

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
бакалавр
(ступінь вищої освіти)

на тему РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ
ПРАВИЛ ДОРОЖНЬОГО РУХУ
DEVELOPMENT OF A MOBILE APPLICATION FOR LEARNING
TRAFFIC RULES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-120
Спеціальності 121 Інженерія програмного
забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

БЕЙНИК А.О
(ПРІЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.
(ПРІЗВИЩЕ та ініціали)

Рецензент ТЕРЕЩЕНКО Е.В.
(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(КОД І НАЙМЕНУВАННЯ)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
« » 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Бейник Анастасії Олексіївної
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка мобільного застосунку для вивчення правил дорожнього руху
Development of a Mobile Application for Learning Traffic Rules

керівник проєкту (роботи) к.т.н., ШИТІКОВА Олена Вікторівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «24» квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Матеріали і методи. 3. Проєктування застосунку. 4. Розробка застосунку
5. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ШИТІКОВА О.В., доцент кафедри ПЗ	24.04.2024	Підпис
Нормоконтроль	Дейнега Л.Ю., ст. викл кафедри ПЗ	11.06.2024	Підпис

7. Дата видачі завдання “ 24 ” квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	При мітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

Підпис

(підпис)

Анастасія БЕЙНИК

(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Підпис

(підпис)

Олена ШИТІКОВА

(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
166 с., 15 табл., 55 рис., 3 дод., 14 джерел.

МОБІЛЬНИЙ ЗАСТОСУНОК, ПРАВИЛА ДОРОЖНЬОГО РУХУ,
JAVA, ANDROID.

Об'єкт дослідження – засоби вивчення правил дорожнього руху.

Предмет дослідження – мобільні застосунки для вивчення правил дорожнього руху.

Мета роботи – створення мобільного застосунку для вивчення та перевірки знань правил дорожнього руху.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, метод флеш-карток, мови програмування Java, мобільний пристрій з операційною системою Android.

Результати. Створено програмний Android-застосунок, який у формі флеш карток надає можливість швидкому та ефективному вивченню матеріалу.

Висновки. Розроблено програмний засіб для вивчення правил дорожнього руху з використанням актуальних методів навчання.

Галузь використання – особисте використання для вивчення правил та їх практика в домашніх умовах.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 166 pages, 15 tables, 55 figures, 3 appendixes, 14 sources.

MOBILE APPLICATION, TRAFFIC RULES, JAVA, ANDROID.

The object of research is -tools for studying traffic rules.

The subject of research is mobile applications for studying traffic rules.

The purpose of the work is to create a mobile application for studying and testing knowledge of traffic rules.

Materials, methods and technical means: structural and object-oriented programming, the method of flash cards, Java programming languages, a mobile device with the Android operating system.

The results. An Android software application has been created, which in the form of a flash card enables quick and efficient use of the material.

Conclusions. A software tool has been developed for studying traffic rules using current teaching methods.

The field of use is personal use for learning the rules and their practice at home.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ	9
1 Аналіз предметної області	11
1.1 Актуальність теми розробки.....	11
1.2 Аналіз методів навчання	12
1.3 Дослідження програмних засобів для вивчення пдр.....	16
1.3.1 Застосунок пдр 2024	16
1.3.2 Застосунок офіційні пдр україни та тести.....	20
1.4 Постановка завдання роботи.....	22
1.5 Висновки за розділом 1	22
2 Матеріали і методи	23
2.1 Порівняння та аналіз мов програмування	23
2.1.1 Java.....	24
2.1.2 Kotlin	25
2.1.3 Порівняння мов програмування	26
2.2 Аналіз середовищ розробки	28
2.3 Аналіз системи автоматично складання.....	31
2.4 Висновок за розділом.....	34
3 Проєктування застосунку.....	35
3.1 Функціональні вимоги до програми та інтерфейсу.....	35
3.2 Розробка користувацького інтерфейсу	35
3.3 Розробка архітектури застосунку	45
3.4 Висновок за розділом 3	47
4 Розробка застосунку	48
4.1 Структурна схема програми.....	48
4.2 Файли, методи та функції.....	51
4.2.1 Модуль activity	51
4.2.2 Модуль Adapter	54

4.2.3 Модуль Data.....	55
4.2.4 Модуль Layout.....	56
4.3 Висновок за розділом.....	57
5 Експлуатація, тестування та експериментальне дослідження програми	58
5.1 Призначення та умови застосування програми	58
5.2 Опис роботи програми.....	58
5.3 Тестування програми	74
5.4 Висновки до 5 розділу	78
Висновки.....	79
Перелік джерел посилання.....	80
Додаток А Технічне завдання.....	82
Додаток Б Текст програми	88
Додаток В Слайди презентації	158

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- API – Application Programming Interface;
CDMA – Code Division Multiple Access;
IDE – Integrated Development Environment,
ПЗ – програмне забезпечення;
ПДР –правила дорожнього руху;
ОС - операційна система.

