуто декілька прикладів реалізацій методик та засобів заохочення студентів до навчання. Ці додатки є одними з найпопулярніших у своєму сегменті і хоча вони подобаються багатьом людям, у них є свої недоліки й переваги. Перейдемо до їх розгляду:

Duolingo:

Переваги:

- Захоплюючий дизайн у формі гри, що робить процес вивчення мови захоплюючим та цікавим.
- Різноманітна персоналізація, а саме підбір рівня складності та швидкості вивчення, що дозволяє адаптувати навчання під індивідуальні проблеми.
- Велика мобільність, завдяки якій сервіс зручно використовувати на мобільних пристроях у будь-якому місці та часі.

Недоліки:

- Обмеженість глибини знань: Можливість глибокого вивчення мови порівняно обмежена, особливо на початкових рівнях.
- Відсутність практики у реальних ситуаціях: Деякі користувачі відзначають, що навички, отримані на Duolingo, можуть бути менше

ClassCraft:

Переваги:

- Мотивація та залучення: Використання елементів гри стимулює учнів до більш активного та зацікавленого навчання.
- Співпраця та соціалізація: Classcraft сприяє розвитку командних навичок та соціальної взаємодії серед учнів.
- Персоналізовані завдання: Вчителі можуть створювати завдання, які відповідають індивідуальним потребам студентів.

Недоліки:

- Залежність від вчителя: Ефективність Classcraft може залежати від активності та вмінь вчителя використовувати платформу.
- Вартість для повноцінного використання: Деякі функції та можливості доступні тільки у преміум-версії, що може бути витратно для шкіл та вчителів.

Kahoot!:

Переваги:

- Захоплюючий та динамічний: Вікторини та ігри створюють захоплююче середовище та стимулюють учнів.
- Інтерактивність: Забезпечує активну участь студентів у процесі навчання через відповіді на питання та завдання.
- Легкість використання: Простий та доступний інтерфейс сприяє швидкому запуску та використанню.

Недоліки:

- Короткочасність ефекту: Ігровий формат може призвести до короткочасної мотивації, але не завжди забезпечити глибоке засвоєння матеріалу.
- Можливість переваги в розумінні гри над матеріалом: Деякі учні можуть акцентувати увагу на веселоощах гри, замість глибокого розуміння предметного матеріалу.

Swift Playground:

Переваги:

- Навчання забезпечене інтерактивністю: Це інтерактивне середовище, де можна навчатися програмуванню у формі ігор та вправ, що зробить процес навчання цікавим та захоплюючим.

- Простота використання: Платформа призначена для початківців та дозволяє швидко ознайомитися з основами Swift та програмування.

- Миттєвий зворотний зв'язок: Користувачі отримують негайний результат своєї роботи, бачачи, як їхні зміни в коді впливають на результат в реальному часі.

- Візуалізація концепцій: Інтерактивність допомагає візуалізувати складні концепції, що полегшує їх розуміння.

- Безкоштовність та доступність: Доступний безкоштовно для користувачів пристроїв від Apple, що робить його доступним для широкого кола користувачів.

Недоліки:

- Обмежена функціональність: Swift Playgrounds призначений для навчання базовим рівням мови Swift, тож для більш складних проектів він може бути обмеженим.

- Не підходить для всіх вікових груп: Хоча Swift Playgrounds створено таким чином, щоб залучити початківців, він може бути надто простим для більш досвідчених розробників.

- Залежність від пристроїв від Apple: Цей інструмент доступний лише для пристроїв від Apple, тому для користувачів інших платформ він може бути недоступним.

- Обмежений доступ до ресурсів: У порівнянні з повноцінною розробкою на комп'ютері, доступ до ресурсів (таких як бібліотеки, інструменти, розширені можливості) може бути обмеженим.

2.2. Вибір засобів та технологій розробки

Для розробки Android додатку для полегшення навчання та сприяння активного залучення студентів можна використати будь-яку IDE, що підтримує

розробку Android додатків. Інтегровані середовища розробки (IDE) грають ключову роль у створенні Android-додатків, надаючи розробникам потужні інструменти для написання, тестування та вдосконалення своїх програм.

Android Studio



Рис. 2.10 - Логотип IDE Android Studio

Android Studio — інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року на конференції Google I/O менеджером по продукції корпорації Google — Еллі Паверс (англ. Ellie Powers). 8 грудня 2014 року компанія Google випустила перший стабільний реліз Android Studio 1.0

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

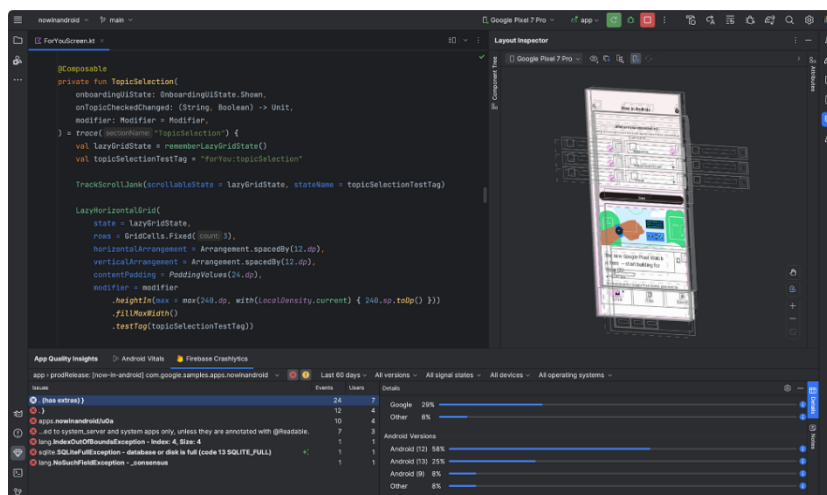


Рис. 2.11 Інтерфейс IDE Android Studio

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.[10]

NetBeans



Рис. 2.12 – Логотип IDE NetBens

NetBeans IDE — вільне інтегроване середовище розробки (IDE) для мов програмування Java, JavaFX, C/C++, PHP, JavaScript, HTML5, Python, Groovy. Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації. Середовище розробки NetBeans за замовчуванням підтримує розробку для платформ J2SE і J2EE.

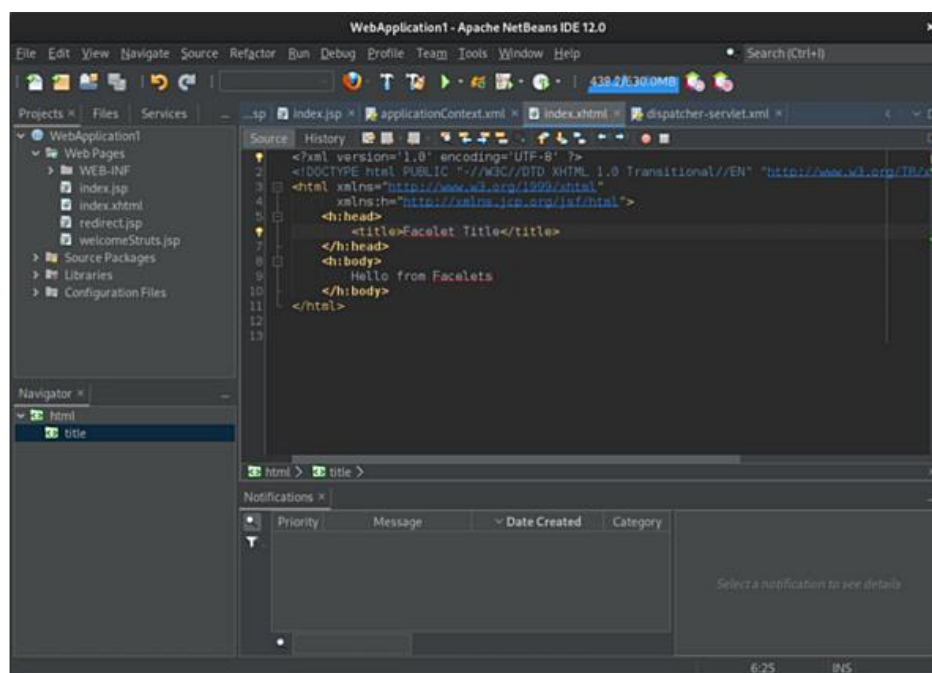


Рис. 2.13 - Інтерфейс IDE NetBeans

У випуску 7.4 у жовтні 2013 продовжено розвиток засобів для розробки з використанням технологій HTML5, додана підтримка створення гібридних HTML5-застосунків для платформ Android і Apple iOS з використанням фреймворку Apache Cordova, реалізовані засоби використання HTML5 в проєктах Java EE і PHP, представлена експериментальна підтримка майбутнього випуску JDK8.

Visual Studio with Xamarin:

Xamarin — американська компанія в галузі розробки ПЗ. Займається розробкою і підтримкою Mono та інструментів для розробки застосунків мовою C# для iOS, Android, Windows, Mac.



Рис. 2.14 - Логотип Xamarin

Xamarin стверджує, що є єдиною IDE, яка дозволяє розробляти додатки для Android, iOS і Windows у Microsoft Visual Studio. Компанія Xamarin постачає додаткові модулі до Microsoft Visual Studio, що дозволяє розробникам створювати програми для Android, iOS та Windows у середовищі IDE за

допомогою завершення коду та IntelliSense. Xamarin для Visual Studio також має розширення в Microsoft Visual Studio, які надають підтримку для створення, розгортання та налагодження програм на тренажері або пристрої. Наприкінці 2013 року компанія Xamarin і Microsoft оголосили про партнерство, яке включало подальшу технічну інтеграцію та програми для клієнтів, щоб вони могли створювати спільні бази розробників для всіх мобільних платформ.

