

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

факультет інформаційних технологій

(факультет)

Кафедра інформаційних систем та технологій та комп'ютерної інженерії

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра

(бакалавра, спеціаліста, магістра)

Студента Ярмака Романа Юрійовича

(ПІБ)

академічної групи 126-17-1

(шифр)

спеціальності 126 «Інформаційні системи та технології»

(код і назва спеціальності)

за освітньо-професійною програмою

«Інформаційні системи та технології»

(офіційна назва)

на тему «Розробка мобільного додатку для вивчення мов»

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	професор Нікулін С.Л.			
розділів:	професор Нікулін С.Л.			

Рецензент	доц. Коротенко Л.М.			
-----------	---------------------	--	--	--

Нормоконтролер	доц. Коротенко Г.М.			
----------------	---------------------	--	--	--

Дніпро
2021

ЗАТВЕРДЖЕНО:

завідувач кафедри

інформаційних технологій
та комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« _____ » _____ 2021_ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр
(бакалавра, спеціаліста, магістра)

студенту Ярмаку Р.Ю. академічної групи 126-17-1
(прізвище та ініціали) (шифр)

спеціальності 126 « Інформаційні системи та технології »

за освітньою-професійною програмою _____

« Інформаційні системи та технології »

на тему Розробка мобільного додатку для вивчення мов”

затверджену наказом ректора НТУ «Дніпровська політехніка» від 07.06.2021р. №317-с

Розділ	Зміст	Термін виконання
Розділ 1. Аналіз стану області рішення задач	На основі матеріалів з відповідних джерел та даних проаналізувати стан області задачі	18.03.2021 р.
Розділ 2. Проектна частина	На основі матеріалів виробничої практики і інших науково-технічних джерел проаналізувати методи і засоби по створенню сайтів та створити відповідну розробку	23.06.2021 р.

Завдання видано _____
(підпис керівника) (прізвище, ініціали)

Дата видачі _____

Дата подання до екзаменаційної комісії 23.06.2021 р.

Прийнято до виконання _____ Ярмак Р.Ю.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 73 сторінки, 20 рисунків, 1 таблицю, 10 джерел, 6 додатків.

Об'єкт розробки: мобільний додаток для вивчення мов.

Мета дипломної роботи: розробка мобільного додатку для вивчення мов.

У вступі вказана мета дипломної роботи, обґрунтована її актуальність, конкретизовані завдання, які необхідно вирішити.

У першому розділі пояснювальної записки наведено опис та аналіз дистанційного навчання на сучасному етапі. Вказані його мета, завдання, негативні та позитивні ознаки. Також наведенні приклади визначень ДН науковців з США, Європи та інших провідних країн.

У другому розділі показана структура та інтерфейс додатку.

У третьому розділі описана реалізація додатку мовою Java.

Практична значимість даного проекту полягає в вивченні іноземних мов та покращенні умов дистанційного навчання для студентів.

Abstract

The explanatory note contains 73 pages, 20 Figures, 1 table, 10 sources, and 6 appendices.

Object of development: a mobile application for learning languages.

The purpose of the thesis: development of a mobile application for learning languages.

The introduction indicates the purpose of the thesis, justifies its relevance, and specifies the tasks that need to be solved.

The first section of the explanatory note describes and analyzes distance learning at the present stage. Its purpose, tasks, and negative signs are indicated. There are also examples of definitions of DN by scientists from the USA, Europe and other leading countries.

The second section shows the structure and interface of the application.

The third section describes the implementation of the application in Java.

The practical significance of this project is to learn foreign languages and improve the conditions of distance learning for students.

Оглавление

Вступ	6
РОЗДІЛ 1	8
АНАЛІЗ ДИСТАНЦІЙНОГО НАВЧАННЯ НА СУЧАСНОМУ ЕТАПІ	8
1. Школи дистанційного навчання	10
1.2 Мета та завдання ДН.....	10
РОЗДІЛ 2	15
ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ МОВ	15
2.1 Опис структури розробки мобільного додатку	15
2.2 Середовище Android Studio.....	16
2.3 Навігація андроїд проекту в Android Studio	18
2.4 Переваги автоматизації проектів	20
2.5 Планова збірка.....	22
2.6 Система збірки Gradle	23
2.7 Контрольні вікна додатку.....	24
2.8 Розробка інтерфейсу користувача.....	25
2.9 Ілюстрація віджетів мобільного додатку.....	27
РОЗДІЛ 3	34
ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ВИВЧЕННЯ МОВ	34
3.1 Опис Dnipro Polytechnic English.....	34
3.2 Приклад коду одного із xml файлів	38
3.3 Реалізація функцій додатку на мові програмування Java	39
3.4 Тестування програмного забезпечення.....	43
ВИСНОВОК.....	46
ПЕРЕЛІК ДЖЕРЕЛ	47
ДОДАТОК А ВІДОМОСТІ МАТЕРІАЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ	49
ДОДАТОК Б Перелік програмних файлів.....	50
ДОДАТОК Б.2 Файл XML розмітки	66
ДОДАТОК В ВІДГУК КЕРІВНИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	Ошибка! Закладка не определена.
Додаток Г РЕЦЕНЗІЯ	Ошибка! Закладка не определена.
Додаток Д СПРАВКА ПЕРЕВІРКИ	74

Вступ

Актуальність теми.

Однією з задач сучасного ВНЗ є надання якісної освіти. Але у зв'язку з теперішніми подіями, а саме поширенням коронавірусної інфекції, яка забрала життя багатьох людей та стала головною загрозою для надання якісної освіти студентам. Залишаючись вдома, на карантині, багатьом студентам важко проходити навчання дистанційно. Будучи студентами, які навчаються на факультеті інформаційних технологій, ми, як ніхто, розуміємо наскільки є важливим і актуальним вивчення іноземних мов. Англійська мова мабуть у 80-90% випадків буде просто необхідна вже працюючому програмісту. В ІТ усе побудоване на міжнародних зв'язках та клієнтах, а англійська — мова міжнародного спілкування. У іноземну компанію без англійської влаштуватися не вдасться, та й в українських зі знанням цієї мови у вас набагато більше можливостей. Тому навчання англійській мові є вкрай важливим для починаючого ІТ-фахівця. Та що ще важливіше — його практика у реальних ситуаціях, які можуть відбутися у роботі. Варто відмітити, що багато хто приходить у сферу ІТ, вже володіючи англійською на хорошому рівні розуміння, але не готові використовувати його у реальних робочих ситуаціях. Технічна документація, завдання та опис проекту — усьому цьому ви зможете відповідати, тільки якщо зможете розібратися у їх тонкощах. Щоб глибоко розуміти сутність робочої документації, вам знов знадобиться англійська.

Саме тому актуальним є впровадження Мобільного додатку для вивчення іноземних мов, який дозволить вивчати саме ті терміни і вирази, котрі потрібні студенту в його спеціальності. Додаток – це інструмент, завдяки якому не тільки значно полегшиться робота викладачів, а й у студентів з'явиться мотивація на самореалізацію в цьому плані.

Теоретичне та методичне вирішення питань, пов'язаних з упровадженням дистанційного навчання в закладах освіти, не є новим завданням для науковців та практиків. Адже про створення необхідних умов та моделі забезпечення громадян і суспільства своєчасною, достовірною та повною інформацією шляхом широкого використання інформаційних технологій, зокрема й у сфері освіти і науки, було заявлено в Національній програмі інформатизації ще в 1998 році.

Мета і задачі дослідження. Метою дослідження є вирішення питання вивчення іноземних мов та покращення умов дистанційного навчання для студентів. Для досягнення цієї мети необхідно розробити програмний продукт Android для вивчення іноземних мов, у якому студенти зможуть вивчати слова на англійській мові, які необхідні їм на обраній спеціальності.

Об'єкт дослідження – процес розробки Мобільного додатку вивчення іноземних мов.

Предмет дослідження – методи і засоби розробки Мобільного додатку вивчення мов.

Мобільний додаток реалізує наступні функції:

- авторизація за допомогою Google-аккаунту;
- самостійне заповнення словника користувачем;
- вивчення слів за допомогою тренувань з карточками;
- лічильник вивчених та доданих слів користувачем;
- ручний та зрозумілий графічний інтерфейс для комфортного користування.

РОЗДІЛ 1

АНАЛІЗ ДИСТАНЦІЙНОГО НАВЧАННЯ НА СУЧАСНОМУ ЕТАПІ

Аналіз наукової літератури дав підстави стверджувати, що нині в педагогічній науці не існує єдиного підходу до визначення сутності поняття «дистанційне навчання». Зауважимо, що в багатьох випадках термін «дистанційне навчання» використовується поряд або замість терміна «дистанційна освіта» або «дистантне навчання». Так, О. Муковіз акцентує увагу на тому, що термін «дистанційне навчання» часто використовують як заміник поняття «дистанційна освіта». Різниця між цими термінами полягає в тому, що надання освіти контролюють освітні установи й викладачі, тоді як за навчання відповідає студент, тобто дистанційне навчання є результатом дистанційної освіти.

У контексті вищевикладеного доречно проаналізувати матеріали Асоціації дистанційного навчання США (The United States Distance Learning Association – USDLA), які визначають «дистанційне навчання» (Distance Learning) як вид навчання, який використовується для визначення фізичного віддалення викладачів та осіб, які отримують освіту під час навчального процесу, що у свою чергу призводить до збільшення відповідальності студентів за своє навчання, яке відбувається на відстані.

Європейська комісія визначає дистанційне (електронне) навчання (англ. distance learning) як «використання нових технологій мультимедіа й Інтернет для підвищення якості навчання шляхом поліпшення доступу до ресурсів і сервісів, а також віддаленого обміну знаннями й спільною роботою».

У контексті заявленої проблематики доречно звернути увагу до вітчизняних нормативно-правових актів, пов'язаних із дистанційним навчанням. Так, у «Положенні про дистанційне навчання» наведено визначення терміна «дистанційне навчання» – «індивідуалізований процес передання і засвоєння знань, умінь, навичок і способів пізнавальної діяльності людини, який відбувається за опосередкованої взаємодії віддалених один від одного учасників навчання у спеціалізованому середовищі, створеному на основі сучасних психолого-педагогічних та інформаційно-комунікаційних технологій».

Ряд вітчизняних і зарубіжних науковців дають визначення дистанційного навчання як:

- упорядкована взаємодія викладача та студентів, спрямована на досягнення поставленої мети (І. Підласий);
- специфічний процес пізнання, яким керує викладач (В. Сластьонін);
- освітня технологія, завдяки якій кожна людина отримує можливість вивчати програму будь-якого коледжу або університету на відстані (В. Кухаренко);
- новітня форма отримання освіти, яка базується на принципі самостійного навчання студента (М. Єлашин);
- організація навчального процесу, де викладач розробляє навчальну програму, у якій превалує самостійність студента в навчанні (П. Підкасистий);
- навчання на відстані, коли педагог і той, кого навчають, розділені просторово (П. Рєзько);
- форма організації навчального процесу, за якою її активні учасники (об'єкт і суб'єкт навчання) досягають цілей навчання, здійснюючи навчальну взаємодію принципово і переважно на відстані (В. Биков);
- нова специфічна форма, що використовує своєрідні засоби, методи, способи навчання, взаємодії вчителя та учнів, учнів між собою (Е. Полат).