ВСТУП

Розробка мобільного застосунку для вивчення правил дорожнього руху (ПДР) – актуальна і важлива тема в сучасному світі де володіння особистим транспортним засобом вже давно розглядається не як розкіш, а базова потреба для пересування містом, країною тощо. Для керування автомобільним транспортом необхідно знати ПДР, які раніше обов'язково вивчалися в спеціалізованих школах разом з інструкторами.

13 липня 2023 року верховною радою України було внесені зміни до статті 15 Закону України "Про дорожній рух", одна з яких полягала в тому що особа, яка бажає отримати право на керування транспортними засобами зобов'язана самостійно опанувати зміст теоретичного модуля типової навчальної програми.[1]

Ритм життя сучасної людини занадто перевантажений необхідними справами для виживання. Тому все частіше майбутнім водіям важко виділити окремий вільний час для вивчення нового матеріалу. У зв'язку з цим розробка саме мобільного застосунку стане відмінним рішенням для зручності вивчення ПДР, адже можна скористатись ним в будь яку вільну можливість, наприклад, під час поїздки в громадському транспорті або всього лише 10 хвилин перерви на роботі.

Об'єкт дослідження – засоби вивчення правил дорожнього руху.

Предмет дослідження – мобільні застосунки для вивчення правил дорожнього руху.

Метою даної дипломної є розробка застосунку який дозволить не лише вчити а й відслідковувати свій прогрес користувача тестування застосунку.

Для досягнення зазначеної мети необхідно виконати наступні завдання:

- провести аналіз предметної області;
- дослідити існуючі додатки для вивчення правил дорожнього руху;
- обрати інструменти для розробки;

- розробити програму;
- провести тестування.

Основний результат розробки застосунку полягає в надані людям можливості освоювати отримувати необхідні їм знання в області ПДР, не зважаючи на протяжність кількість вільного часу чи місце знаходження користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність теми розробки

З повноцінним впровадженням діджиліталізації в усі сфери життя починаючи від наукових процесів закінчуючи базовими домашніми справами мобільні додатки стали тим самим незамінним ресурсом. Одна із важливих сфер, де за допомогою технологій було досягнуто скачок не тільки в швидкості, а й в доступності – це сфера освіти, а саме вивчення ПДР. Тим паче враховуючи нові зміни в закон України про можливості самовивчення правил [1], мобільні додатки з вивчення ПДР отримали неабияку популярність серед учнів.

Розробка мобільного застосунку для вивчення ПДР в форматі флеш-карт, облегшить і додатково стане полегшенням при запам'ятовуванні правил.

Знання правил дорожнього руху є базою не тільки для отримання посвідчення водія, але й для безпеки учасників дорожнього руху. В класичному варіанті правила дорожнього руху вивчалися на теоретичному курсі в спеціалізованих школах, але на разі в нашій країні громадянин має право скласти тест з ПДР пройшовши самопідготовку.

Також не варто забувати про зростання популярності симбіозу навчання та електронних девайсів, особливо після карантину у 2020 році через Covid-19, де увесь навчальний процес перейшов в дистанційний режим. Освіта завдяки мобільному девайсу поєднує в собі комфорт та портативність телефону, який в сучасних реаліях, майже кожную секунду знаходиться в руках користувача. Тим паче навчання завдяки телефону матиме прихильність серед молодих людей, так як вони виростили на постійній взаємодії зі смартфонами. Тому саме мобільний додаток буде доречним до розробки.

1.2 Аналіз методів навчання

Основна мета додатку полягає в швидкому і доступному навчанню. Для дослідження необхідно розглянути актуальні методики навчання:

а) метод sq3r (the sq3r method) – це аббревіатура, яка означає п'ять кроків процесу розуміння прочитаного[2]:

- 1) опитування (survey);
- 2) запитання (question);
- 3) прочитайте (read);
- 4) переказуйте (recite);
- 5) перегляньте (review);

б) практика розшуку (retrieval practice) – ґрунтується на концепції пригадування в більш пізній час. пригадування відповіді на питання покращує навчання більше, ніж пошук відповіді в підручнику;

в) розподілена практика (spaced practice) – заохочує студентів навчатися протягом більш тривалого періоду часу замість того, щоб зубрити матеріал напередодні ввечері;

г) метод rq4r (the rq4r method) rq4r - це аббревіатура, що означає шість кроків у процесі навчання:

1) попередній перегляд (preview): перегляньте інформацію перед тим, як почати читати, щоб отримати уявлення про тему. перегляньте матеріал і читайте тільки заголовки, підзаголовки та виділений текст;

2) запитання (question): задавайте собі питання, пов'язані з темою, наприклад, що я очікую дізнатися? що я вже знаю про цю тему?;

3) читайте (read): прочитайте інформацію по одному розділу за раз і спробуйте знайти відповіді на свої запитання;

4) поміркуйте (reflect): чи відповіли ви на всі свої запитання? якщо ні, поверніться назад і спробуйте знайти відповідь;

5) перекажіть (recite): своїми словами проговоріть або запишіть короткий зміст інформації, яку ви щойно прочитали;

6) повторення (review): перегляньте матеріал ще раз і дайте відповіді на всі питання, які залишилися без відповіді;

г) техніка фейнмана (the feynman technique) - це ефективний метод швидкого засвоєння концепції шляхом пояснення її простими і зрозумілими словами. він заснований на ідеї: "якщо ви хочете щось добре зрозуміти, спробуйте пояснити це просто"[2];

д) система лейтнера (leitner system) - це методика навчання, заснована на використанні флеш-карт;

е) кольорове кодування – це динамічний спосіб організувати інформацію, яку ви вивчаєте. для обозначення використовується така схема:

- 1) записуйте ключові моменти червоним кольором;
- 2) виділяйте важливу інформацію жовтим кольором;
- 3) організуйте теми за кольором;
- 4) не розфарбовуйте все - лише найважливішу інформацію;

є) картування свідомості (mind mapping)[2] – техніка організації інформації у вигляді графічної схеми, де до кожного блоку інформації додаються, нові вивчені блоки інформації пов'язані з темою[2].

Аналіз недоліків та переваг кожного методу подано у вигляді порівняльної таблиці 1.1.

Таблиця 1.1 – Порівняння методів навчання

Метод	Переваги	Недоліки	Доцільність імплементатції в мобільний додаток
SQ3R (The SQ3R Method)	- Організованість - Чіткість -Впорядковано представляє інформацію.	- Суб'єктивне тлумачення -Вимагає особистісної оцінки особи	Низька: Не можливо додати в програму усі необхідні пункти, адже необхідна суб'єктивна думка окремого користувача
Практика розшуку (Retrieval Practice)	Підсилює довготривалу пам'ять; допомагає виявляти прогалини в знаннях	Вимагає активного зусилля для відновлення інформації.	Висока: інтеграція тестів може підвищити ефективність, але може бути стресом, через сильні зусилля
Розподілена практика (Spaced Practice)	покращує довготривале запам'ятовування ; знижує когнітивне навантаження.	Вимагає ретельного планування; потребує тривалого часу для повного засвоєння матеріалу.	Висока: нагадування та сповіщення можуть підтримувати розподілене навчання.
Метод PQ4R (The PQ4R Method)	Критичний аналіз: структурує процес навчання.	Може бути складним для новачків; потребує значного часу. Суб'єктивна	Низька: складний та суб'єктивний метод

Продовження таблиці 1.1

Метод	Переваги	Недоліки	Доцільність імплементатії в мобільний додаток
Техніка Фейнмана (The Feynman Technique)	- Ідентифікація шаблону: Допомагає виявити зв'язки в інформації.	- Вимагає активного вивчення та розуміння.	Низька: Необхідна суб'єктивна думка кожного окремого користувача .
Система Лейтнера (Leitner System)	Ефективне використання флеш-карток; підсилює довготривалу пам'ять.	Вимагає систематичного використання.	Висока: цифрові картки легко інтегруються в додаток і підтримують адаптивне навчання.
Кольорове кодування	Покращує організацію та виділення важливої інформації; підсилює візуальне запам'ятовування	-Може бути неефективним для людей, які не схильні до візуального навчання;	Середня: інтерфейси можуть використовувати кольорове кодування для покращення організації інформації.
Картування свідомості (Mind Mapping)	Сприяє креативному мисленню; покращує організацію інформації.	Може бути складним для початківців; вимагає додаткового часу на створення карт.	Середня: інтерактивні карти можуть бути корисними для організації та розуміння складних тем.

Враховуючи проведений аналіз, можна побачити, що кожен метод має недоліки та переваги, але враховуючи потреби в ефективності, швидкості та доступності для додатку було обрано Систему Лейтнера, як основну з інтегруванням методу розподіленої практики для підвищення кількості навчальних сесій .