Java

Java - об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.



Рис. 2.15 - Логотип Java

При створенні Android (операційної системи з відкритим кодом) Google взяла за основу Java. Незважаючи на те, що операційна система, збудована на ядрі Linux, була написана переважно на C, Android SDK використовує Java для розробки застосунків для платформи Android.[18]

Java широко застосовується у Android розробці, оскільки є основою для самого Android SDK.

Kotlin

Kotlin (Котлін) — статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Також компілюється в JavaScript. Мову названо на честь острова Котлін у Фінській затоці, на якому розміщена частина Кронштадту.



Рис. 2.16 - Логотип мови Kotlin

Автори ставили перед собою ціль створити лаконічнішу та типобезпечнішу мову, ніж Java, і простішу, ніж Scala. Наслідками спрощення, порівняно з Scala стали також швидша компіляція та краща підтримка IDE.

Kotlin є відносно новою мовою програмування, але вже користується широкою популярністю у розробці Android додатків. Kotlin підтримує зворотній виклик мови Java, тобто в одному проекті можна використовувати як Kotlin так і Java.[6]

Retrofit

Retrofit - це відома серед Android-розробників бібліотека для мережевої взаємодії, деякі навіть вважають її до певної міри стандартом. Причин для такої популярності безліч: бібліотека відмінно підтримує REST API, легко тестується і налаштовується, а запити по мережі з її допомогою виконуються дуже просто.



Рис. 2.17 - Логотип бібліотеки Retrofit

Room

Room - це новий спосіб зберегти дані додатків в Android-програмі, представлений цього року на Google I/O. Це частина нової Android Architecture, група бібліотек від Google, які підтримують доречну архітектуру програм. Room пропонується як альтернатива Realm, ORMLite, GreenDao і багатьом іншим.

Для розробки додатку для полегшення навчання та активного залучення студентів потрібні сучасні та корисні інструменти, у користування яких є досвід та впевненість в можливості реалізації тих чи інших функцій, що стануть у нагоді або є основними при створенні додатку для поставленого завдання.

Такими інструментами було обрано такі засоби, як Android Studio, Kotlin, Room, Retrofit, mockapi.io.

Android studio було обрано через великий досвід розробки саме у цій IDE, вона рекомендована компанією Google, власницею платформи Android, має багато влаштованих інструментів для якісної розробки програмного забезпечення і підтримує безліч, розроблених спільнотою плагінів, які ще більше полегшують або відкривають нові шляхи розробки.

Kotlin було обрано, як більш лаконічну та зручну для розуміння мову, крім того вона підтримує виклик функцій мови Java напряму з будь-якої точки коду, також Java є основною мовою розробки на Android.

Room було обрано, оскільки це зручний інструмент для роботи з базами даних.

Retrofit було обрано, як найсучаснішу бібліотеку для роботи з веб-сервісами.

Mockapi.io було обрано через легкість створення даних на веб-сервісі, та отримання доступу до цих даних.

2.3. Проектування внутрішньої структури додатку

При розробці будь-якого програмного забезпечення дуже важливо правильно підібрати архітектуру та структуру додатку, оскільки від цього буде залежати складність розробки та підтримки додатку у майбутньому.

У підрозділі 2.3 ми зосередимось на проектуванні внутрішньої структури Android-додатку, що спрямований на полегшення процесу навчання студентів. Цей етап роботи передбачає детальний розгляд архітектурних виборів, визначення основних модулів, їх взаємодії та способів реалізації функціональності для досягнення мети дослідження. Важливою складовою цього етапу є увага до зручності користування додатком, його ефективності та

можливості майбутнього розвитку. Ми розглянемо основні кроки та рішення, спрямовані на створення добре структурованого додатку, який відповідатиме потребам користувачів у сфері освіти та навчання.

Існує багато різних архітектур для створення додатків, MVC, Singleton, MVVM та багато інших. Серед них, як основу для розробки Android-додатку для полегшення навчання та активного залучення студентів було обрано паттерн Model-View-ViewModel(MVVM).

Model-View-ViewModel — це шаблон проектування додатків, який використовується в розробці програмного забезпечення. Основні компоненти MVVM: Model - модель даних додатка. Зазвичай це класи, які описують дані та бізнес-логіку; View - представлення даних користувачеві. Це можуть бути екрани, форми, сторінки тощо; ViewModel - клас, який з'єднує Model та View. ViewModel отримує дані з Model та надає їх у вигляді, зручному для відображення у View. Також обробляє події користувача.

Основні переваги MVVM:

- Розділення коду на логічні частини, що полегшує супровід та тестування.
- Незалежність View від змін в Model. View знає тільки про ViewModel.
- Легкість у тестуванні ViewModel, окремо від View та Model.
- ViewModel може містити бізнес-логіку, що дозволяє звільнити Controller від надмірної складності.

MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), відомої як модель (можна також сказати, що це відокремлення представлення від моделі). Модель представлення є частиною, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати патерн медіатор,

організовуючи доступ до бек-енд логіки навколо множини правил використання, які підтримуються представленням(рис 2.20).

MVVM була створена з метою поділу праці дизайнера і програміста, що є неможливим, коли Java-розробник намагається побудувати GUI в Swing або розробник на Visual C++ намагається створити користувацький інтерфейс в MFC. Розробники кмітливі і мають безліч навичок, але створення зручних і привабливих інтерфейсів вимагає абсолютно інших талантів, ніж ті, якими вони володіють. Ця робота більше підходить для дизайнерів інтерфейсів. Хороші дизайнери інтерфейсів краще знають, чого бажають користувачі, ніж експерти в області проєктування і написання коду. Зрозуміло, буде краще, якщо дизайнер інтерфейсів створить інтерфейс, а розробник напише код, який реалізує логіку цього інтерфейсу, але технології типу Swing або MFC просто-напросто не дозволяють чинити таким чином.[7]

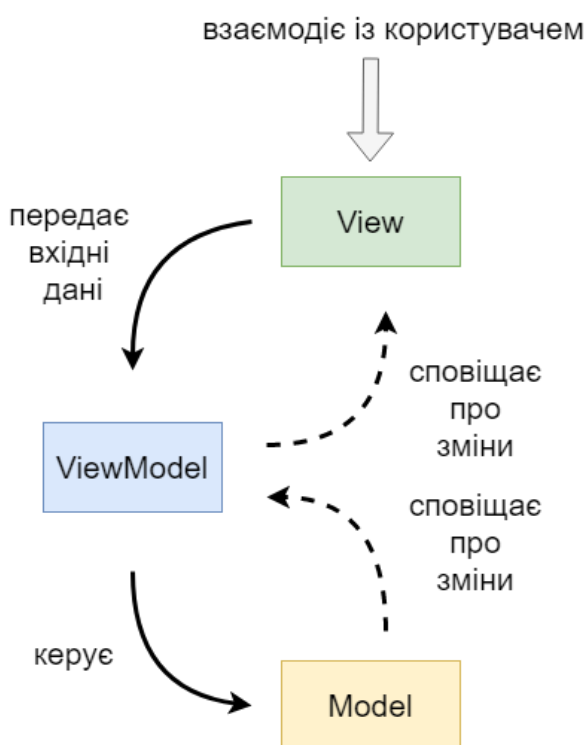


Рис. 2.20 – Діаграма роботи патерну MVVM

Однією з великих проблем із записом у коді всієї або більшої частини логіки та поведінки представлення є те, що логіку, розташовану там, просто неможливо перевірити. Переплітання погляду з його поведінкою означає, що обидва добре знають одне одного. Цей двонаправлений жорсткий зв'язок робить

тестування розглядання поведінки неймовірно складним, якщо не неможливим, завданням. Навіть якщо ви не пишете модульні тести для свого коду (хоча вам варто наполегливо це зробити) і не вважаєте можливість тестування проблемою, реалізація MVVM у вашій програмі все ще забезпечує багато інших переваг:

- Паттерн дозволяє створювати змішувані проекти, дозволяючи легко заповнювати ваші перегляди з даними під час проектування, щоб дизайнер міг бачити, що таке інтерфейс користувача виглядає як заповнений даними без необхідності компілювати та запускати проект.
- Паттерн покращує робочий процес дизайнера/розробника, зменшуючи помилки, коли обидві сторони одночасно працюють над виглядом, поведінкою та логікою погляду, не заважаючи одна одній.
- MVVM - це крок до структурування вашого проекту для підтримки впровадження принципу інверсії управління та введення залежностей.
- MVVM дозволяє потенційно повторно використовувати логіку перегляду в кількох різних переглядах реалізації.
- Слабкий зв'язок дозволяє потенційно застосовувати різні Views до одного ViewModel. Наприклад, ви можете повторно використовувати свої ViewModels із Silverlight проект у версії Windows Phone 7 із застосуванням різних переглядів відповідно до Елементи управління та форм-фактор Windows Phone 7.
- Навіть якщо ви вирішите не писати модульні тести для свого проекту, реалізувавши MVVM у вашому проекті дозволяє пізніше модернізувати ваш проект модульними тестами часу, не вимагаючи занадто великої (якщо така є) реструктуризації.[2]

Важливою складовою правильної структури та архітектури проекту є чистота та лаконічність коду. Правильний код з легкістю стає читабельним, зрозумілим, як для автора, так і для тих хто буде продовжувати роботу або робити рев'ю, чи просто давати приклади. Під добре написаним кодом мається на увазі те, що усі змінні мають мати своє ім'я, яке буде «кричати» для чого створена та чи інша змінна. Назви функцій мають коротко відображати суть того, що у них

буде відбуватись. Також не можна допускати розростання функцій до великих масштабів, у випадках, коли таке трапляється їх потрібно ділити на менші. Те саме має бути і з класами, їх не можна перегружати.