До визначення поняття «дистанційне навчання» звертаються багато дослідників. Серед них і Г. Савицька, яка у своїх працях надає таке визначення поняття «дистанційного навчання»: спосіб навчання, що базується на використанні широкого спектру традиційних, нових інформаційних і телекомунікаційних технологій і технічних засобів, які надають тому, кого навчають, свободу вибору освітніх дисциплін, що відповідають стандартам, діалогового обміну з викладачем, при цьому процес навчання не залежить від розташування студента в просторі й часі.

О. Жерновникова розглядає дистанційне навчання як «нову форму організації навчального процесу, що з'єднує в собі традиційні та нові інформаційні технології навчання, який ґрунтується на принципі самостійного отримання знань, що припускає в основному телекомунікаційний принцип доставки студентам основного навчального матеріалу та інтерактивну взаємодію студентів і викладачів як у процесі навчання, так і при оцінці отриманих ними в процесі навчання знань і навичок».

1. Школи дистанційного навчання

У цей час в Україні снує три школи дистанційного навчання: л

Львівська, Київська й Харківська. Львівська школа представлена Львівським інститутом менеджменту й розвиває проект UDL System, присвячений бізнес-освіті. Проект працює кілька років і має не один десяток розроблених дистанційних курсів. До цієї школи відноситься МБЕРИФ Бізнес-центр Івано-Франківська. Київська школа представлена Науковим учбово-методичним центром ЮНЕСКО, що одним з перших почав в Україні проводити дистанційні заняття, Українським центром дистанційної освіти (УЦДО) Національного технічного університету України «Київський політехнічний інститут» (НТУУ «КПІ»), що розробив Концепцію дистанційної освіти в Україні й займається розробкою законодавчої бази, і Центром дистанційної освіти (ЦДО) Світового банку при Українській академії державного управління при Президенті України (УАДУ). Харківська школа представлена Національним технічним університетом «Харківський політехнічний інститут» (НТУ «ХПІ»).

Спільними зусиллями науковців Інституту професійно-технічної освіти НАПН України і практиків створено Систему дистанційного навчання на базі платформи Moodle (elearning.org.ua); на базі ДНЗ «Дніпропетровський центр професійно-технічної освіти туристичного сервісу» проводиться дослідно-експериментальна робота за темою «Підготовка кваліфікованих робітників із професій «Квітникар», «Декоратор вітрин», «Флорист» з використанням технології дистанційного навчання» (науковий керівник Базелюк О. В.); систематично відбуваються науково-практичні заходи, орієнтовані на підвищення рівня готовності педагогів до впровадження дистанційного навчання в ЗП(ПТ)О. Безумовно, що функціонування в них організаційної структури дистанційної освіти мають забезпечувати педагоги, беручи активну участь в реалізації організаційно-управлінського, нормативно-правового, навчально-методичного, інформаційно-телекомунікаційного, економічно-фінансового компонентів.

1.2 Мета та завдання ДН

Метою ДН є надання освітніх послуг шляхом застосування у навчанні сучасних інформаційно-комунікаційних технологій за певними освітніми або освітньо-кваліфікаційними рівнями відповідно до державних стандартів

освіти; за програмами підготовки громадян до вступу у навчальні заклади, підготовки іноземців та підвищення кваліфікації працівників.

Завданням ДН є забезпечення громадянам можливості реалізації конституційного права на здобуття освіти та професійної кваліфікації, підвищення кваліфікації незалежно від статі, раси, національності, соціального і майнового стану, роду та характеру занять, світоглядних переконань, належності до партій, ставлення до релігії, віросповідання, стану здоров'я, місця проживання відповідно до їх здібностей.

На думку А. Хуторського, дистанційне навчання не є електронним варіантом традиційного навчання, що адаптує форми занять та паперові засоби в телекомунікаційні засоби навчання, а покликане вирішувати специфічні завдання щодо розвитку творчої складової освіти:

- підвищення активної ролі слухача у власному навчанні в постановці освітньої мети, доборі домінантних напрямів, форм і темпів навчання;
- різке збільшення обсягу доступних освітніх масивів, культурно-історичних досягнень людства, доступ до світових культурних і наукових скарбів з будь-якого пункту, де є телезв'язок;
- отримання можливості спілкування слухача з педагогами, консультування у фахівців високого рівня незалежно від їх територіального розташування;
- збільшення евристичної складової навчального процесу завдяки застосуванню інтерактивних форм занять, мультимедійних навчальних програм;
- більш комфортні, порівняно з традиційними, умови для творчого самовираження слухача, можливість демонстрації ним продуктів власної творчої діяльності для всіх, широкі експертні можливості оцінки творчих досягнень;
- можливість змагання з великою кількістю бажаючих, що мешкають у різних містах і країнах, за допомогою участі в дистанційних проектах, конкурсах, олімпіадах.

Вважаємо доцільним навести позитивні та негативні характеристики дистанційного навчання. Так, до позитивних характеристик дистанційного навчання можна віднести:

- переваги у гнучкості трати часу, економія на транспортних витратах;
- зацікавленість сучасної молоді користуванням сучасними інформаційними засобами в повсякденному житті;
- простота й доступність такого навчання;

- набуття досвіду й додаткової інформації студентом за допомогою ресурсів мережі Інтернет;
- зручність для компаній та організацій у навчанні своїх співробітників без відриву від постійної роботи;
- більш нижчий рівень оплати за дистанційне навчання порівняно із традиційним;
- індивідуальний темп навчання;
- можливість індивідуального та групового навчання студентів;
- вирішення проблеми втрати концентрації, уникнення обмежень у розкладі;
- оновлення навчального матеріалу за необхідністю.

Щодо негативних ознак дистанційного навчання, то нимамаожна вважати:

- вміст може залежати від технологічних характеристик;
- можливість проблеми мовного бар'єру або перекладу;
- перешкоди в часі через часові пояси;
- проблеми з авторським правом у застосуванні навчальних матеріалів;
- висока коштовність обладнання для проведення дистанційного навчання.

Для вузів також існує ряд переваг впровадження дистанційної форми навчання:

- залучення висококласних викладачів незалежно від їхнього місця проживання;
- кількість студентів може бути в багато разів більше, ніж при традиційних видах навчання. Збільшення чисельності студентів, які отримують очну освіту прямо пропорційно зростанню додаткових фінансових витрат й обмежено масштабами самого навчального закладу, у той час як зростання кількості дистанційних курсів для студентів вимагає набагато менше фінансових вкладень і не залежить від розмірів навчальної установи;
- економія засобів при проведенні навчального процесу з використанням сучасних дорогих мультимедійних проекторів, камер й іншого обладнання. Дане обладнання потрібно тільки для підготовки навчальних матеріалів дистанційного курсу;
- значне вивільнення аудиторного фонду;
- економія витрат на експлуатаційних витратах і відповідно комунальних платежах;
- вирівнювання інфраструктури освітнього комплексу по всій території країни й інтеграція у світовий освітній простір.

Незважаючи на значні переваги дистанційної освіти в Україні існує ряд об'єктивних і суб'єктивних проблем, які стримують процес розвитку й впровадження дистанційного навчання у вітчизняних вузах:

- термінологічні – дистанційне навчання це нова форма й система навчання, а не тільки технологія викладання або модифікація заочної освіти. Неоднозначність у термінології приводить до того, що часто передбачається фінансування саме технологічної й технічної частини розвитку дистанційного навчання, а педагогічна складова залишається без належного фінансування;

- юридичні – недостатність нормативної, законодавчої, правової й документальної бази. Фінансові й бухгалтерські служби не знають, як оплачувати працю викладача дистанційного навчання, також не досить розвинений механізм захисту прав інтелектуальної власності;

- технічні – недостатність технічної бази, невелика швидкість Інтернет доступу, дорожнеча програмних засобів;

- методичні – відсутність методик створенім й використання дистанційних курсів у навчальному процесі на основі сучасних педагогічних, інформаційних і комунікаційних технологій;

- комунікаційні – відсутність прямого очного спілкування між студентом і викладачем, що обумовлено новим способом трансляції знань, при якому навчання організується за індивідуальним навчальним планом, а студенти й викладачі здійснюють комунікацію в рамках дистанційної освітньої системи;

- кадрові – впровадження технологій дистанційного навчання вимагають змін у підготовці педагогів, розвитку в них більшою мірою компетенцій, пов'язаних із інформаційно-телекомунікаційними технологіями.

Українські дослідники серед недоліків дистанційного навчання вказують на потребу створення особливих вимог щодо організації навчального процесу, необхідність у персональному комп'ютері та доступі до мережі Інтернет, організацію мотивації слухачів. Дистанційне навчання потребує відсутності прокрастинації; його результат безпосередньо залежить від самостійності та свідомості студента, оскільки відсутній постійний контроль за слухачами. Однією із ключових проблем дистанційного навчання залишається проблема аутентифікації користувача у перевірці знань.

Окрім уже визначених недоліків дистанційного навчання, варто врахувати зауваження сербських дослідників, які вважають, що для розвитку сучасного фахівця необхідними у кваліфікації є комунікативні навички, які, на жаль, не розвиваються в умовах дистанційного навчання такою мірою, як за традиційного.

Дистанційна освіта в Україні регулюється Концепцією розвитку дистанційної освіти й Положенням про дистанційне навчання МОН України. Під дистанційним навчанням розуміється комплекс освітніх послуг, які надаються вилученим від навчального закладу студентам за допомогою спеціалізованого інформаційно-освітнього середовища, що базується на засобах обміну навчальною інформацією за допомогою сучасних телекомунікаційних технологій. Дистанційне навчання можна визначити як інноваційну освітню технологію, в якій провідна роль належить самостійній навчальній діяльності здобувача освіти, шляхом використання телекомунікаційних технологій та мінімізації аудиторної взаємодії між студентом і викладачем у навчальному процесі вищого навчального закладу. Нині впровадження дистанційного навчання в навчально-виховний процес в Україні має недостатню підтримку органів державного управління, місцевого самоврядування та громадських організацій. Це зумовлено браком відповідних законодавчих актів, регулювальних документів і в цілому відсутністю системи управління в освітній галузі.

РОЗДІЛ 2

ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ МОВ

2.1 Опис структури розробки мобільного додатку

Основна мета розробки Мобільного додатку для вивчення мов:

- 1) Підвищення рівня дистанційної освіти НТУ ДП.
- 2) Комфорт та профорієнтованість користувача додатку.

Додаток забезпечує зберігання такої інформації:

- Контент, введений студентом або викладачем
- Google-акаунт користувача
- Посилання на сайт НТУ ДП

Основні функції додатку:

- 1) Додавання слів у додаток користувачем
- 2) Перегляд, редагування, видалення слів, введених користувачем
- 3) Підрахунок вивчених слів
- 4) Зберігання даних користувача за допомогою Google-акаунту

Відмінною рисою даного додатку є те, що він може існувати поза громад інформаційних систем конкретного ВНЗ, незалежно від інших модулів загальної системи. Розробка повноцінного самостійного додатку, спрямованого на вивчення іноземних мов, дозволить використовувати додаток в рамках навчальної кафедри конкретного ВНЗ або використовувати в інших ВНЗ. Основні функції додатку не є специфічно спрямованими на діяльність будь-якої кафедри певного навчального закладу, цей факт робить додаток універсальним.