Гейміфікація в контексті проектування внутрішньої структури додатку стає потужним інструментом для підвищення мотивації користувачів і покращення їхнього досвіду взаємодії з платформою. Інтеграція елементів гри, таких як завдання, бейджі, рівні та конкуренція, в процесі взаємодії з додатком стимулює активність користувачів і сприяє позитивній динаміці використання. Метою впровадження гейміфікації є не лише надання функціональності але і створення захоплюючого середовища, яке надихає на досягнення цілей. Застосування такого підходу у внутрішній структурі додатку дозволяє зберегти користувачів, створити унікальний ідентифікаційний досвід та підвищити рівень зацікавленості у використанні платформи. Персоналізовані завдання, система винагород, можливість спілкування та співпраці в грі роблять гейміфікацію важливим елементом при створенні привабливого та ефективного додатку.

Проектуючи додаток, було враховано та проаналізовано аналоги, враховано випадки з власного досвіду та проведено міні опитування. В результаті досліджень було сформовано оптимальну схему роботи додатку. Вона включає в себе: вікторини, розклад, рейтингова таблиця, лекції, корисні відео, контакти з вчителем, кабінет, досягнення та новини. (рис 2.21) Ці пункти є одними з найкорисніших для імплементації в Android-додатку для полегшення навчання та активного залучення студентів.

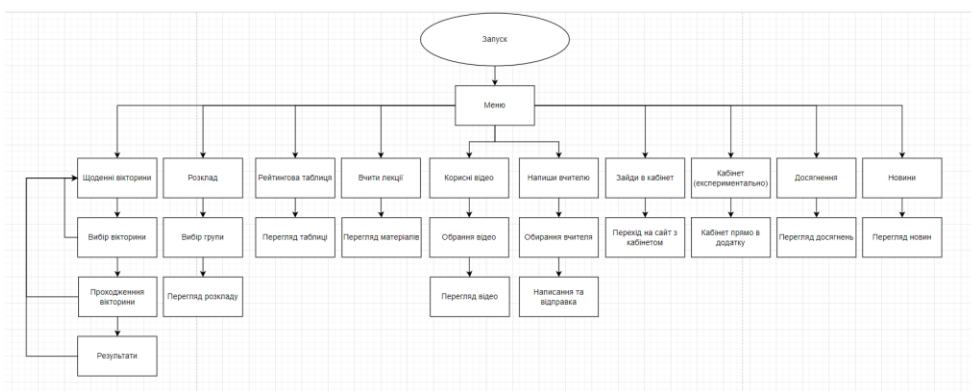


Рис. 2.21 Алгоритм роботи додатку

Такі пункти, як щоденні цікаві вікторини не тільки допоможуть залучити студентів, але й в процесі їх проходження покращити та, навіть, закріпити отримані знання. Розклад – невід’ємна частина життя студента, йому постійно треба знати свій розпорядок і тому він буде частіше звертатись в додаток та з більшою вірогідністю відвідуватиме такі пункти як вивчення лекцій, відео та вікторин. Рейтингова таблиця: прямий ефір результатів роботи студентів допоможе розвину певну конкурентність між студентами, оскільки змагання за кращі бали та позиції в списку призведуть до активної участі в навчальних процесах, що не тільки покращить загальні бали студентів, але й неабияк підвищить якість знань. Лекції під рукою завжди корисні, коли потрібно звернутись до якогось матеріалу швидко. Корисні відео – пункт з відео, які будуть оновлюватись з певним періодом є втіленням мультимедійності в процес навчання. Контакти з вчителем – база даних пошт вчителів прямо в телефоні, потрібно просто обрати вчителя та написати йому листа. Кабінет в телефоні, що завжди під рукою та не потребує участі браузера це дуже зручно. Найбільша кількість аудиторії очікується через введення кабінету прямо в додаток, а не через сайт. Реалізація кабінету в додатку дозволить зберігати незмінні дані, такі як групи, ПБ, ступені, форми навчання, спеціальності, предмети і т.д. За своєчасні та правильно пройдені вікторини передбачається нагорода, а саме пункт з досягненнями, отриманими за ті чи інші заслуги. Також цікавим є пункт новин університету(чи будь якого навчального закладу), прямо в додатку можна буде переглядати, що нового відбулось у світі та університеті.

Висновки до другого розділу.

У розділі 2 було описано аналоги існуючих додатків та платформ для полегшення навчання та активного залучення студентів. Було перелічено їх переваги та недоліки. Був проведений аналіз засобів та технологій розробки, серед них було обрано ті, що будуть використані у додатку. Також було проведено проектування внутрішньої структури проекту, від архітектури додатку до приблизного функціоналу.

На основі аналізу аналогів та порівняння їх переваг та недоліків з'являється можливість запозичити та застосувати найкращі методики та інструменти для досягнення основної мети дослідження, а також проаналізувати їх прямий вплив на популярність та важкість навчального процесу

У розділі 2.2 було проаналізовано та вивчено певний набір засобів та технологій, що будуть використані або стануть у нагоді. Завдяки правильно обраним засобам розробка та тестування повинна бути якісною та швидкою.

У розділі 2.3 було спроектовано внутрішню структуру додатку. Було згадано декілька архітектур, обумовлено важливість застосування методик гейміфікації, було обґрунтовано та зроблено вибір паттерну MVVM, було згадано про важливість чистого та якісного коду. Також було сформовано певний алгоритм роботи програми з приблизним функціоналом

Метою даного розділу було провести аналіз аналогів додатку, проаналізувати їх переваги та недоліки щоб застосувати кращі сторони та методики у власній реалізації Android-додатку для полегшення навчання та активного залучення студентів. Також на меті стояла проблема вибору засобів та технологій розробки додатку на платформі Android, серед них було обрано найпідходящі та зручні. Останньою ціллю цього розділу було спроектувати внутрішню структуру додатку, визначити архітектуру та приблизний функціонал.

Отже, підсумовуючи розділ, розглянувши аналоги, ми отримали найрезультативніші методики для спрощення навчального процесу та активного залучення студентів, завдяки огляду засобів та технологій, ми визначились з інструментами, що допоможуть створити корисний додаток швидко та якісно. Спроектуювавши внутрішню структуру проекту ми отримали приблизний план та почали розробку проекту.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ

3.1. Імплементация гейміфікації у додатку

Імплементация гейміфікації у додатку представляє собою ключовий аспект реалізації та оптимізації користувацького досвіду в процесі навчання. За допомогою впровадження гейміфікованих елементів, таких як завдання, бали, рівні та бейджі, з'являється мета створення захопливого та стимулюючого середовища для студентів.

Для імплементации методів гейміфікації у Android-додатку для полегшення навчання та сприяння активному залученню студентів було обрано такий перелік моделей та методів реалізації гейміфікації як моделі: бейджі, рейтинги, прогрес-бари та соціальна гейміфікація; методи: гра, бали та досягнення, ранги та рейтинги, місії та завдання та соціальна інтеракція

У кабінеті студента реалізовано функціонал профілю, де відображається інформація з кабінету студента університета, а також тут реалізовано відображення балів, досягнень та завдань, за які також можна отримати бали. Завдання можуть включати в себе проходження вікторин, або відкриття екрану. Наприклад при першому запуску в кабінеті відображається завдання «Повчити лекції», задачою якого є перейти на екран з лекціями.

На екрані з вікторинами, що є реалізацією методу гейміфікації «Гра» на кожній вікторині відображається бейдж, який відображає поточний результат вікторини. Після проходження вікторини, користувач отримує бали, які відображаються в кабінеті. За проходження певної серії вікторин, користувач отримує досягнення за проходження серії вікторин, яке буде відображуватись в кабінеті. Також на екрані вікторин відображаються прогрес-бари, що позначають статус проходження певної серії вікторин.

На екрані рейтингу знаходиться імплементация методики рангів та рейтингів, а саме знаходиться таблиця усіх користувачів додатку, яка відображає нараховані бали, ранги стосовно цих балів та успішність загалом. Зображення

цієї інформації повинно надихати користувачів до змагання та поліпшення своїх результатів

Реалізацією соціальної інтеракції є екран взаємодії з викладачами. На ній користувач може обрати викладача та розпочати живий чат, а викладач в тому числі може створювати цілі групи з студентами.

3.2. Розгортання та тестування моделі у вигляді Android-додатку

За результатами дослідження було спроектовано та створено Android-додаток для полегшення навчання та активного залучення студентів під назвою «Education App». Усі його функції націлені на спрощення процесу навчання та максимальної зручності для користувача.



Рис. 3.1. Splash Fragment

Після запуску додатку, юзер спочатку потрапляє на екран Splash(рис. 3.1.), після чого опиняється в головному меню, де перед ним відкривається весь функціонал додатку.