2.2 Середовище Android Studio

Сьогодні кожна мова програмування має інтегроване середовище розробки. Ми використовуємо Android Studio як середовище розробки. Це інтегроване середовище розробки Java, яке дозволяє розробникам використовувати інструменти для побудови додатків. Середовище можна встановити на різні операційні системи. Android Studio можна безкоштовно завантажити та використовувати. Він містить макет, що використовується для створення користувальницького інтерфейсу, де зазвичай починається робота програми.

Середовище Android Studio містить інструменти для створення додатків для смартфонів, планшетів, годинників та автомобілів.

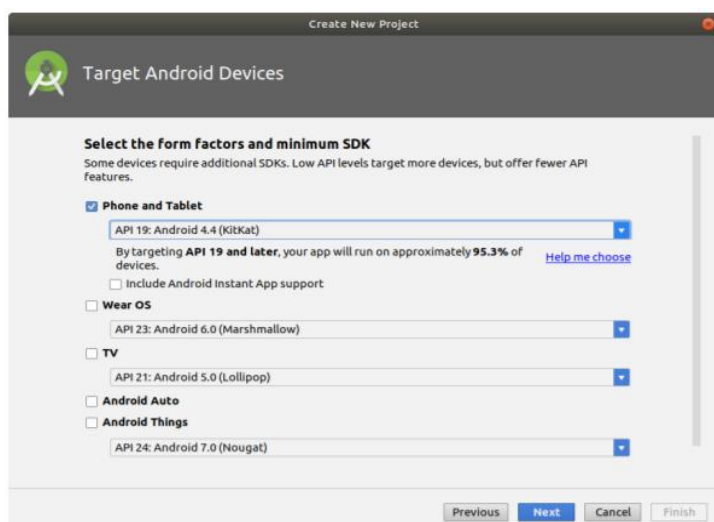


Рисунок 1.1 — Екран вибору форми фактору в Android Studio

Середовище Android Studio є офіційним інтегрованим середовищем розробки для проектування та розробки програм для Android. Він базується на IntelliJ IDEA, інтегрованому середовищі розробки Java, включаючи засоби редагування коду та середовище розробки додатків. Вперше студія Android була оголошена на Google I / O у травні 2013 року, а перша стабільна версія випущена в грудні 2014 року.

Android Studio підходить для настільних платформ, таких як Windows, Mac та Linux. Він замінив Eclipse і став основним інструментом для розробки додатків на Android. Щоб почати користуватися Android Studio, спочатку потрібно встановити безкоштовний пакет JDK (Java Development Kit) для розробників. Після встановлення та налаштування цього пакету вам потрібно завантажити Android Studio з офіційного веб-сайту. Далі потрібно знайти завантажений виконуваний файл інсталяції Android Studio та розпочати процес встановлення. Коли з'явиться майстер встановлення Android Studio, вам потрібно налаштувати установку відповідно до вимог користувача. Розташування файлової системи, де має бути встановлене середовище Android Studio, і розташування файлової системи, до якої можуть отримати доступ інші користувачі системи. Після встановлення інтегрованого середовища розробки можна приступати до створення додатків.

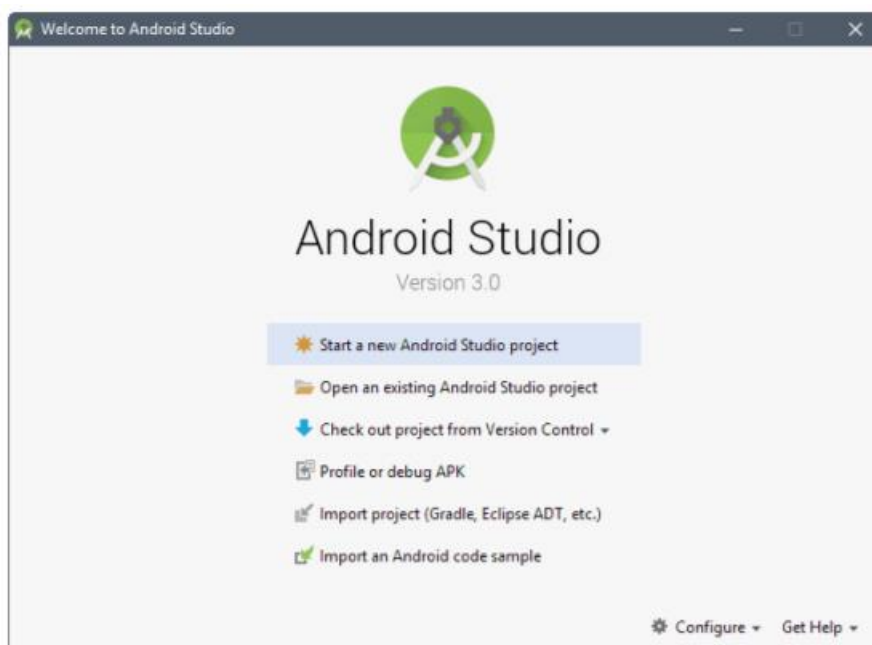


Рисунок 1.2 — Стартовий екран проекту в Android Studio

Інтегроване середовище розробки Android Studio - це вікно. Щоб повною мірою використовувати екранний простір і не перевантажувати розробників, Android Studio відображає лише невелику частину доступних вікон у будь-

який час. Деякі вікна є контекстно-залежними і з'являються лише тоді, коли ви здійснюєте контекстні дзвінки за допомогою певних інструментів. Android Studio також містить приховані вікна, які розробники можуть активувати або деактивувати в меню налаштувань. Однією з найважливіших функцій будь-якого інтегрованого середовищарозробки є навігація.

2.3 Навігація андроїд проекту в Android Studio

Проект Android зазвичай складається з безлічі папок, каталогів та файлів, організованих разом. Програмні додатки з великою кількістю файлів вихідного коду та ресурсів впорядковані у 45 структур проектів. Це робиться для кращої класифікації файлів та визначення компіляції вихідного коду та генерації байт-коду. Взагалі кажучи, ця файлова структура називається проектом, який є організаційною одиницею і повним програмним рішенням. Проекти додатків зазвичай складаються із зростаючих файлів Java або Kotlin, файлів конфігурації XML, зображень, стилів та інших ресурсів в організаційній структурі.

При створенні програми Android Studio допомагає побудувати проект. На етапі складання проекту створюється файл конфігурації і каталог будується в ієрархічній структурі. Результати цього процесу слід повторити для всіх, хто розпочинає збірку проекту. Крім того, переваги включають портативні компоненти. Можливість запуску збірок IDE дуже обмежена. По-перше, розробникам програмного забезпечення потрібно встановлювати на свої пристрої певні продукти. По-друге, IDE можна використовувати лише в певній операційній системі. Автоматизована збірка не вимагає певного середовища виконання, будь то операційна система чи середовище розробки.

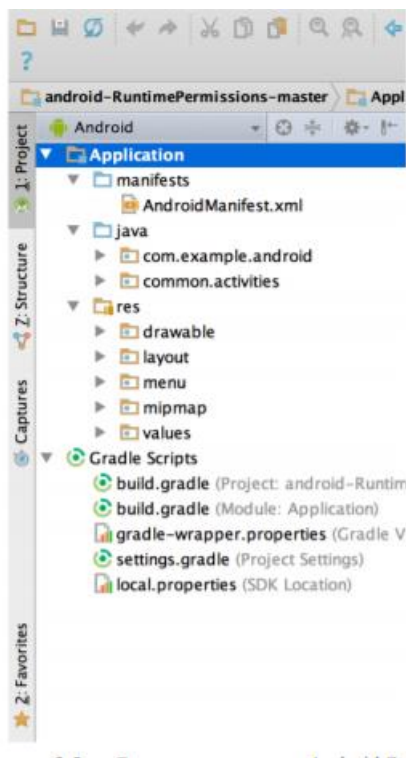


Рисунок 2.1 — Структура проекту в Android Studio

У процесі розробки додатків в інтегрованому середовищі розробки Android Studio додаток потрібно компілювати та запускати багато разів. Розроблений додаток Android можна протестувати, встановити та запустити на фізичному пристрої або віртуальному пристрої Android (AVD-андроїд віртуальний пристрій), який називається емулятором. Перш ніж використовувати віртуальний пристрій, спочатку потрібно завантажити та налаштувати його. Емулятор забезпечує майже всі функції справжнього пристрою Android. Користувачі можуть імітувати вхідні дзвінки та текстові повідомлення, вказувати місце розташування пристрою, імітувати обертання екрана та інші апаратні датчики, отримувати доступ до Google Play тощо. Тестування програм на віртуальних пристроях зручніше та швидше, ніж на фізичних пристроях. Наприклад, ви можете передавати дані на симулятор швидше, ніж підключений фізичний пристрій.



Рисунок 2.2 — Віртуальний девайс в Android Studio

2.4 Переваги автоматизації проектів

Автоматизація проектів має кілька очевидних переваг у процесі розробки програмного забезпечення. Один - запобігання ручному втручанню в процес складання. Потреба в ручному виконанні операцій зі створення та забезпечення вимог до програмного забезпечення займає багато часу та збільшує ризик помилок. Розробники та системні адміністратори можуть вирішувати багато різних завдань при компіляції вручну. Будь-який крок у процесі розробки, який можна автоматизувати, повинен бути автоматизованим.

Процес створення програмних продуктів для розробників програмного забезпечення та тестувальників без автоматизованих інструментів є дуже повторюваним, нудним і значно збільшує ризик помилок. Кожен крок у процесі розробки програмного забезпечення, починаючи від вихідного коду скомпільованого коду і закінчуючи збіркою програмного забезпечення для випуску та виробничого тестування, потрібно робити вручну. Автоматизація проекту допомагає зменшити обсяг ручного втручання в проект, підвищити ефективність процесу розробки та мінімізувати можливість відмови програмного забезпечення.

Ще однією перевагою є створення дублікатів збірок. Звичайна побудова програмного проекту відповідає заздалегідь визначеним і упорядкованим крокам. Наприклад, розробник повинен спочатку скопіювати вихідний код, потім запустити тест і, нарешті, скопіювати проект, щоб отримати конкретний результат. Ви повинні щодня виконувати одні і ті ж дії знову і знову. Цей процес повинен бути таким простим, як натискання кнопки.

У найкращому випадку автоматизовані завдання слід виконувати з командного рядка, щоб ви могли в будь-який час розпочати процес складання з будь-якого пристрою. Автоматизація проекту поділяється на кілька основних типів. Перший - збирати на замовлення. Типовий випадок використання автоматизації на вимогу - це коли користувачі починають використовувати певний пристрій. Як правило, система контролю версій VSC управляє файлами перевірки збірок та висхідних кодів. У більшості випадків користувачі виконують сценарії в командному рядку, які виконують завдання у визначеній послідовності.

Наприклад, компіляція вихідного коду, копіювання файлів з одного каталогу в інший або компіляція результатів. Цей тип автоматизації зазвичай виконується кілька разів на день. Наступним типом автоматизації є запуск збірки. Якщо розробник займається гнучкою розробкою програмного забезпечення, він зацікавлений в отриманні швидких відгуків про стан проекту. Не менш важливою є інформація про те, чи можна компілювати вихідний код без помилок, а також інформація про потенційні дефекти програмного забезпечення в проекті, помічені під час блокування помилок або інтеграційних тестів. Цей тип автоматизації зазвичай є результатом перевірки коду системи контролю версій. Іншим видом автоматизації є планова збірка.

2.5 Планова збірка

Ідея полягає в тому, щоб планувати автоматизацію як часовий план завдань. Він виконується через певний інтервал часу або в певний час. Автоматизація розкладу зазвичай виконується на виділеному сервері. Цей тип автоматизації особливо корисний для створення звітів або документів для проектів. Практика впровадження планування та ведення зборів зазвичай визначається як безперервна інтеграція КІ (рисунок 3.1).

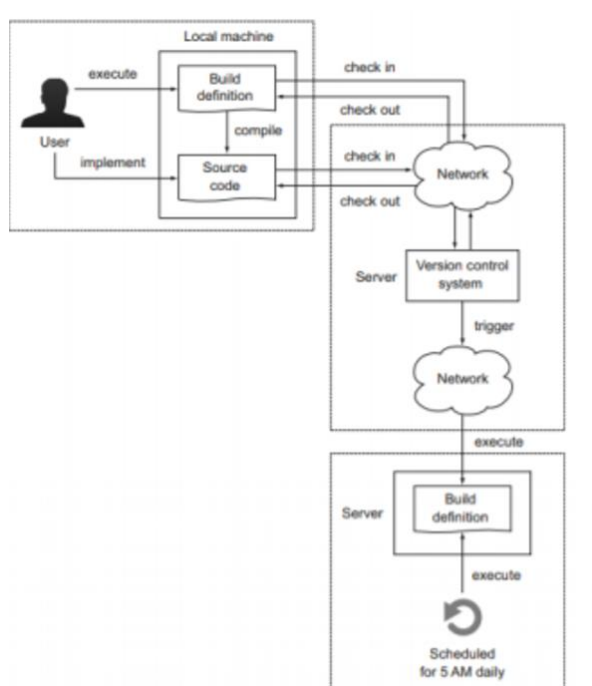


Рисунок 3.1 — Приклад запланованої збірки

Протягом багатьох років вимоги до складання програмного забезпечення для збірки та упаковки були простими. Однак ландшафт сучасної розробки програмного забезпечення змінився, що вимагало автоматизованої побудови проекту. Сьогодні проект включає великий і різноманітний стек програмного забезпечення, що включає велику кількість мов програмування, і використовує широкий спектр стратегій тестування. Зі збільшенням гнучкості збірки повинні підтримувати інтеграцію коду якомога швидше, а також швидко і легко поставляти продукти в тестові та виробничі середовища.

Встановлені будівельні інструменти зазвичай не досягають цих цілей простим способом.

2.6 Система збірки Gradle

Автоматизована система збірки Gradle була вперше випущена в 2007 році. Gradle - це наступний еволюційний крок інструменту побудови JVM. Система спирається на досвід вже існуючих автоматизованих інструментів збірки (таких як Ant і Maven) і виводить їхні найкращі ідеї на новий рівень. Дотримуючись методу побудови, Gradle дозволяє вам декларативно моделювати предметну область, використовуючи потужну та виразну доменну мову DSL, реалізовану в Groovy замість XML. Оскільки Gradle є рідною для JVM, це дозволяє писати власну логіку в Groovy або Java.

У Java існує надзвичайна кількість бібліотек та фреймворків. Управління залежностями використовується для автоматичного завантаження цих артефактів із сховища та надання їх програмному коду. Після розуміння недоліків існуючих рішень щодо управління залежностями, Gradle забезпечує власну реалізацію. Він не тільки ретельно налаштований, але й прагне бути максимально сумісним із існуючими інфраструктурами управління залежностями, такими як Maven та Ivy. Gradle забезпечує потужну підтримку для визначення та організації багатьох збірок проєктів та моделювання взаємозв'язків між проєктами. Для розробників автоматизація проєктів є частиною їх щоденної роботи. Сцена побудови Gradle є декларативною, читабельною та зрозумілою. Використання Groovy з філософськими елементами Build-by Convention замість XML для написання коду дозволяє значно зменшити розмір сценарію збірки та зробити його більш читабельним, як показано на малюнку

Кожна збірка Gradle починається зі сценарію. За замовчуванням ім'я сценарію Gradle - build.gradle. Коли ви запускаєте базову команду в оболонці, система збірки шукатиме файл із таким іменем. Якщо його не знайти,

відображається повідомлення про помилку. Після створення цього файлу потрібно визначити одне або кілька завдань.

Підводячи підсумок, ми можемо визначити, що Gradle - це автоматична система складання, яка походить від декларативного та виразного Groovy DSL. Він поєднує в собі гнучкість та легкість масштабування, а також конфігурацію та підтримку традиційних ідей управління залежностями. Gradle стає першим вибором для багатьох проектів з відкритим кодом. Висновки цього розділу У цьому розділі проаналізуйте та огляньте програмні технології та інструменти, що використовуються в процесі створення програмних продуктів, та обґрунтовано виберіть інструменти, що використовуються при розробці програмного забезпечення.

2.7 Контрольні вікна додатку

Система додатку буде складатися з декількох контрольних вікон. Головне - це базове вікно для вибору аккаунту. Також буде вікно для самого курсу, наповнене вмістом самим студентом. Система також матиме кілька допоміжних вікон для взаємодії між програмними продуктами та користувачами. Завдяки чіткій та зрозумілій організації модулів, обрана структура буде простою та інтуїтивно зрозумілою для користувача. На рисунку 3.2 показана схема структурної системи, де розташовані всі програмні модулі.

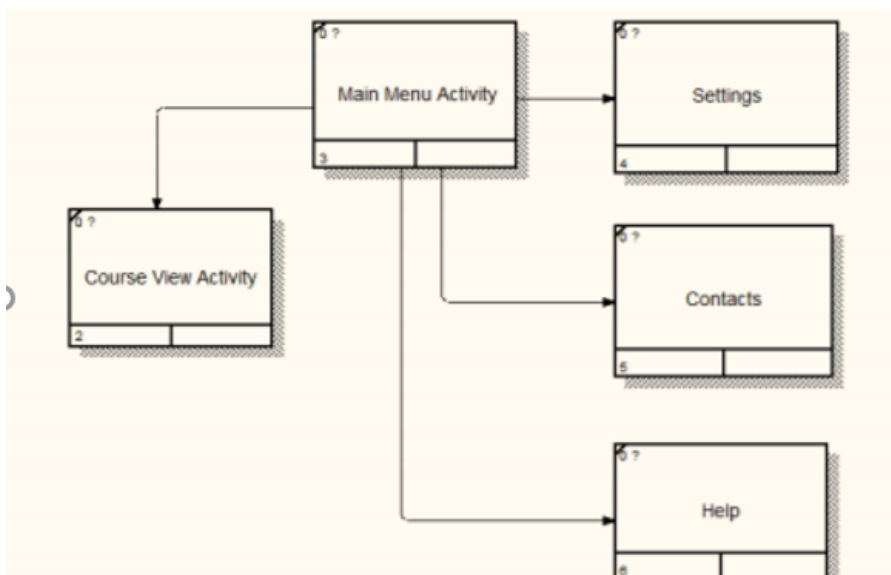


Рисунок 3.2 — Приклад запланованої збірки

2.8 Розробка інтерфейсу користувача

Графічний інтерфейс користувача повинен відповідати таким вимогам: чіткість, актуальність, виразність та послідовність. Для задоволення цих вимог розроблені інтерфейси програмних продуктів. Він включає кілька модулів. Це головне меню, сторінка курсу, налаштування, посилання на сайт “НТУ ДП”. Коли користувач закриє програму, його успішність буде збережена. Вікно налаштування довідки дозволяє користувачам персоналізувати програму для зручності використання. Вікно контактів надає користувачам інформацію про те, як спілкуватися з розробниками. Вікно каталогу призначене для надання інформації та інструкцій щодо використання системи.

Вікно головного меню. Це вікно використовується для відображення всіх кнопок і дозволяє користувачеві вибрати відповідний варіант. Воно складається з кількох елементів, а саме: - назва вибраного варіанту на верхній панелі; - список усіх слів програми; -тестування; -редактор слів; - персональна статистика . Вікно курсу використовується для відображення інформації про курс та вмісту курсу на екрані

Вікно налаштувань. Це вікно використовується для зміни та персоналізації слів. Воно складається з декількох елементів, а саме: -дизайну змін.

Цей розділ уточнює роль розроблених програмно-апаратних агентів у системі дистанційного навчання та визначає методи розробки агентів у подальшій розробці. Отримано структуру системи розробленої системи, опис основних змінних та функцій агента та схему придбання апаратної частини. Програма призначена для мобільних пристроїв (смартфонів та планшетів), що працюють під управлінням операційної системи "Android", використовуючи Інтернет для підключення та взаємодії з інтерактивними елементами. Додаток встановлюється на пристрій користувача шляхом встановлення файлу "apk" та його автоматичної конфігурації.

Оскільки додаток розроблено на платформі "Android", пристрій користувача повинен встановити цю операційну систему на своєму пристрої. Щоб програма працювала належним чином, версія операційної системи повинна бути принаймні 4.0.0. Пристрій користувача також повинен мати доступ до Інтернету.

23:06 M m 🔒 ·



Dnipro University of Technology

**The app is designed specifically
for the university**

[University website](#)

[Go next >](#)

Інструкції щодо використання програмних продуктів. Під час входу, відкривається головне меню, рисунок 4.1



Рисунок 4.1 — Головне меню

Користувач з'являється в у вікні авторизації, студент повинен обрати свій Google-акаунт входу у програму. Акаунти відображені у виді списку.

Вікно авторизації відображено на рисунку 4.2

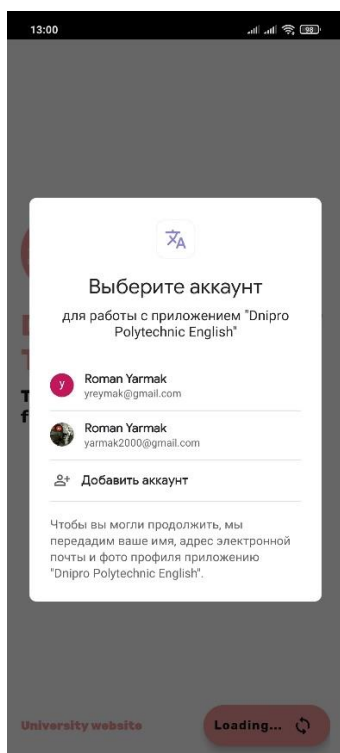


Рисунок 4.2 — Вікно авторизації

У головному меню можна знайти кнопку, для переходу на сайт “НТУ ДП”. Відображена на рисунку 4.3.



Рисунок 4.3 — перехід на сайт “НТУ ДП”.