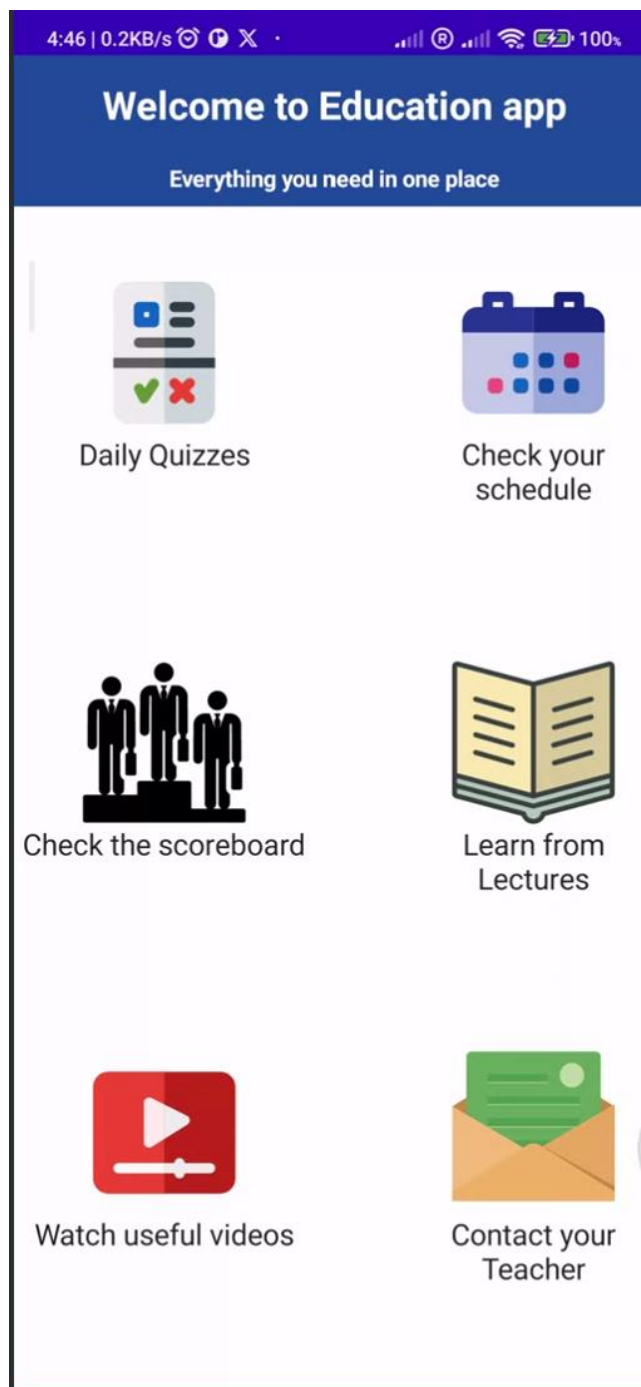


Рис. 3.2 - Головне меню додатку

На вибір перед ним постає дев'ять варіантів та один експериментальний. Наприклад почнемо з огляду та тестування пункту «Daily Quizzes». Цей фрагмент містить декілька варіантів вікторин за різними темами. За задумкою, кожен день користувачу з API має приходити або декілька JSON файлів, або просто

завантажуватись та зберігатись питання в базі даних. Після переходу на екран «Daily quizzes», користувач може обрати яку саме вікторину він хоче пройти.(рис. 3.3)

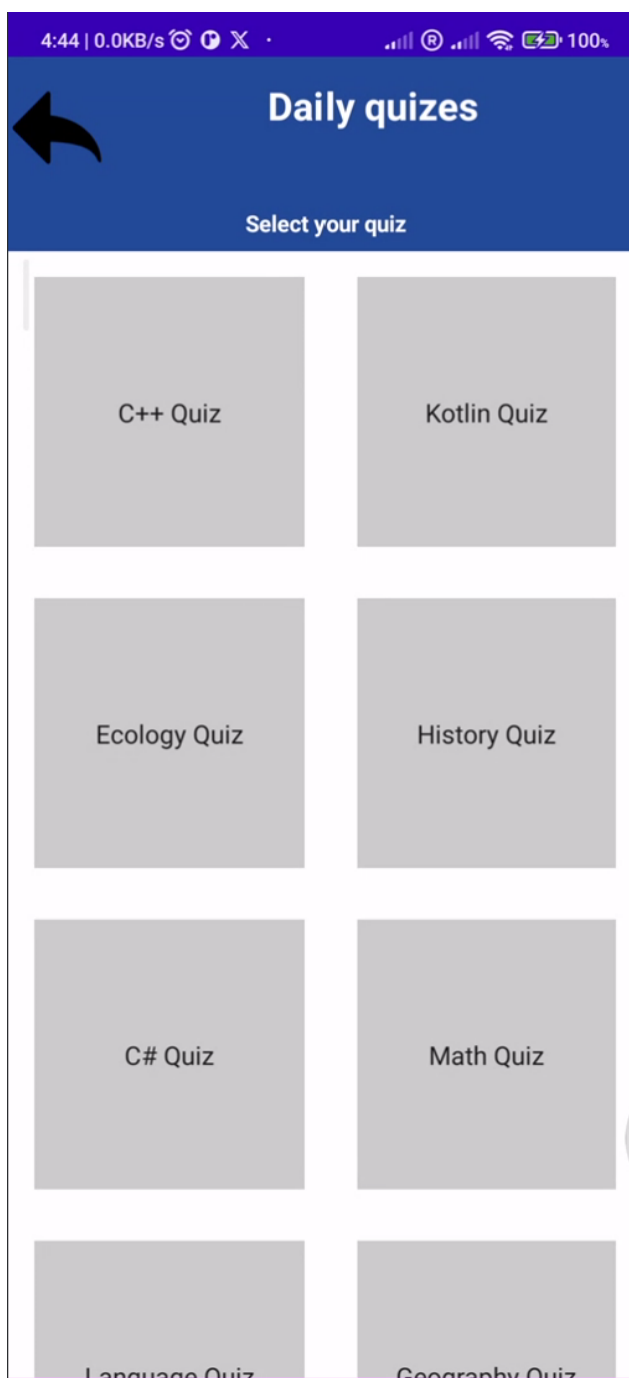


Рис. 3.3 - Екран обрання вікторини.

Після обрання бажаної вікторини, власне й розпочинається сама вікторина. Кількість запитань залежить від самої вікторини, що на прислала API, проте, в умовах відсутності API, додаток черпає вікторини з локального ресурсу, а саме

JSON файлів, в яких знаходяться запитання та відповіді до них. На рисунку 3.4 наведено приклад завантаження інформації з JSON файлів.

```
private val moshi by lazy {
    Moshi.Builder()
        .addLast(KotlinJsonAdapterFactory())
        .build()
}

private val adapter = moshi.adapter(QuestionsDTO::class.java)

val question = Channel<List<QuestionDTO>>(Channel.CONFLATED)

suspend fun loadQuestions(jsonInputStream: InputStream) {
    val questions = withContext(Dispatchers.IO) {
        kotlin.runCatching {
            jsonInputStream
                .bufferedReader(Charsets.UTF_8)
                .use { it.readText() }
                .let { rawJson ->
                    adapter.fromJson(rawJson)?.questions ?: throw Exception("Could not parse questions")
                }
        }
    }.getOrNull()
    question.send(questions)
}
```

Рис. 3.4 – Функція зчитування даних з JSON файлів.

Таким самим чином, не змінюючи багато коду, після підключення працюючої API, ми зможемо без проблем зчитувати нові вікторини.

Процес проходження вікторини полягає у підборі правильних відповідей на запитання та натиску на кнопки наступне, назад або вийти. (рис. 3.5.)

Рис. 3.5 – Процес проходження вікторини

Після закінчення процесу проходження, користувачу показується отриманий результат та пропонується вийти на початкову сторінку.(рис. 3.6)

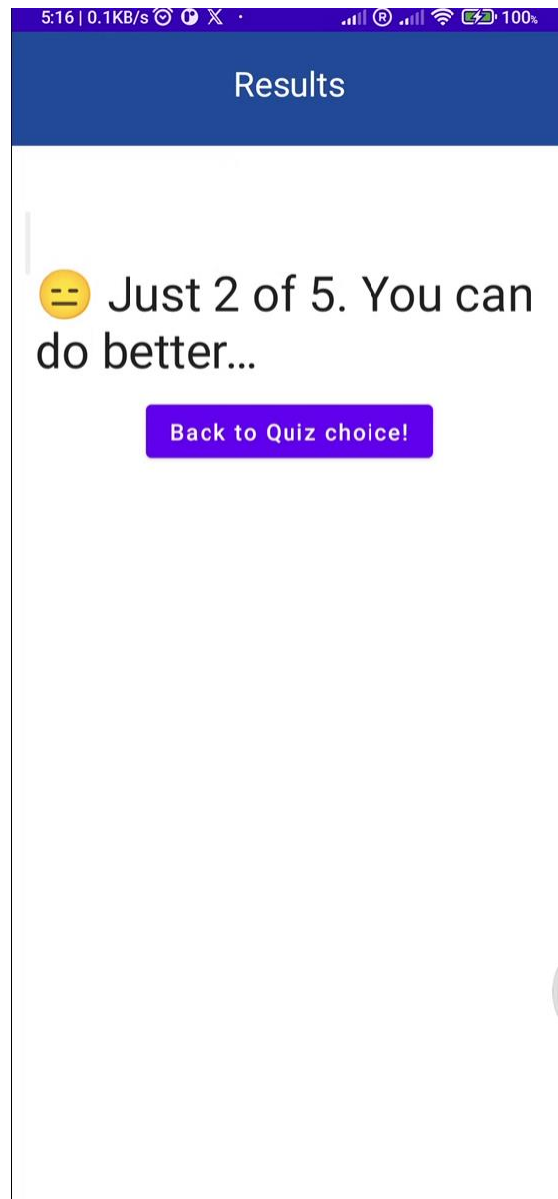


Рис. 3.6 – Результат проходження вікторини

Наступною функцією для розгляду буде функція відображення розкладу за групою. У головному меню, користувач натискає на пункт «Check your schedule», після чого одразу потрапляє на екран, де відображається розклад групи за замовчуванням.(рис. 3.7)