Пройшовши авторизацію, користувач потрапляє до віджету, у якому є можливість перелистування карточок зі словами та поясненням до них (рисунок 4.4). Цю дію можна реалізувати за допомогою кнопки (рисунок 4.5).



Рисунок 4.4 — віджет з карточками слів



Рисунок 4.5 — кнопка для перелистування слів

Наступним віджетом є тести для тренування доданих термінів (рисунок 4.6).

Якщо студент вводить правильний переклад слова, програма викликає знизу вікно з надписом англійською мовою “Success” (рисунок 4.7). В інакшому випадку під текстовим полем для вводу перекладу, з’являється надпис “Bad Translation” (рисунок 4.8).

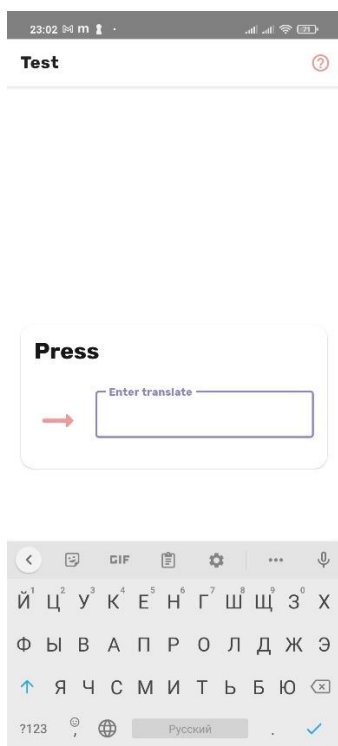


Рисунок 4.6 — тренування доданих слів.

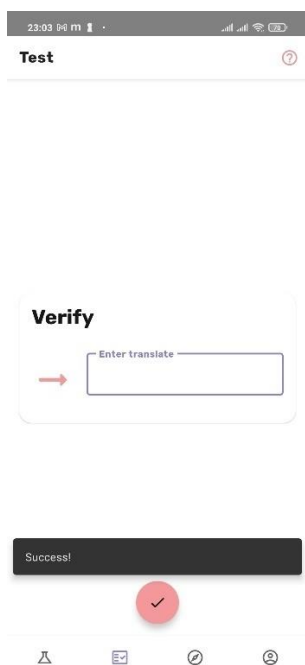


Рисунок 4.7 — відображення правильного вводу.



Рисунок 4.8 — відображення хибної відповіді

Пройшовши усі вищеперечисленні кроки, користувач переходить до віджету зі списком доданих слів, які необхідно вивчити (рисунок 4.9). Це можна зробити за допомогою спеціальної кнопки (рисунок 4.10).



Рисунок 4.9 — віджет зі списком слів



Рисунок 4.10 — кнопка для додавання нових термінів

Останній віджет — статистика користувача, яка зберігається у данному додатку. На рисунку 4.11 відображений лічильник, вгаданих студентом, слів.

Також внизу розташована кнопка виходу з акаунту, яка направляє користувача на головну сторінку додатку та автоматично анулює статистику вивчених термінів (рисунок 4.12).



Рисунок 4.11 — відображення статистики акаунту



Рисунок 4.12 — кнопка виходу з акаунту

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ВИВЧЕННЯ МОВ

У попередньому розділі було обрано технологію розробки мобільних додатків, обрано дизайн, обрано кольорову схему та розроблено базу даних та модулі. Наступним кроком є програмна реалізація мобільного додатку для вивчення мови. Розробка модуля авторизації та реалізація програмного забезпечення для захисту.

Аутифікація - це автентифікація ідентифікатора користувача. Одним із методів перевірки особи в інформаційних системах є попередня ідентифікація на основі ідентифікатора користувача, імені для входу та пароля. Після того, як комп'ютер отримує ім'я користувача та пароль, введені користувачем, він порівнює їх зі значеннями, що зберігаються в спеціальній базі даних безпеки, і якщо аутифікація успішна, авторизація виконується, і тоді користувачеві дозволяється працювати в системі.

3.1 Опис Dnipro Polytechnic English

I. Бібліотека

Використані бібліотеки в додатку:

1. androidx.appcompat:appcompat:1.3.0

Графічні елементи програми

2. com.google.android.material:material:1.3.0

Графічні елементи програми згідно material.io

3. androidx.room:room-runtime:2.3.0

Бібліотека для роботи з базою даних програми. Room-це обгортка над SQLite

4. androidx.lifecycle:lifecycle-*

Lifecycle бібліотеки для створення архітектури додатки-MVVM, для написання простого і красивого коду

5. com.google.code.gson:gson:2.8.7

Перетворення JSON в POJO і назад

6. androidx.navigation:navigation-*

Бібліотека для навігації в додатку.

II. /adapter

1. RecyclerViewListAdapter

Адаптер для RecyclerView для відображення списку слів у додатку (рисунок 5.1)

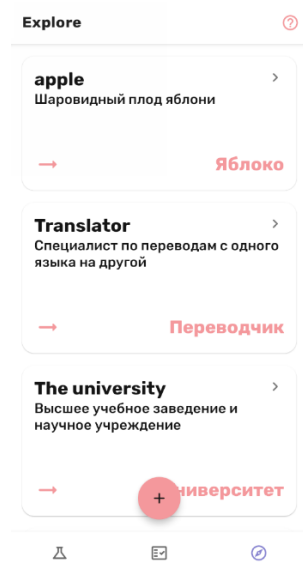


Рисунок 5.1 — відображення списку слів у додатку

III. /model

1. Response

POJO-модель щоб парситидані з JSON

2. Word Model

Сама POJO-модель, в якій зберігається модель слова: слово, переклад, опис

IV. /repository

4.1. Repository

Singleton-клас, який живе весь час, коли додаток запущено. За допомогою нього створюємо базу даних і отримуємо до неї доступ:

```
Room.databaseBuilder(this, AppDatabase.class, Const.DB_NAME)
    .allowMainThreadQueries()
    .build();
```

4.2. App Data base

Клас бази даних

4.3. DaoDb

Dao-клас, в якому описуються всі запити до бази даних, наприклад:

```
@Query("SELECT * FROM word model ORDER BY RANDOM() LIMIT
1")
Word Model getByRandom();
);
```

V. /ui

/explore

/learn

/test

Тут механізм однаковий: є фрагмент і view model до нього.

Фрагмент відповідає за UI, а view model відповідає за всі розрахунки.

При зміні якихось даних в view model, вони відразу відображаються в UI, тобто, вони працюють в зв'язці.

Це відбувається завдяки бібліотекам lifecycle і архітектури MVVM, за рахунок цього код стає набагато чистіше, легше, швидше і простіше

VI. /ui/Main Activity

Тут відбувається зв'язка навігації і меню для навігації, тобто:

з res / navigation / nav_graph.XML беруться екрани програми і вони автоматично зв'язуються з меню завдяки

```
private void loadNavController() {
    NavController = ((NavHostFragment)
        getSupportFragmentManager().findFragmentById(binding.navHostFragment.getId(
            )))NavController();
    AppBarConfiguration appBarConfiguration = new
        AppBarConfiguration.Builder(binding.bottomNavigation.getMenu()).build();
    NavigationUI.setupActionBarWithNavController(this, NavController,
        appBarConfiguration);
    NavigationUI.setupWithNavController(binding.bottomNavigation, NavController);
    NavController.navigate(R.id.learnFragment);
}
```

VII. Const

Файл констант, сайту університету, назви БД, JSON_NAME – цей файл зберігається в assets / words.json, при першому запуску програми звідси беруться слова і ними заповнюється база даних

Loading Status, Error Status для View Model щоб розуміти state

VIII. xml

Всі ці файли відповідають за дизайн програми (рис 5.2)

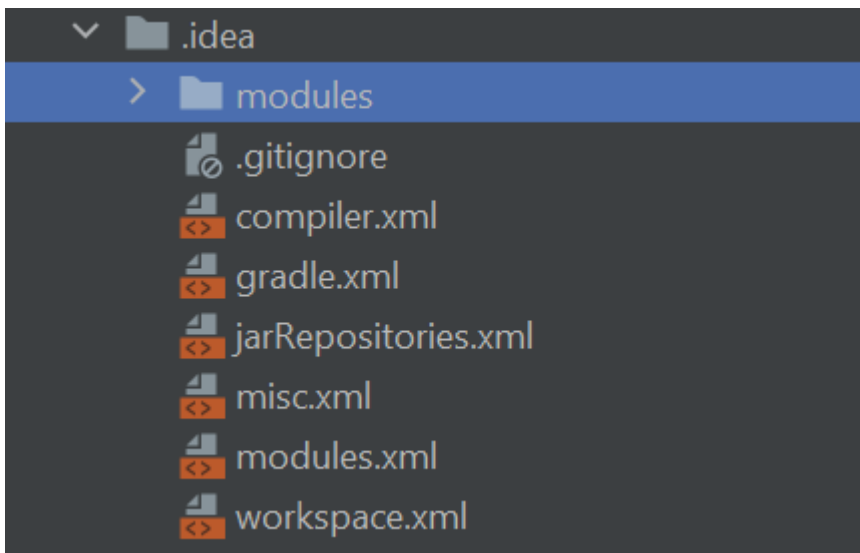


Рисунок 5.2 — відображення xml файлів в IDE

3.2 Приклад коду одного із xml файлів

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
<component name="GradleMigrationSettings" migrationVersion="1" />
<component name="GradleSettings">
<option name="linkedExternalProjectsSettings">
<GradleProjectSettings>
<option name="testRunner" value="PLATFORM" />
<option name="distributionType" value="DEFAULT_WRAPPED" />
<option name="externalProjectPath" value="$PROJECT_DIR$" />
```

```

<option name="modules">
<set>
<option value="$PROJECT_DIR$" />
</set>
</option>
<option name="resolveModulePerSourceSet" value="false" />
</GradleProjectSettings>
</option>
</component>
</project>

```

3.3 Реалізація функцій додатку на мові програмування Java

Завантажуємо інформацію про профіль кожен раз при ON_RESUME

```

@OnLifecycleEvent(Lifecycle.Event.ON_RESUME)
private void loadUser() {
    Glide.with(this)
        .load(FirebaseAuth.getInstance().getCurrentUser().getPhotoUrl())
        .circleCrop()
        .into(binding.imageView2);

    new SharedPrefs(requireContext());
    int i = SharedPrefs.getWins();
    int j = SharedPrefs.getAll();

    binding.textV1.setText(Html.fromHtml("<font color=#F4989C>" + i +
"</font><font color=#171717>/" + j + "</font>"));

```

```
binding.textV.setText(FirebaseAuth.getInstance().getCurrentUser().getDisplayName());
```

```
binding.btnLog.setOnClickListener(v -> {

    FirebaseAuth.getInstance().signOut();
    SharedPreferences.setAll(0);
    SharedPreferences.setWins(0);

    startActivity(new Intent(requireActivity(), SplashActivity.class));
    requireActivity().finish();
});
}
```

Завантажуємо список з бази даних

```
public void loadApp() {

    AppDatabase appDatabase = Repository.database();
    words = appDatabase.dao().getAll();
}
```

Видаляємо слово с бази даних

```
public void deleteWord(WordModel wordModel) {

    Repository.database().dao().delete(wordModel);
    loadApp();
}
```



```
}
```

Клік на айтем зі списку

```
public void onItemClick(WordModel wordModel) {

    new MaterialAlertDialogBuilder(requireContext())
        .setTitle("Delete")
        .setMessage("Are you sure you want to delete «" +
wordModel.getWordOriginal() + " - " + wordModel.getWordTranslated() + "» ?")
        .setPositiveButton("Delete", (dialog, which) ->
viewModel.deleteWord(wordModel))
        .setNegativeButton("Cancel", null)
        .show();

}
```

Перевіряємо завантажили чи слова в базу даних

```
viewModel.getStatus().observe(this, status -> {
    switch (status) {
        case LOADING:
            binding.fabNext.setText("Loading...");
            binding.fabNext.setIconResource(R.drawable.ic_round_loop_24);
            break;
    }
});
```

View model робить все сама, завантажує слова при необхідності за
ДОПОМОГОЮ

ВИПАДКОВЕ ЧИСЛО

Override

```
public View onCreateView(LayoutInflater inflater,
                        ViewGroup container,
                        Bundle savedInstanceState) {

    binding = FragmentLearnBinding.inflate(inflater, container, false);

    viewModel = new
ViewModelProvider(this).get(LearnFragmentViewModel.class);
    viewModel.loadApp();
    binding.setViewModel(viewModel);
    binding.setLifecycleOwner(getViewLifecycleOwner());

    return binding.getRoot();
}
```

Якщо ми не увійшли в аккаунт-необхідна авторизація

case SUCCESS:

```
if (FirebaseAuth.getInstance().getCurrentUser() != null) {
    startActivity(new Intent(this, MainActivity.class));
    finish();
} else {
```

```
    GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
```

```

        .requestIdToken("419557326703-
e5kdkl6p3pd6i6nu6puc9dmggc9h0mrj.apps.googleusercontent.com")
        .requestEmail()
        .build();

    mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

    signIn();

}
break;
}

```

3.4 Тестування програмного забезпечення

Тестування є важливою частиною процесу розробки програмного забезпечення. Тестування проводиться протягом усього процесу розробки програми. Тест модуля перевіряє, чи працює кодовий блок належним чином. В ООП такою одиницею є клас. Отже, модульний тест повинен перевірити, що кожен відкритий метод класу працює належним чином. Іншими словами, за вхідними даними тесту, тест перевірить, чи повертає метод очікуваний результат. Тест функцій перевіряє, чи працюють належним чином такі функції (як керування записами). Порівняно з модульним тестуванням, функціональне тестування знаходиться на вищому рівні, оскільки найчастіше перевіряє роботу декількох класів.

Щоб створити нову програму, буде створено такий каталог, який дозволить вам писати та запускати тести:

```

testdrive/
protected/      файли
tests/          тести

```

fixtures/	фікстури БД
functional/	функціональні тести
unit/	модульні тести
report/	звіт про випробування коду

Як показано вище, тестовий код в основному складається з трьох каталогів: світильники, функціональний та блок. Каталог звітів використовується для зберігання згенерованих звітів про покриття тесту. Під час запуску тесту, що передбачає використання бази даних, необхідно перевизначити базу даних, щоб не знищити фактичні дані або дані, що використовуються при розробці. Для цього розкоментуйте наведену вище конфігурацію db і вкажіть DSN тестової бази даних у властивості connection String.

За допомогою такого вхідного сценарію під час запуску тесту ми отримали приклад програми, максимально наближений до реального додатку. Головна відмінність полягає в тому, що він підтримує світильники та використовує тестову базу даних. Тест конфігурації. Перевірка тесту конфігурації показує, чи повністю відображаються ресурси на різних розширеннях екрану.

Параметр	Наявність параметру
Час, що витрачається при заданій швидкості з'єднання з мережею Інтернет	+
Адекватне відображення за допомогою різних екранів	+
Правильність відображення шрифтів різними	+
Коректності гіперпосилань	+
Відображення всіх графічних матеріалів: малюнків, фотографій та інше	+
Перевірка орфографії та пунктуації текстів	+
Перевірка відповідності наповнення сторінок вихідного контенту	+

Тому дослідження рівня навантаження показує хороші результати для мобільних програм вивчення мови.

ВИСНОВОК

Ми розробили мобільний додаток для вивчення мов. Система управління програмами реалізує функції для кожного модуля, а саме: додавання, редагування та видалення. Додаток також надійно захищений від можливих атак, а саме: недостатня автентифікація та авторизація, а також перевірена конфігурація та навантаження. Всі модулі системи працюють нормально відповідно до завдання.

Розроблений інтерфейс простий у використанні, що може запобігти введенню користувачами неправильних даних та неправильних операцій. Якщо в системі є помилки, відобразиться повідомлення про те, як їх виправити за допомогою красивих кольорних схем. Проаналізували можливість існуючої системи навчання та її функцій на практиці.

Під час стажування були розроблені програмні продукти, які повністю відповідають вимогам. Додаток дозволяє користувачам отримати доступ до термінології, яка викладається у вищих навчальних закладах. Це дозволяє отримати конкретні, профільні знання. Вивчено методи впровадження програмного забезпечення та інструменти продукту. Тому його було обрано для створення додатків для мобільної платформи "Android" на основі парадигми об'єктно-орієнтованого програмування.

Використовуємо платформу "Android" для розробки програм мовою "Java". Це дозволяє збільшити гнучкість та зручність системи під час проектування та використання. Розроблена система перевірена і відповідає всім вимогам програмного забезпечення. Цей додаток призначений для тих, хто бажає здобути якісну освіту вдома. Тому в результаті був розроблений мобільний додаток, що відповідає усім вимогам, і отримані нові знання щодо розробки програмних продуктів для мобільних систем.

ПЕРЕЛІК ДЖЕРЕЛ

1. Дистанційне навчання у вищій школі Ляшенко І.В. [Електронний ресурс] – Режим доступу до ресурсу: https://www.narodnaosvita.kiev.ua/?page_id=2682
2. ЦЕНТР ПІДВИЩЕННЯ КВАЛІФІКАЦІЇ ТА ДИСТАНЦІЙНОЇ ОСВІТИ[Електронний ресурс] – Режим доступу до ресурсу: <https://knutd.edu.ua/university/institutes/izdo/tsentr-distantsijnoyi-osviti/>
3. Абдалова О. И. Использование технологий электронного обучения в учебном процессе / О. И. Абдалова О. Ю. Исакова // Дистанц. и виртуал. обучение. – 2014. – № 12. – С. 50-55
4. Алієв Х. М. Дистанційні освітні технології: сутність, ознаки, особливості [Електронний ресурс] / Х. М. Алієв // Педагогічний альманах. – 2017. – №35. – С. 36-40. – Режим доступу: http://nbuv.gov.ua/UJRN/pedalm_2017_35_7
5. Биков В.Ю. Проектний підхід і дистанційне навчання у професійній підготовці управлінських кадрів [Електронний ресурс] / В.Ю. Биков. – Режим доступу: <http://www.ime.edu-ua.net/cont/Bykov1.doc>
6. Жуковський В. Особливості сучасного дистанційного навчання: потенціал та нові можливості використання [Електронний ресурс] / В. Жуковський, С. Галецький // Людинознавчі студії. Педагогіка. – 2018. – №6. – С. 154–165. – Режим доступу: http://nbuv.gov.ua/UJRN/Lstud_2018_6_15

7. Pardanjac M., Radosav D., Jokic S. Difficult iesand possibilities of distance learning //Intelligent System sand Informatics, 2009. SISY'09. 7th International Symposiumon. – IEEE, 2009. – С. 361–365.
8. Дмитро Виноградов: Можливості Lifecycle. [Електронний ресурс] – Режим доступу до ресурсу: <https://startandroid.ru/ru/courses/architecture-components/27-course/architecture-components/524-urok-1.html>
9. IntelliJ Idea — Краткое руководство. [Електронний ресурс] – Режим доступу до ресурсу: <https://coderlessons.com/tutorials/java-tehnologii/izuchite-intellidzh-ideia/intellij-idea-kratkoe-rukovodstvo>
10. Newand Improved You Track Plugin For IntelliJ-based IDEs[Електронний ресурс] – Режим доступу до ресурсу: <https://blog.jetbrains.com/idea/2021/06/new-and-improved-youtrack-plugin-for-intellij-based-ides/>

ДОДАТОК Б

Перелік програмних файлів

AddActivity.java

```
package org.nmu.dnipropolytechnicenglish.ui;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import org.nmu.dnipropolytechnicenglish.databinding.ActivityAddBinding;
import org.nmu.dnipropolytechnicenglish.model.WordModel;
import org.nmu.dnipropolytechnicenglish.repository.Repository;

public class AddActivity extends AppCompatActivity {

    private ActivityAddBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityAddBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setSupportActionBar(binding.toolbar);

        binding.fab.setOnClickListener(v -> {

            String s1 = binding.textInputLayout.getText().toString();
            String s2 = binding.textInputLayout2.getText().toString();
            String s3 = binding.textInputLayout3.getText().toString();

            if (s1.trim().isEmpty()) binding.textInputLayout.setError("Cannot be empty");
            if (s2.trim().isEmpty()) binding.textInputLayout2.setError("Cannot be empty");

            /*
```

Додаємо слово в бд */

```
if (!s1.trim().isEmpty() && !s1.trim().isEmpty()) {
    Repository.database().dao().insert(new WordModel(
        s1,
        s3,
        s2
    ));
    finish();
}

});

}
```

MainActivity.java

```
package org.nmu.dnipropolytechnicenglish.ui;
```

```
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.fragment.NavHostFragment;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
```

```
import com.google.android.material.dialog.MaterialAlertDialogBuilder;
import com.google.android.material.snackbar.Snackbar;
```

```

import org.nmu.dnipropolytechnicenglish.R;
import org.nmu.dnipropolytechnicenglish.databinding.ActivityMainBinding;
import org.nmu.dnipropolytechnicenglish.util.Const;

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding;

    private NavController navController;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setSupportActionBar(binding.toolbar);
        loadNavController();
    }

    /*
    Створюємо навігацію. Бібліотека все робить автоматично
    використовуємо чин nav_graph.xml
    */

    private void loadNavController() {
        navController = ((NavController)
            getSupportFragmentManager().findFragmentById(binding.navHostFragment.getId()).getNavController());
        AppBarConfiguration appBarConfiguration = new
            AppBarConfiguration.Builder(binding.bottomNavigation.getMenu()).build();
        NavigationUI.setupActionBarWithNavController(this, navController,
            appBarConfiguration);
        NavigationUI.setupWithNavController(binding.bottomNavigation,
            navController);
        navController.navigate(R.id.learnFragment);
    }

    @Override

```

```

public void onBackPressed() {
    new MaterialAlertDialogBuilder(this)
        .setTitle("Exit")
        .setMessage("Do you really want to quit the app?")
        .setPositiveButton("Exit", (dialog, which) -> finishAffinity())
        .setNegativeButton("Cancel", null)
        .show();
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_app_bar, menu);
    return super.onCreateOptionsMenu(menu);
}

```