Розклад занять Житомирської політехніки

Розклад групи BI-23-1

I тиждень

Сьогодні

Понеділок	Вівторок	Середа	Четвер	П'ятниця
8:30-9:55				BI-23-1, TP-23-1 Фізика
10:00-11:30				Практика, 277 Колісник Роман Олександрович
11:40-13:00	BI-23-1, TP-23-1 Фізичне виховання Практика, 322 Спортивний 1 Данилюк Сергій Миколайович	Фізика Лекція, 322 Колісник Роман Олександрович	BI-23-1, TP-23-1, IB-23-1 Обчислювальна техніка та програмування Лекція, 248 Морозов Дмитро Сергійович	BI-23-1, TP-23-1, IB-23-1 Інформаційна технологія Лекція, 4019 Парчук Катерина Ірина
13:30-14:55	Медична інформатика Практика, 122 Насічук Тетяна Миколаївна	Фізика Лекція, 322 Колісник Роман Олександрович	BI-23-1, TP-23-1, IB-23-1 Висока математика Практика, 428 Степанів Роман Миколайович	AI-1, IB-23-1, IB-23-1, IB-23-1, ET-2, TP-23-1, AT-30 Висока математика Лекція, 334 Бондарчук Василь Миколайович
15:00-16:25	Медична інформатика Лекція, 120 Насічук Тетяна Миколаївна	BI-23-1, TP-23-1 Українська мова, професійне та академічне письмо Практика, 315 Білошицька Зоя Анатоліївна	BI-23-1, TP-23-1, IB-23-1 Фізика Лекція, 334 Колісник Роман Олександрович	AI-1, IB-23-1, IB-23-1, IB-23-1, ET-2, TP-23-1, AT-30 Висока математика Лекція, 334 Бондарчук Василь Миколайович
16:30-17:55	BI-23-1, TP-23-1 Українська мова, професійне та академічне письмо Практика, 315 Білошицька Зоя Анатоліївна	BI-23-1, TP-23-1, IB-23-1 Комп'ютерна база електронних апаратів Лекція, 428 Колісник Роман Олександрович	BI-23-1, TP-23-1, IB-23-1 Комп'ютерна база електронних апаратів Лекція, 428 Колісник Роман Олександрович	BI-23-1, TP-23-1, IB-23-1 Обчислювальна техніка та програмування Лекція, 248 Морозов Дмитро Сергійович

II тиждень

Понеділок	Вівторок	Середа	Четвер	П'ятниця
8:30-9:55				BI-23-1, TP-23-1, IB-23-1
10:00-11:30				Обчислювальна техніка та програмування Лекція, 248 Морозов Дмитро Сергійович
11:40-13:00	BI-23-1, TP-23-1 Українська мова, професійне та академічне письмо Практика, 315 Білошицька Зоя Анатоліївна	BI-23-1, TP-23-1, IB-23-1 Інформаційна технологія Лекція, 428 Колісник Роман Олександрович	BI-23-1, TP-23-1, IB-23-1 Комп'ютерна база електронних апаратів Лекція, 428 Колісник Роман Олександрович	BI-23-1, TP-23-1, IB-23-1 Обчислювальна техніка та програмування Лекція, 248 Морозов Дмитро Сергійович
13:30-14:55	Медична інформатика Практика, 122 Насічук Тетяна Миколаївна	Фізичне виховання Практика, 322 Спортивний 2 Данилюк Сергій Миколайович	Медична інформатика Лекція, 122 Насічук Тетяна Миколаївна	Комп'ютерна база електронних апаратів Лекція, 428 Колісник Роман Олександрович

Рис. 3.7 – Сторінка перегляду розкладу обраної групи

Реалізація такої функції у такому форматі стала можливою завдяки існуванню в Android таких функцій як `WebView` та `ExposedDropDownMenuBox`(рис. 3.8)

```

AndroidView(
    factory = { context ->
        WebView(context).apply {
            settings.javaScriptEnabled = true
            webViewClient = WebViewClient()
            settings.loadWithOverviewMode = true
            settings.useWideViewPort = true
            settings.setSupportZoom(true)
        }
    },
    update = { webView ->
        try {
            webView.loadUrl(link)
        } catch (e: Exception) {
            Toast.makeText(
                screenContext,
                "Error loading Url: ${e.message}",
                Toast.LENGTH_SHORT
            )
                .show()
        }
    }
)

ExposedDropDownMenuBox(
    expanded = expanded,

```

```

onExpandedChange = {
    expanded = !expanded
}
) {
    TextField(
        value = selectedText,
        onValueChange = {},
        readOnly = true,
        trailingIcon = { ExposedDropDownMenuDefaults.TrailingIcon(expanded = expanded) },
        modifier = Modifier.menuAnchor(),
        colors = TextFieldDefaults.textFieldColors(textColor = Color.White)
    )

    ExposedDropDownMenu(
        expanded = expanded,
        onDismissRequest = { expanded = false }
    ) {
        list.forEach { item ->
            DropDownMenuItem(
                text = { Text(text = item) },
                onClick = {
                    selectedText = item
                    text.value = "https://rozklad.ztu.edu.ua/schedule/group/$item"
                    expanded = false
                    Toast.makeText(context, item, Toast.LENGTH_SHORT).show()
                }
            )
        }
    }
}

```

Рис. 3.8 – Реалізація WebView та ExposedDropDownMenuBox

При зміні групи через випадаючий список, розклад моментально оновлюється та показує актуальну інформацію.(рис. 3.9)

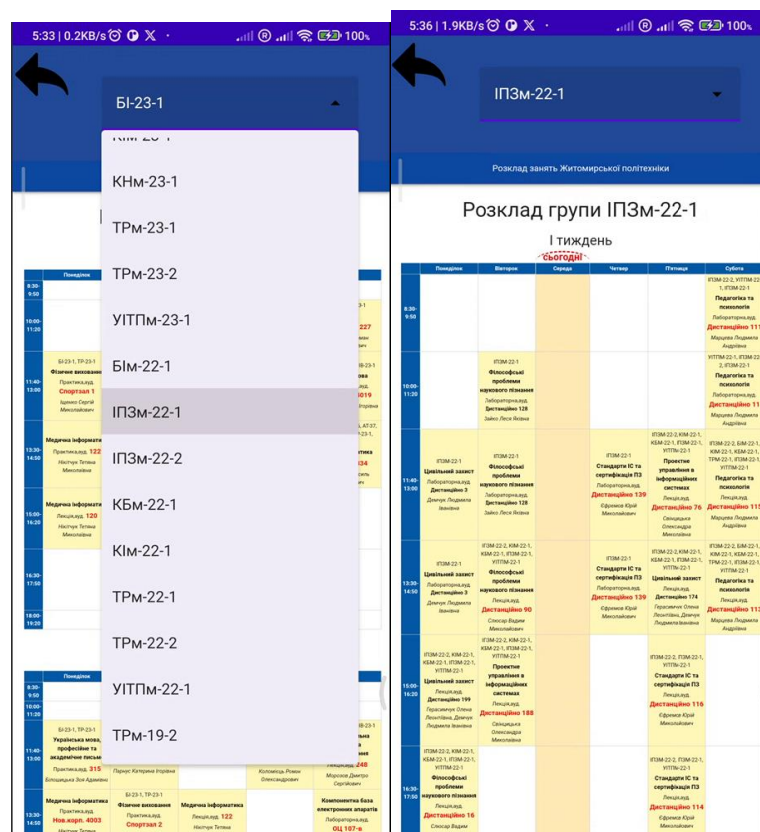


Рис. 3.9 – Демонстрація вибору групи та миттєве оновлення розкладу відповідно до обраної групи.

Продовжуємо тестування та переходимо до пункту рейтинговий список. При натисканні на пункт «Check the scoreboard», ми переходимо на рейтингову таблицю з фейковими даними, отриманими через інтернет за допомогою сервісу москарі.іо. Протягом завантаження ми бачимо Loader, який показує нам, що відбувається завантаження даних. Якщо ж виникне помилка при завантаженні даних, ми отримаємо про це повідомлення.

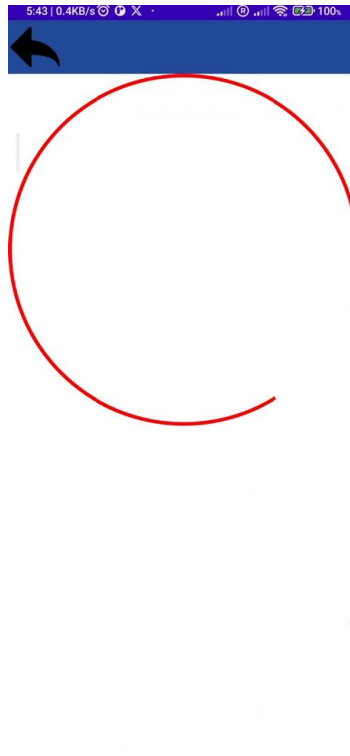


Рис. 3.10 – Завантаження даних з інтернету

Після завантаження ми отримаємо 48 записів студентів з випадковими даними. Для випадкових даних було створено інформаційний клас «Студент», що включає в себе дату створення, повне ім'я, середній бал, спеціальність та номер ідентифікатор(рис. 3.11)

```
data class Student(
    @SerializedName("createdAt")
    val createdAt:String,
    @SerializedName("name")
    val name:String,
    @SerializedName("avgPoints")
    val avgPoints: String,
    @SerializedName("Grade")
    val grade: String,
    @SerializedName("id")
    val id: Int
)
```

Рис 3.11 – Інформаційний клас «Студент»

На завантаженому екрані ми можемо побачити лише потрібну нам для порівняння інформацію, таку як імена, оцінки та спеціальності(рис. 3.12.)