```

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    switch (item.getItemId()) {

        case R.id.item_help:

            Snackbar.make(binding.getRoot(), "Check Out!",
                Snackbar.LENGTH_SHORT)
                .setAnchorView(binding.bottomNavigation)
                .setAction("Open", v -
                    >startActivity(new Intent(Intent.ACTION_VIEW,
                        Uri.parse(Const.UNIVERSITY_SITE_URL))))
                .show();

            break;

    }

    return super.onOptionsItemSelected(item);
}
}

```

TestFragmentViewModel.java

```

package org.nmu.dnipropolytechnicenglish.ui.test;

import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

import org.nmu.dnipropolytechnicenglish.model.WordModel;
import org.nmu.dnipropolytechnicenglish.repository.Repository;
import org.nmu.dnipropolytechnicenglish.repository.db.AppDatabase;
import org.nmu.dnipropolytechnicenglish.util.Const;

public class TestFragmentViewModel extends ViewModel {

    public MutableLiveData<WordModel> word = new MutableLiveData<>();

    public MutableLiveData<Const.ErrorStatus> status =
new MutableLiveData<>();

    /*
    Якщо слово правильне-SUCCESS
    Інакше-ERROR

    loadApp() - загрузаем новое слово */
    public void checkWord(String s) {
        if
(word.getValue().getWordTranslated().toLowerCase().equals(s.toLowerCase())
) {
            status.setValue(Const.ErrorStatus.SUCCESS);
            loadApp();
        } else status.setValue(Const.ErrorStatus.ERROR);
    }

    public void loadApp() {
        status.setValue(Const.ErrorStatus.NONE);

        AppDatabase appDatabase = Repository.database();

```

```
word.setValue(appDatabase.dao().getByRandom());

    }

}
```

SplashActivity.java

```
package org.nmu.dnipropolytechnicenglish.ui.splash;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.lifecycle.ViewModelProvider;

import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.GoogleAuthProvider;

import org.nmu.dnipropolytechnicenglish.R;
import org.nmu.dnipropolytechnicenglish.databinding.ActivitySplashBinding;
import org.nmu.dnipropolytechnicenglish.ui.MainActivity;
import org.nmu.dnipropolytechnicenglish.util.Const;

public class SplashActivity extends AppCompatActivity {
```

```

private ActivitySplashBinding binding;
private SplashActivityViewModel viewModel;
private GoogleSignInClient mGoogleSignInClient;
private FirebaseAuth mAuth;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivitySplashBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    mAuth = FirebaseAuth.getInstance();

    viewModel = new
ViewModelProvider(this).get(SplashActivityViewModel.class);

    binding.fabNext.setOnClickListener(v -> viewModel.loadApp());
    binding.tvLink.setOnClickListener(v -> startActivity(new
Intent(Intent.ACTION_VIEW, Uri.parse(Const.UNIVERSITY_SITE_URL))));

/*
Перевіряємо завантажили чи слова в базу даних */

    viewModel.getStatus().observe(this, status -> {
        switch (status) {
            case LOADING:
                binding.fabNext.setText("Loading...");

binding.fabNext.setIconResource(R.drawable.ic_round_loop_24);
                break;

/*
Якщо ми не увійшли в аккаунт-необхідна авторизація */

            case SUCCESS:
                if (FirebaseAuth.getInstance().getCurrentUser() != null) {
                    startActivity(new Intent(this, MainActivity.class));
                    finish();
                }
            }
        }
    });

```



```

    } else {

        GoogleSignInOptions gso = new
        GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestIdToken("419557326703-
e5kdkl6p3pd6i6nu6puc9dmggc9h0mrj.apps.googleusercontent.com")
            .requestEmail()
            .build();

        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

        signIn();

    }
    break;
}
});

}

/*
Створюємо інтент для авторизації
*/
private void signIn() {
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, 100);
}

/*
После авторизации через гугл, авторизируем файербейс
*/
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 100) {
        Task<GoogleSignInAccount> task =
        GoogleSignIn.getSignedInAccountFromIntent(data);

```

```

    try {
        GoogleSignInAccount account =
task.getResult(ApiException.class);
        firebaseAuthWithGoogle(account.getIdToken());
    } catch (ApiException e) {

    }
}
}

/*
Після авторизації з файрбейс-завантажуємо додаток
*/
private void firebaseAuthWithGoogle(String idToken) {
    AuthCredential credential = GoogleAuthProvider.getCredential(idToken,
null);
    mAuth.signInWithCredential(credential)
        .addOnCompleteListener(this, task -> {
            if (task.isSuccessful()) {
                startActivity(new Intent(this, MainActivity.class));
                finish();
            } else {

            }
        });
}
}
}

```

ExploreFragment.java

```
package org.nmu.dnipropolytechnicenglish.ui.explore;
```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.ViewModelProvider;
import androidx.recyclerview.widget.LinearLayoutManager;

import com.google.android.material.dialog.MaterialAlertDialogBuilder;
import com.google.android.material.transition.MaterialFadeThrough;

import org.nmu.dnipropolytechnicenglish.adapter.RecyclerViewListAdapter;
import
org.nmu.dnipropolytechnicenglish.databinding.FragmentExploreBinding;
import org.nmu.dnipropolytechnicenglish.model.WordModel;
import org.nmu.dnipropolytechnicenglish.ui.AddActivity;

public class ExploreFragment extends Fragment {

    private FragmentExploreBinding binding;
    private ExploreFragmentViewModel viewModel;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setEnterTransition(new MaterialFadeThrough());
        setExitTransition(new MaterialFadeThrough());
    }

    /*
    Тут працює view model і завантажує список і оновлює
    його автоматично */
    @Override
    public View onCreateView(LayoutInflater inflater,
        ViewGroup container,

```

```

BundlesavedInstanceState) {
binding = FragmentExploreBinding.inflate(inflater, container, false);

viewModel =
new ViewModelProvider(this).get(ExploreFragmentViewModel.class);

binding.fab.setOnClickListener(v ->startActivity(new Intent(requireActivity(),
AddActivity.class)));

binding.progressCircular.setVisibility(View.VISIBLE);
binding.recyclerView.setVisibility(View.GONE);

viewModel.getWords().observe(getViewLifecycleOwner(), words -> {

binding.recyclerView.setAdapter(null);
RecyclerViewListAdapteradapter = new RecyclerViewListAdapter(words,
this);
LinearLayoutManagerlinearLayoutManager =
newLinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL,
false);

binding.recyclerView.setLayoutManager(linearLayoutManager);
binding.recyclerView.setAdapter(adapter);

binding.progressCircular.setVisibility(View.GONE);
binding.recyclerView.setVisibility(View.VISIBLE);

});

returnbinding.getRoot();
}

/*
Клік на айтем зі списку */
publicvoidonItemClick(WordModelwordModel) {

```

```

newMaterialAlertDialogBuilder(requireContext())
    .setTitle("Delete")
    .setMessage("Areyousureyouwanttodelete «" +
wordModel.getWordOriginal() + " - " + wordModel.getWordTranslated() + "»
?")
    .setPositiveButton("Delete", (dialog, which) -
>viewModel.deleteWord(wordModel))
    .setNegativeButton("Cancel", null)
    .show();

}

@Override
publicvoidonDestroyView() {
super.onDestroyView();
binding = null;
}

}

```

SplashActivityViewModel.java

```

packageorg.nmu.dnipropolytechnicenglish.ui.splash;

importandroidx.lifecycle.LiveData;
importandroidx.lifecycle.MutableLiveData;

```

```

import androidx.lifecycle.ViewModel;

import com.google.gson.Gson;

import org.nmu.dnipropolytechnicenglish.model.Response;
import org.nmu.dnipropolytechnicenglish.repository.Repository;
import org.nmu.dnipropolytechnicenglish.repository.db.AppDatabase;
import org.nmu.dnipropolytechnicenglish.util.Const;

import java.util.ArrayList;

public class SplashActivityViewModel extends ViewModel {

    private MutableLiveData<Const.LoadingStatus> status;

    public LiveData<Const.LoadingStatus> getStatus() {
        if (status == null) {
            status = new MutableLiveData<>();
        }
        return status;
    }

    /*
    Завантажуємо слова в базу даних, якщо їх немає */

    public void loadApp() {
        status.setValue(Const.LoadingStatus.LOADING);

        AppDatabase appDatabase = Repository.database();

        if (appDatabase.dao().getAllStatic().isEmpty()) {
            Response response =
            new Gson().fromJson(Repository.instance().loadJsonBackup(), Response.class);
            appDatabase.dao().insert(new ArrayList<>(response.getWords()));
        }
        status.setValue(Const.LoadingStatus.SUCCESS);
    }
}

```

```

    }
}

```

LearnFragment.java

```

package org.nmu.dnipropolytechnicenglish.ui.learn;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.ViewModelProvider;

import com.google.android.material.transition.MaterialFadeThrough;

import org.nmu.dnipropolytechnicenglish.databinding.FragmentLearnBinding;

public class LearnFragment extends Fragment {

    private FragmentLearnBinding binding;

    private LearnFragmentViewModel viewModel;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setEnterTransition(new MaterialFadeThrough());
        setExitTransition(new MaterialFadeThrough());
    }

    /*

```

View model робить все сама, завантажує слова при необхідності за допомогою

випадкового числа */

```

    @Override
    public View onCreateView(LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState) {
        binding = FragmentLearnBinding.inflate(inflater, container, false);

        viewModel =
            new ViewModelProvider(this).get(LearnFragmentViewModel.class);
        viewModel.loadApp();
        binding.setViewModel(viewModel);
        binding.setLifecycleOwner(getViewLifecycleOwner());
    }

```

```

    return binding.getRoot();
}

```

```

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}

```

LearnFragmentViewModel.java

```

package org.nmu.dnipropolytechnicenglish.ui.learn;

```

```

import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;

```

```

import org.nmu.dnipropolytechnicenglish.model.WordModel;
import org.nmu.dnipropolytechnicenglish.repository.Repository;
import org.nmu.dnipropolytechnicenglish.repository.db.AppDatabase;

```

```

public class LearnFragmentViewModel extends ViewModel {

```

```

    public MutableLiveData<WordModel> word = new MutableLiveData<>();
}

```



```
    /*  
    Завантажуємо слова з бд    */  
    public void loadApp() {  
  
        AppDatabase appDatabase = Repository.database();  
        word.setValue(appDatabase.dao().getByRandom());  
  
    }  
  
}
```