StudentsList		
Name:	Violet Wyman,	Average Point = 9, Grade = strategy
Name:	Bobbie Grimes,	Average Point = 9, Grade = typhoon
Name:	Miss Woodrow Wunsch,	Average Point = 9, Grade = Associate
Name:	Reginald Tillman,	Average Point = 9, Grade = Tugrik
Name:	Miss Leah Denesik,	Average Point = 9, Grade = because
Name:	Luz Muller,	Average Point = 9, Grade = ivory
Name:	Yolanda Purdy,	Average Point = 9, Grade = radian
Name:	Donnie Reilly,	Average Point = 9, Grade = Intelligent
Name:	Jamie Windler Jr.,	Average Point = 8, Grade = purple
Name:	Gloria Stiedemann,	Average Point = 8, Grade = withdrawal
Name:	Jeannette Leffler MD,	Average Point = 8, Grade = Frozen
Name:	Derek Kiehn,	Average Point = 8, Grade = Smart
Name:	Ella Frami,	Average Point = 8, Grade = Orchestrator
Name:	Spencer Tillman,	Average Point = 8, Grade = digital
Name:	Tracey Bechtelar,	Average Point = 8, Grade = Automotive
Name:	Bertha Ledner,	Average Point = 7, Grade = inventore
Name:	Darrel Satterfield,	Average Point = 7, Grade = blue
Name:	Wilma Maggio,	Average Point = 7, Grade = lumen

Рис. 3.12 – Таблиця рейтингу

Дані завантажуються, за допомогою бібліотеки Retrofit, запит складається з декількох строк коду, спочатку ініціалізуються OkHttpClient клієнт, потім Retrofit клієнт, після чого вже ініціалізується сам ручний ApiClient, через який ми робимо запити в інтернет.(Рис. 3.13 – 3.14)

```
@GET("https://657f6af96ae0629a3f53502c.mockapi.io/ap1/v1/Students")
suspend fun getStudents():List<Student>
```

Рис. 3.13 – Функція в класі ApiClient

```
@Provides
@Singleton
fun provideOkHttpClient(): OkHttpClient {
    return OkHttpClient.Builder().build()
}

@Provides
@Singleton
fun provideRetrofit(okHttpClient: OkHttpClient): Retrofit {
    return Retrofit.Builder().baseUrl("https://google.com")
        .client(okHttpClient)
```

```

        .addConverterFactory(GsonConverterFactory.create())
        .build()
    }

    @Provides
    @Singleton
    fun provideApiService(retrofit: Retrofit): ApiService {
        return retrofit.create(ApiService::class.java)
    }

```

Рис. 3.14. Ініціалізація бібліотек

Цікавою функцією є корисні відео. Після обрання користувачем даного пункту меню він переходить на екран, де будуть відображатись відео, посилання та інформація про яких надходять також з API. Нажаль, влаштований плеєр став застарілим та його заборонено використовувати аби уникнути витоків інформації, тому функція не повністю реалізована, при обранні відео, користувача одразу переносить в додаток Youtube або в браузер, якщо останнього немає на телефоні.(рис. 3.15.)



Рис. 3.15 – Сторінка з цікавими відео

Також у додатку реалізовані функції миттєвого переходу до матеріалів для навчання, а саме сайту learn.ztu.edu.ua при натиску на пункт «Learn from Lectures». При виборі розташованих нижче функцій, а саме «OpenYour Cabinet», «Cabinet(Experimental)», «Achievements» та «University News» ми також отримуємо корисну інформацію.(рис 3.16.)

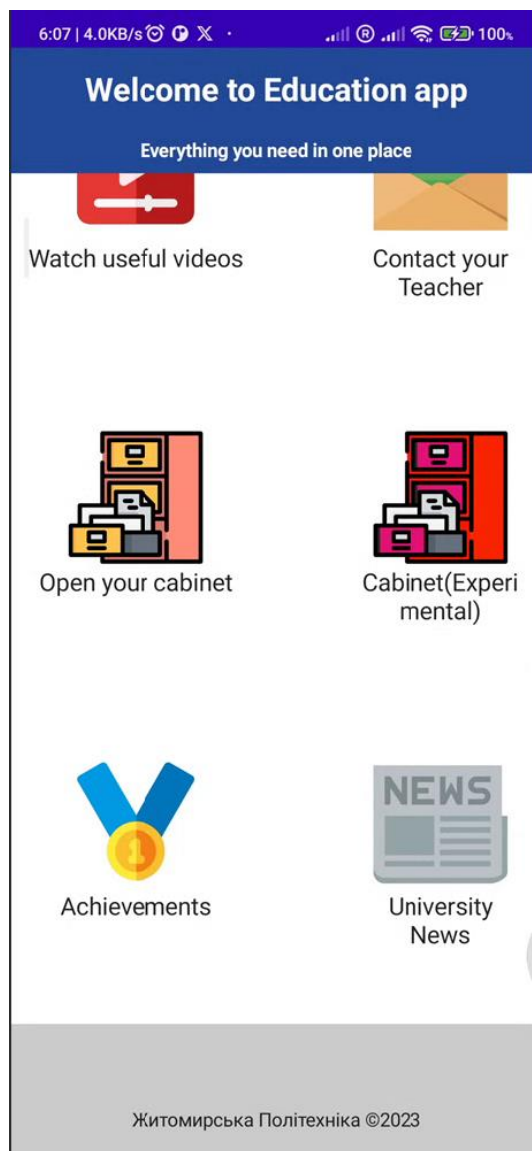


Рис. 3.16 – Додаткові функції

Обравши пункт «Open your cabinet», користувача перекине в браузер на сайт cabinet.ztu.edu.ua, де він зможе керувати та здійснювати діяльність зі своїм профілем. Якщо ж обрати пункт Cabinet(Experimental), то користувач потрапить у не підключений до API приклад кабінету, але не на сайті, а вже всередині додатку, тобто зможе виконувати дії або керувати чимось у своєму кабінеті прямо в додатку, що є неабияк зручно. Нажаль без справжньої API, неможливо

повністю реалізувати функціонал кабінету в середині додатку. Справжня API надала б можливість реалізації логіну в додатку, що з собою в свою чергу підтягувало б і персоніфіковані вікторини, відео, одразу показувало б правильну групу користувача, та відкрило б доступ до повноцінної реалізації досягнень. Останній пункт – новини, додаток наразі просто перекидає користувача у браузер, але з написаною API, цей розділ зможе відображати пости в реальному часі.

3.3. Оцінка та аналіз ефективності роботи додатку.

Розгорнувши та протестувавши Android-додаток для полегшення навчання та сприяння активного залучення студентів також було проведено аналіз та оцінку ефективності роботи додатку. Завдяки тому, що в додатку вже реалізовано і буде дороблено в майбутньому можна без вагань сказати, що ефективність роботи додатку є на високому рівні. Серед усіх присутніх функцій, назвати найбільш популярними та притягуючими до себе користувачів є кабінет в додатку, розклад та вікторини.

За допомогою такого сервісу як Firebase ми зможемо абсолютно безкоштовно відслідковувати дії користувачів у додатку, тривалість сесій, несподівані завершення роботи додатку та багато іншого, що в цілому дає нам великий відгук про ефективність додатку.[11]

Купа студентів перейде в цей додаток тільки для того, щоб зручно користуватись сервісами університету, такими як кабінет та розклад. Зручність додатку в цьому плані просто не передати. Коли сайт може бути недоступним, кабінет в додатку відображатиме збережену інформацію, що дає великий плюс до автономності. При збоях на серверах студент також залишиться з інформацією, яка йому потрібна.

В цілому оцінка ефективності роботи додатку є досить високою, оскільки майже кожна людина наразі має під рукою мобільний телефон і не буде проти від зручного додатку, що не лише полегшить процес навчання в університеті, а й урізноманітнить його.

Висновки до третього розділу

У останньому розділі під назвою «ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ», присвяченому детальному аналізу та огляду розробленого Android-додатку для полегшення навчання та сприяння активного залучення студентів, було розглянуто пророблену роботу, підходи та реалізації, описано імплементацию моделей, методів та підходів гейміфікації. У ході тестування було знайдено та усунено баги. Використання архітектурного паттерну MVVM полегшило тестування.

Було детально описано яким чином було імплементовано моделі, методики та підходи імплементации гейміфікації.

Було проведено детальне тестування кожного компоненту програми, важливі функції були описані та показані в рисунках. Оглянуто кожен екран, протестовано кожну функцію. Усі знайдені недоліки було усунуто.

Також було проведено оцінку та аналіз ефективності роботи додатку, що повернула високий результат ефективності. Додаток зможе залучати більше студентів до навчання та не тільки набагато полегшити їм навчальний процес а й зробити його цікавішим

ВИСНОВОК

У цій магістерській кваліфікаційній роботі було успішно розроблено, протестовано та досліджено Android-додаток для полегшення навчання та сприяння активного залучення студентів.

Встановлена мета, актуальність теми та завдання дослідження дали можливість правильно, швидко та якісно розробити правильний підхід до магістерської роботи.

У першому розділі було проведено аналіз предметної області, спрямований на визначення напрямків в полі полегшення навчання та сприяння активному залученню студентів. Цей аналіз дав нам загальне поняття та показав поточну ситуацію у сфері освіти. Під час аналізу існуючих рішень було виявлено ключові проблеми та можливості для подальшого розвитку. Було перераховано найпопулярніші методи полегшення навчання та підвищення залученості студентів до навчального процесу. Важливим пунктом цього розділу було дослідження основних моделей, методів та підходів гейміфікації. Опис Постановка задачі дослідження чітко визначила цілі та завдання, що спрямовані на створення ефективного Android-додатку.

У другому розділі було проведено проектування Android – додатку для полегшення навчання та сприяння активного залучення студентів. Аналіз аналогів додатку дозволив нам знайти вже готові популярні та дієві рішення, які ми могли б запозичити та імплементувати у нашому додатку. Також ми співставили всі переваги й недоліки, що також повпливало на вибір шляху розробки додатку. Було оглянуто та обрано найкорисніші та зручні засоби розробки, що дозволять швидко та якісно виконати поставлені завдання. Ми спроектували внутрішню структуру додатку, обрали правильну архітектуру та зобразили приблизний алгоритм роботи програми.

У третьому розділі ми розгорнули та протестували модель у вигляді Android-додатку. Було детально описано імплементацию моделі, методики та підходи імплементации гейміфікації. В процесі проведення тестів та розгортання

додатку було виявлено, проаналізовано та усунуто причини появи недоліків. Було описано декілька функцій. Також було проведено оцінку та аналіз ефективності роботи додатку, в результаті якої було визначено, що додаток виконує поставлені завдання, а саме полегшує навчальний процес та сприяє активному залученню студентів.

Отже, у результаті виконання магістерської кваліфікаційної роботи було успішно розроблено, досліджено та протестовано Android-додаток для полегшення навчання та сприяння активного залучення студентів. Висновки підкреслюють важливість проведеної роботи. Важливо зазначити, що існує необхідність у подальшому дослідженні проблеми, застосування масових соціальних теорій, що повинні призвести до ще більшого зацікавлення у сфері навчання за допомогою Android-додатків, або мобільних пристроїв в цілому, чи то електронний годинник, чи електричний чайник, чи розумні окуляри.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ляшенко О. І. Адаптивне навчання, як ознака сучасних дидактичних систем / О. І. Ляшенко., 2017. – 224 с. – (Інститут педагогіки НАПН України).
2. Chris Anderson. Pro Business Applications with Silverlight 5 / Chris Anderson., 2012. – 670 с.
3. Сергеева Л. . ГЕЙМІФІКАЦІЯ: ІГРОВІ МЕХАНІКИ У МОТИВАЦІЇ ПЕРСОНАЛУ. / Л. Сергеева // Theory and methods of educational management / Л. Сергеева., 2014. – (2). – (14).
4. Сергієнко В. П. Мультимедійні системи як засоби інтерактивного навчання. / В. П. Сергієнко, Ю. М. Богачков.. – 112 с.
5. Dawn G. Head First Android Development: A Brain-Friendly Guide / Griffiths Dawn., 2017. – 928 с. – (Head First).
6. Michael F. Kotlin and Android Development featuring Jetpack. Build Better, Safer Android Apps / Fazio Michael., 2021. – 446 с.
7. Saurabh S. Solutions Architect's Handbook: Kick-start your career as a solutions architect by learning architecture design principles and strategies / S. Saurabh, S. Neelanjali., 2022. – 590 с. – (2).
8. Murphy M. L. The Busy Coder's Guide to Advanced Android Development / Mark Murphy., 2011. – 632 с.
9. Sabine G. Intelligent and Adaptive Learning Systems: Technology Enhanced Support for Learners and Teachers / G. Sabine, L. Fuhua, Kinshuk., 2011. – 426 с.
10. Clifton C. Learn Android Studio: Build Android Apps Quickly and Effectively / C. Clifton, G. Adam., 2015. – 511 с. – (1).
11. Ashok K. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase / Kumar Ashok., 2018. – 394 с.
12. Ніколайчук М.Ю., Вакалюк Т.А. Четверта промислова революція та її складові. Комп'ютерні інтелектуальні системи та мережі: матеріали XVI Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених, 21-23 березня 2023 року. Кривий ріг: Криворізький

національний університет, 2023. С.212-214.

13. Ніколайчук М.Ю., Вакалюк Т.А. Design in programming. Комп'ютерні інтелектуальні системи та мережі: матеріали XVI Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених, 21-23 березня 2023 року. Кривий ріг: Криворізький національний університет, 2023. С.130-132.

14. Бугайчук К. Л. ГЕЙМІФІКАЦІЯ У НАВЧАННІ: СУТНІСТЬ, ПЕРЕВАГИ, НЕДОЛІКИ [Електронний ресурс] / К. Л. Бугайчук // Харківський національний університет внутрішніх справ – Режим доступу до ресурсу: <https://dspace.univd.edu.ua/server/api/core/bitstreams/ca164dcf-553f-479b-ac3d-0813da7e3183/content>.

15. Кухаренко. Конструктивізм [Електронний ресурс] / Кухаренко – Режим доступу до ресурсу: http://kvn-e-learning.blogspot.com/2011/05/blog-post_4476.html.

16. Сингаївський В. В. Використання мультимедійних технологій у навчанні. [Електронний ресурс] / В. В. Сингаївський, Ф. М. Гарєєва – Режим доступу до ресурсу: <http://surl.li/oqznw>

17. Невідомий А. Що таке активне навчання [Електронний ресурс] / Артем Невідомий – Режим доступу до ресурсу: <https://drukarnia.com.ua/articles/sho-take-aktivne-navchannya-ch-1-rYlgpkfj>.

18. Копитко М. Ф. Основи програмування мовою Java [Електронний ресурс] / М. Ф. Копитко, К. С. Іванків. – 2002. – Режим доступу до ресурсу: <https://ami.lnu.edu.ua/wp-content/uploads/2017/05/Java.pdf>.

19. Критична теорія, Конструктивізм, Функціоналізм, Фемінізм [Електронний ресурс] – Режим доступу до ресурсу: https://studies.in.ua/mvidn_seminar/501-kritichna-teorya-konstruktivizm-funkcionalizm-femnzm.html.

20. Анна Л. Що таке адаптивні технології та як ними користуватися [Електронний ресурс] / Ляшенко Анна. – 2017. – Режим доступу до ресурсу: <http://blog.ed-era.com/adaptivni-technologii/>.

ДОДАТКИ

Лістинг файлу MainFragment

```

package com.education.app.ui.main

import android.content.Context
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 android.widget.Toast
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxHeight
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Alignment.Companion.BottomCenter
import androidx.compose.ui.Alignment.Companion.CenterHorizontally
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.ComposeView
import androidx.compose.ui.platform.ViewCompositionStrategy
import androidx.compose.ui.res.colorResource
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat.startActivity
import androidx.fragment.app.Fragment
import androidx.navigation.NavController
import androidx.navigation.findNavController
import com.education.app.R
import com.education.app.ui.theme.EducationAppTheme

class MainFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        return ComposeView(requireContext()).apply {
            setViewCompositionStrategy(ViewCompositionStrategy.DisposeOnViewTreeLifecycleDes

```

```

troyed)
    setContent {
        EducationAppTheme {
            Box(modifier = Modifier.fillMaxSize(), contentAlignment =
Alignment.Center) {
                Column(
                    modifier = Modifier
                        .fillMaxHeight(),
                    horizontalAlignment = CenterHorizontally
                ) {
                    TopBar()
                    MenuItems(findNavController(), context)
                }
            }
        }
    }
}

private fun navigateToScoreBoard(navController: NavController) {
    navController.navigate(MainFragmentDirections.actionMainFragmentToScoreBoardFrag
ment())
}

private fun navigateToQuizes(navController: NavController) {
    navController.navigate(MainFragmentDirections.actionMainFragmentToWelcomeFragmen
t())
}

private fun navigateToSchedule(navController: NavController) {
    navController.navigate(MainFragmentDirections.actionMainFragmentToScheduleFragme
nt())
}

private fun navigateToVideos(navController: NavController) {
    navController.navigate(MainFragmentDirections.actionMainFragmentToYoutubeVideosF
ragment())
}

@Composable
fun ButtonImage(action: () -> Unit, resource: Int, text: String) {
    Column(
        modifier = Modifier
            .size(200.dp)
            .padding(end = 16.dp),
        horizontalAlignment = CenterHorizontally
    ) {
        Image(
            modifier = Modifier.clickable {
                action()
            },
            painter = painterResource(id = resource),
            contentDescription = null
        )
        Text(text = text, fontSize = 18.sp, textAlign = TextAlign.Center)
    }
}

```

```

@Composable
fun MenuItems(navController: NavController, context: Context) {
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .verticalScroll(rememberScrollState()),
        horizontalAlignment = CenterHorizontally
    ) {
        Row(
            modifier = Modifier.padding(top = 40.dp),
            horizontalArrangement = Arrangement.spacedBy(60.dp)
        ) {
            ButtonImage(
                action = { navigateToQuizzes(navController) },
                resource = R.drawable.quiz,
                text = "Daily Quizzes"
            )
            ButtonImage(
                action = { navigateToSchedule(navController) },
                resource = R.drawable.schedule,
                text = "Check your schedule"
            )
        }
        Row(
            modifier = Modifier.padding(top = 40.dp),
            horizontalArrangement = Arrangement.spacedBy(60.dp)
        ) {
            ButtonImage(
                action = { navigateToScoreBoard(navController) },
                resource = R.drawable.podium,
                text = "Check the scoreboard"
            )
            ButtonImage(
                action = { openBrowser(context, "https://learn.ztu.edu.ua") },
                resource = R.drawable.book,
                text = "Learn from Lectures"
            )
        }
        Row(
            modifier = Modifier.padding(top = 40.dp),
            horizontalArrangement = Arrangement.spacedBy(60.dp)
        ) {
            ButtonImage(
                action = { navigateToVideos(navController) },
                resource = R.drawable.video,
                text = "Watch useful videos"
            )
            ButtonImage(
                action = { showToast(context, "Work in progress") },
                resource = R.drawable.email,
                text = "Contact your Teacher"
            )
        }
        Row(
            modifier = Modifier.padding(top = 40.dp),
            horizontalArrangement = Arrangement.spacedBy(60.dp)
        ) {
            ButtonImage(
                action = { openBrowser(context,
                    "https://cabinet.ztu.edu.ua/site/index") },
                resource = R.drawable.cabinet,
                text = "Open your cabinet"
            )
        }
    }
}