ДОДАТОК Б.2

compiler.xml

```
<?xmlversion="1.0" encoding="UTF-8"?>
<projectversion="4">
<componentname="CompilerConfiguration">
<bytecodeTargetLeveltarget="11" />
</component>
</project>
```

gradle.xml

```
<?xmlversion="1.0" encoding="UTF-8"?>
<projectversion="4">
<componentname="GradleMigrationSettings" migrationVersion="1" />
<componentname="GradleSettings">
<optionname="linkedExternalProjectsSettings">
<GradleProjectSettings>
<optionname="testRunner" value="PLATFORM" />
<optionname="distributionType" value="DEFAULT_WRAPPED" />
<optionname="externalProjectPath" value="$PROJECT_DIR$" />
<optionname="modules">
<set>
<optionvalue="$PROJECT_DIR$" />
</set>
</option>
<optionname="resolveModulePerSourceSet" value="false" />
</GradleProjectSettings>
</option>
</component>
</project>
```

jarRepositories.xml

```

<?xmlversion="1.0" encoding="UTF-8"?>
<projectversion="4">
<componentname="RemoteRepositoriesConfiguration">
<remote-repository>
<optionname="id" value="central" />
<optionname="name" value="MavenCentralrepository" />
<optionname="url" value="https://repo1.maven.org/maven2" />
</remote-repository>
<remote-repository>
<optionname="id" value="jboss.community" />
<optionname="name" value="JBossCommunityrepository" />
<optionname="url"
value="https://repository.jboss.org/nexus/content/repositories/public/" />
</remote-repository>
<remote-repository>
<optionname="id" value="MavenRepo" />
<optionname="name" value="MavenRepo" />
<optionname="url" value="https://repo.maven.apache.org/maven2/" />
</remote-repository>
<remote-repository>
<optionname="id" value="BintrayJCenter" />
<optionname="name" value="BintrayJCenter" />
<optionname="url" value="https://jcenter.bintray.com/" />
</remote-repository>
<remote-repository>
<optionname="id" value="Google" />
<optionname="name" value="Google" />
<optionname="url" value="https://dl.google.com/dl/android/maven2/" />
</remote-repository>
</component>
</project>

```

misc.xml

```

<optionname="myNullables">
<value>
<listsize="14">

```

```

    <itemindex="0" class="java.lang.String"
    itemvalue="org.jetbrains.annotations.Nullable" />
    <itemindex="1" class="java.lang.String"
    itemvalue="edu.umd.cs.findbugs.annotations.Nullable" />
    <itemindex="2" class="java.lang.String"
    itemvalue="android.support.annotation.Nullable" />
    <itemindex="3" class="java.lang.String"
    itemvalue="androidx.annotation.Nullable" />
    <itemindex="4" class="java.lang.String"
    itemvalue="androidx.annotation.RecentlyNullable" />
    <itemindex="5" class="java.lang.String"
    itemvalue="com.android.annotations.Nullable" />
    <itemindex="6" class="java.lang.String"
    itemvalue="javax.annotation.Nullable" />
    <itemindex="7" class="java.lang.String"
    itemvalue="javax.annotation.CheckForNull" />
    <itemindex="8" class="java.lang.String"
    itemvalue="org.checkerframework.checker.nullness.qual.Nullable" />
    <itemindex="9" class="java.lang.String"
    itemvalue="org.checkerframework.checker.nullness.compatqual.NullableDecl"
    />
    <itemindex="10" class="java.lang.String"
    itemvalue="org.checkerframework.checker.nullness.compatqual.NullableType"
    />
    <itemindex="11" class="java.lang.String"
    itemvalue="org.eclipse.jdt.annotation.Nullable" />
    <itemindex="12" class="java.lang.String"
    itemvalue="io.reactivex.annotations.Nullable" />
    <itemindex="13" class="java.lang.String"
    itemvalue="io.reactivex.rxjava3.annotations.Nullable" />
</list>
</value>
</option>
<optionname="myNotNulls">
<value>
<listsize="13">
<itemindex="0" class="java.lang.String"
itemvalue="org.jetbrains.annotations.NotNull" />

```

```

    <itemindex="1" class="java.lang.String"
    itemvalue="edu.umd.cs.findbugs.annotations.NonNull" />
    <itemindex="2" class="java.lang.String"
    itemvalue="android.support.annotation.NonNull" />
    <itemindex="3" class="java.lang.String"
    itemvalue="androidx.annotation.NonNull" />
    <itemindex="4" class="java.lang.String"
    itemvalue="androidx.annotation.RecentlyNonNull" />
    <itemindex="5" class="java.lang.String"
    itemvalue="com.android.annotations.NonNull" />
    <itemindex="6" class="java.lang.String"
    itemvalue="javax.annotation.Nonnull" />
    <itemindex="7" class="java.lang.String"
    itemvalue="org.checkerframework.checker.nullness.qual.NonNull" />
    <itemindex="8" class="java.lang.String"
    itemvalue="org.checkerframework.checker.nullness.compatqual.NonNullDecl"
    />
    <itemindex="9" class="java.lang.String"
    itemvalue="org.checkerframework.checker.nullness.compatqual.NonNullType
    " />
    <itemindex="10" class="java.lang.String"
    itemvalue="org.eclipse.jdt.annotation.NonNull" />
    <itemindex="11" class="java.lang.String"
    itemvalue="io.reactivex.annotations.NonNull" />
    <itemindex="12" class="java.lang.String"
    itemvalue="io.reactivex.rxjava3.annotations.NonNull" />
  </list>
</value>
</option>
</component>
<componentname="ProjectRootManager" version="2"
languageLevel="JDK_11" default="true" project-jdk-name="1.8" project-jdk-
type="JavaSDK">
  <outputurl="file://$PROJECT_DIR$/build/classes" />
</component>
<componentname="ProjectType">
  <optionname="id" value="Android" />
</component>

```

```
</project>
```

modules.xml

```
<?xmlversion="1.0" encoding="UTF-8"?>
  <projectversion="4">
    <componentname="ProjectModuleManager">
      <modules>
        <modulefileurl="file://$PROJECT_DIR$/.idea/modules/xxx.iml"
filepath="$PROJECT_DIR$/.idea/modules/xxx.iml" />
      </modules>
    </component>
  </project>
```

workspace.xml

```
<?xmlversion="1.0" encoding="UTF-8"?>
  <projectversion="4">
    <componentname="AutoImportSettings">
      <optionname="autoReloadType" value="SELECTIVE" />
    </component>
    <componentname="ChangeListManager">
      <listdefault="true" id="36833a68-8091-47bc-9576-7d02189a1183"
name="DefaultChangelist" comment="" />
      <optionname="SHOW_DIALOG" value="false" />
      <optionname="HIGHLIGHT_CONFLICTS" value="true" />
      <optionname="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false"
/>
      <optionname="LAST_RESOLUTION" value="IGNORE" />
    </component>
    <componentname="ExternalProjectsData">
      <projectStatepath="$PROJECT_DIR$">
        <ProjectState />
      </projectState>
    </component>
    <componentname="ProjectId" id="1uG9NyTYZizwPP15FqfQWP4cVTC" />
    <componentname="ProjectViewState">
      <optionname="hideEmptyMiddlePackages" value="true" />
    </component>
  </project>
```

```

<optionname="showLibraryContents" value="true" />
</component>
<componentname="PropertiesComponent">
  <propertyname="RunOnceActivity.OpenProjectViewOnStart" value="true"
/>
  <propertyname="RunOnceActivity.ShowReadmeOnStart" value="true" />
  <propertyname="settings.editor.selected.configurable"
value="preferences.lookFeel" />
</component>
<componentname="SpellCheckerSettings" RuntimeDictionaries="0"
Folders="0" CustomDictionaries="0" DefaultDictionary="application-level"
UseSingleDictionary="true" transferred="true" />
<componentname="TaskManager">
  <taskactive="true" id="Default" summary="Defaulttask">
    <changelistid="36833a68-8091-47bc-9576-7d02189a1183"
name="DefaultChangelist" comment="" />
    <created>1624281025276</created>
    <optionname="number" value="Default" />
    <optionname="presentableId" value="Default" />
    <updated>1624281025276</updated>
  </task>
</servers />
</component>
</project>

```

ВІДГУК ДИПЛОМНОГО КЕРІВНИКА

на дипломну роботу бакалавра «Розробка мобільного додатку для вивчення мов» студента групи 126-17-1 Ярмака Романа Юрійовича

Робота присвячена розробці мобільного додатку в середовищі Android, призначеного для вивчення мов. Розробка Android-додатків є важливим напрямком розвитку сучасних ІТ-технологій, що обумовлює своєчасність та актуальність представленої роботи. Необхідно відзначити, що незважаючи на велику кількість на ринку додатків для вивчення мов, не існує їх представників, який би відповідали всім призначеним для користувача вимогам.

Метою дипломного проекту є розробка Android, що полегшує вивчення мов з використанням мобільних пристроїв. В результаті виконання роботи подібний додаток було створено. Розробка виконувалася в середовищі Android Studio, яка вважається однією з найбільш успішних в своєму класі.

Робота виконана на прийнятному методичному рівні, з високою часткою самостійності, пояснювальна записка оформлена згідно діючих стандартів. Основний і найбільш істотний недолік роботи - слабкий функціонал розробленого додатка, яке є працездатним, але недостатнім для досягнення поставленої мети.

Загалом дипломна робота відповідає вимогам, що пред'являються до кваліфікаційних робіт бакалавра. Вважаю, що робота заслуговує оцінки "задовільно", а студент Ярмак Р.Ю. присвоєння йому кваліфікації бакалавра.

Дипломний керівник, докт. геол. наук,

проф.каф. інформаційних технологій
та комп'ютерної інженерії

С.Л. Нікулін

**Керівник кваліфікаційної роботи,
асистент кафедри ІТКІ**

Додаток Г

РЕЦЕНЗІЯ

на дипломну роботу бакалавра

**«Розробка мобільного додатку для вивчення мов»
студента групи 126-17-1 Ярмака Р.Ю.**

Метою дипломної роботи і покращення умов для вивчення іноземних мов в умовах дистанційного навчання студентів. Для досягнення цієї мети авторами запропоновано розробити відповідний додаток у середовищі Android. Об'єктом дослідження є процес розробки мобільних Android-додатків, спрямованих на вивчення іноземних мов.

Дана тема є актуальною за низкою причин. Зокрема, ОС Android є однією з найрозповсюдніших, а розробка додатків до неї є затребуваною на ринку праці. З іншого боку, додатки, що полегшують вивчення мов, стрімко розвиваються, хоча мають величезний потенціал подальшого розвитку.

В процесі роботи створено мобільний додаток, що реалізує функції: самостійне заповнення словника користувачем, авторизацію за допомогою Google-акаунту, назку інструментів для засвоєння матеріалу та інше.

Але в додатку є суттєвий недолік, що полягає у дуже невеликому спектрі підходів та засобів вивчення мов. Тому додаток програє у функціональності аналогам, які можна знайти у Google Play Store.

Зважаючи на зазначений недолік, вважаю що дипломна робота студента Ярмака Романа Юрійовича виконана в межах існуючих вимог до робіт бакалаврів, і заслуговує оцінки «задовільно»

Рецензент,

канд. техн. наук, доцент кафедри ПЗКС

Коротенко Л.М.

Додаток Д

**Протокол аналізу звіту подібності науковим
керівником**

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Роман Ярмак

Співавтор: Оксана Коваль

Назва: Бакалаврская дипломная работа Ярмак, Коваль.pdf

Науковий керівник: Професор Сергій Нікулін

Підрозділ: Кафедра інформаційних технологій та комп'ютерної інженерії

Коефіцієнт подібності 1: 5%

Коефіцієнт подібності 2: 4.9%

Мікропробіли: 0

Заміна букв: 7

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2021-06-22 13:19:34.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2021-06-22

Професор Сергій Нікулін

Дата

эксперт