```

```

        ButtonImage(
            action = { showToast(context, "Work in progress") },
            resource = R.drawable.cabinet_experimental,
            text = "Cabinet (Experimental)"
        )
    }
    Row(
        modifier = Modifier.padding(top = 40.dp),
        horizontalArrangement = Arrangement.spacedBy(60.dp)
    ) {
        ButtonImage(
            action = { showToast(context, "Not yet implemented") },
            resource = R.drawable.achievements,
            text = "Achievements"
        )
        ButtonImage(
            action = { openBrowser(context, "https://news.ztu.edu.ua") },
            resource = R.drawable.news,
            text = "University News"
        )
    }
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .background(Color.LightGray)
            .height(100.dp), contentAlignment = BottomCenter
    ) {
        Text(
            modifier = Modifier.padding(bottom = 20.dp),
            text = "Житомирська Політехніка ©2023",
            fontSize = 14.sp
        )
    }
}

private fun showToast(context: Context, msg: String) {
    Toast.makeText(context, msg, Toast.LENGTH_SHORT).show()
}

private fun openBrowser(context: Context, link: String) {
    val browserIntent = Intent(Intent.ACTION_VIEW, Uri.parse(link))
    startActivity(context, browserIntent, Bundle.EMPTY)
}

@Composable
fun TopBar() {
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .background(color = colorResource(id = R.color.top_bar_blue)),
        horizontalAlignment = CenterHorizontally
    ) {
        Row(modifier = Modifier.padding(top = 14.dp)) {
            Text(
                text = "Welcome to Education app",
                textAlign = TextAlign.Center,
                fontSize = 24.sp,
                color = Color.White,
                fontWeight = FontWeight.Bold
            )
        }
        Row(modifier = Modifier.padding(top = 20.dp)) {

```

```
        Text(  
            modifier = Modifier.padding(bottom = 8.dp),  
            text = "Everything you need in one place",  
            textAlign = TextAlign.Center,  
            fontSize = 14.sp,  
            color = Color.White,  
            fontWeight = FontWeight.Bold  
        )  
    }  
}
```

Лістинг файлу ScheduleFragment

```

package com.education.app.ui.schedule

import android.content.Context
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.webkit.WebView
import android.webkit.WebViewClient
import android.widget.Toast
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.material.CircularProgressIndicator
import androidx.compose.material.Text
import androidx.compose.material.TextField
import androidx.compose.material.TextFieldDefaults
import androidx.compose.material3.DropdownMenuItem
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.ExposedDropdownMenuBox
import androidx.compose.material3.ExposedDropdownMenuDefaults
import androidx.compose.runtime.Composable
import androidx.compose.runtime.MutableState
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.ComposeView
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.platform.ViewCompositionStrategy
import androidx.compose.ui.unit.dp
import androidx.compose.ui.viewinterop.AndroidView
import androidx.fragment.app.Fragment
import androidx.fragment.app.activityViewModels
import androidx.navigation.NavController
import androidx.navigation.findNavController
import com.education.app.ui.theme.EducationAppTheme
import com.education.app.ui.views.TopBar

class ScheduleFragment() : Fragment() {
    private val mViewModel: ScheduleViewModel by activityViewModels()

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        return ComposeView(requireContext()).apply {
            setViewCompositionStrategy(ViewCompositionStrategy.DisposeOnViewTreeLifecycleDestroyed)
            setContent {

```

```

        EducationAppTheme {
            val selectedTextOut = remember
{mutableStateOf("https://rozklad.ztu.edu.ua/schedule/group/EI-23-1")}
            Column {
                TopBar(selectedTextOut, mViewModel.studentsGroups.value,
isDropdown = true, onClick = { backToMainScreen(findNavController()) })
                WebViewScreen(screenContext = context, link =
selectedTextOut.value)
            }
        }
    }
}

private fun backToMainScreen(navController: NavController) {
    navController.navigate(ScheduleFragmentDirections.actionScheduleFragmentToMainFr
agment())
}

@Composable
fun WebViewScreen(screenContext: Context, link: String) {
    Box(contentAlignment = Alignment.Center) {
        CircularProgressIndicator(
            modifier = Modifier.size(100.dp),
            strokeWidth = 4.dp,
            color = Color.Red
        )
        AndroidView(
            factory = { context ->
                WebView(context).apply {
                    settings.javaScriptEnabled = true
                    webViewClient = WebViewClient()
                    settings.loadWithOverviewMode = true
                    settings.useWideViewPort = true
                    settings.setSupportZoom(true)
                }
            },
            update = { webView ->
                try {
                    webView.loadUrl(link)
                } catch (e: Exception) {
                    Toast.makeText(
                        screenContext,
                        "Error loading Url: ${e.message}",
                        Toast.LENGTH_SHORT
                    )
                        .show()
                }
            }
        )
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ExposedDropdownMenuBox(text: MutableState<String>, list: List<String>) {
    val context = LocalContext.current
    var expanded by remember { mutableStateOf(false) }
    var selectedText by remember { mutableStateOf(list[0]) }
}

```

```
Box(
    modifier = Modifier
        .fillMaxWidth()
        .padding(32.dp)
) {
    ExposedDropDownMenuBox(
        expanded = expanded,
        onExpandedChange = {
            expanded = !expanded
        }
    ) {
        TextField(
            value = selectedText,
            onValueChange = {},
            readOnly = true,
            trailingIcon = {
ExposedDropDownMenuDefaults.TrailingIcon(expanded = expanded) },
            modifier = Modifier.menuAnchor(),
            colors = TextFieldDefaults.textFieldColors(textColor =
Color.White)
        )

        ExposedDropDownMenu(
            expanded = expanded,
            onDismissRequest = { expanded = false }
        ) {
            list.forEach { item ->
                DropdownMenuItem(
                    text = { Text(text = item) },
                    onClick = {
                        selectedText = item
                        text.value =
"https://rozklad.ztu.edu.ua/schedule/group/$item"
                        expanded = false
                        Toast.makeText(context, item,
Toast.LENGTH_SHORT).show()
                    }
                )
            }
        }
    }
}
}
```

Лістинг файлу YoutubeVideosFragment

```

package com.education.app.ui.videos

import android.content.ActivityNotFoundException
import android.content.Context
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.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement.Absolute.Center
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Alignment.Companion.CenterHorizontally
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.ComposeView
import androidx.compose.ui.platform.ViewCompositionStrategy
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat.startActivity
import androidx.fragment.app.Fragment
import androidx.navigation.NavController
import androidx.navigation.findNavController
import com.education.app.ui.theme.EducationAppTheme
import com.education.app.ui.views.TopBar

class YoutubeVideosFragment : Fragment() {

    private val youtubeVideoList = listOf<String>("L1jhYyAY8GE", "SdiGHYQzd2o",
"DhCdBtjdvaM")
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        return ComposeView(requireContext()).apply {

setViewCompositionStrategy(ViewCompositionStrategy.DisposeOnViewTreeLifecycleDestroyed)

            setContent {
                EducationAppTheme {
                    Column {
                        TopBar(isDropdown = false, onClick = {

```

```

        backToMainScreen(findNavController())
    })
    Row(modifier = Modifier.fillMaxWidth(),
horizontalArrangement = Center) {
        Text("Today useful videos", fontSize = 24.sp)
    }
    LazyColumn(modifier = Modifier.padding(top = 16.dp),
horizontalAlignment = CenterHorizontally, content = {
        items(youtubeVideoList) {
            YoutubePlaceholder(link = it, context = context)
        }
    })
    }
    }
    }
    }
}

@Composable
fun YoutubePlaceholder(link: String, context: Context) {
    Column(modifier = Modifier.clickable {
        openYoutubeVideo(context, link)
    }.fillMaxWidth(), horizontalAlignment = CenterHorizontally) {
        Box(
            modifier = Modifier
                .size(width = 160.dp, height = 100.dp)
                .background(Color.Black)
                .clip(RoundedCornerShape(16.dp))
        )
        Text(modifier = Modifier.padding(12.dp), text = "InApp Video
unavailable, press to open")
    }
}

private fun openYoutubeVideo(context: Context, link: String) {
    val appIntent = Intent(Intent.ACTION_VIEW, Uri.parse("vnd.youtube:$link"))
    val webIntent = Intent(
        Intent.ACTION_VIEW,
        Uri.parse("http://www.youtube.com/watch?v=$link")
    )
    try {
        startActivity(context, appIntent, Bundle.EMPTY)
    } catch (ex: ActivityNotFoundException) {
        startActivity(context, webIntent, Bundle.EMPTY)
    }
}

private fun backToMainScreen(navController: NavController) {
    navController.navigate(YoutubeVideosFragmentDirections.actionYoutubeVideosFragme
ntToMainFragment())
}

```