1.3 Дослідження програмних засобів для вивчення ПДР

На момент дослідження існує декілька програм саме для вивчення правил дорожнього руху у вигляді android-застосунків. Для вивчення програмних засобів було обрано декілька додатків з правилами

1.3.1 Застосунок ПДР 2024

ПДР 2024 – це офіційні ПДР України від видавництва моноліт. Скріншоти додатку представлені на рис. 1.1-1.3.

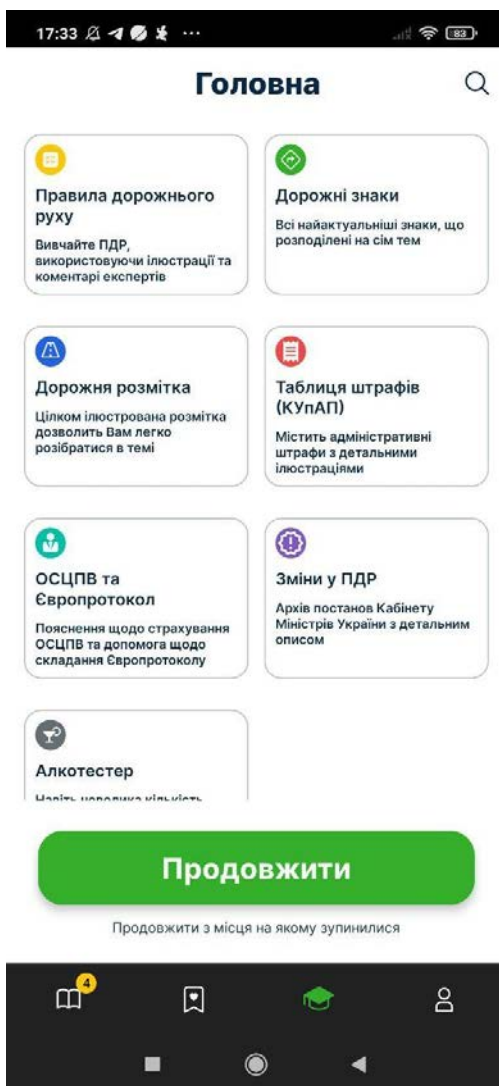


Рисунок 1.1 – Головна сторінка ПДР2024



Рисунок 1.2 – Сторінка з темами

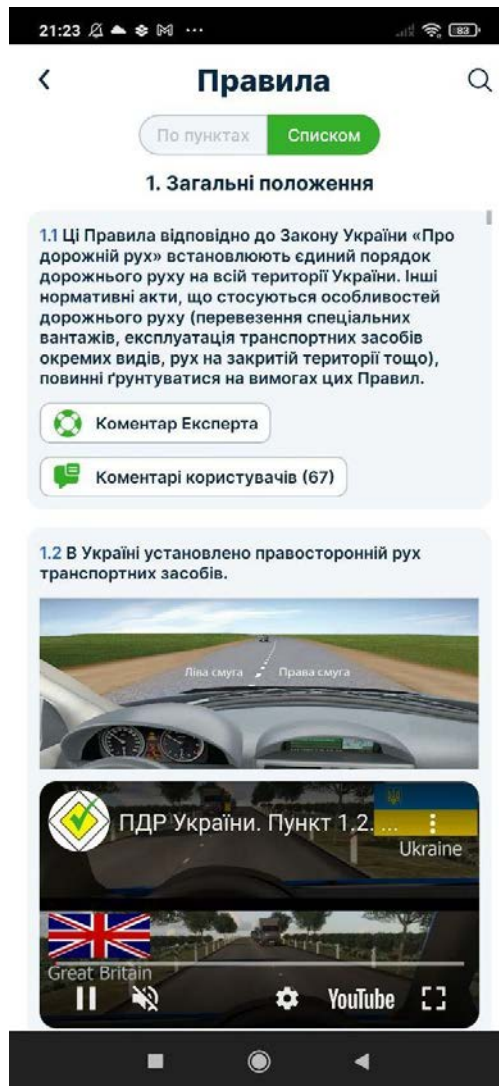


Рисунок 1.3 – Подання матеріалу

Переваги:

- розділення на теми;
- інтуїтивний інтерфейс;
- наявність коментаря експерта до правил;
- наявність графічних зображень;
- відео пояснення до деяких тем;
- збереження моменту де було завершено попередню взаємодію.

Недоліки:

- необхідність авторизації;

- викладення матеріалу без використання актуальних методик навчання;
- необхідний доступ до інтернету;
- відсутність перевірки знань;
- не можливо встановити на телефон фірми huawei.

1.3.2 Застосунок Офіційні ПДР України та тести

Це додаток з актуальним переліком правил та тестами до них. Скріншоти застосунку зображені на рис. 1.4 – 1.5.

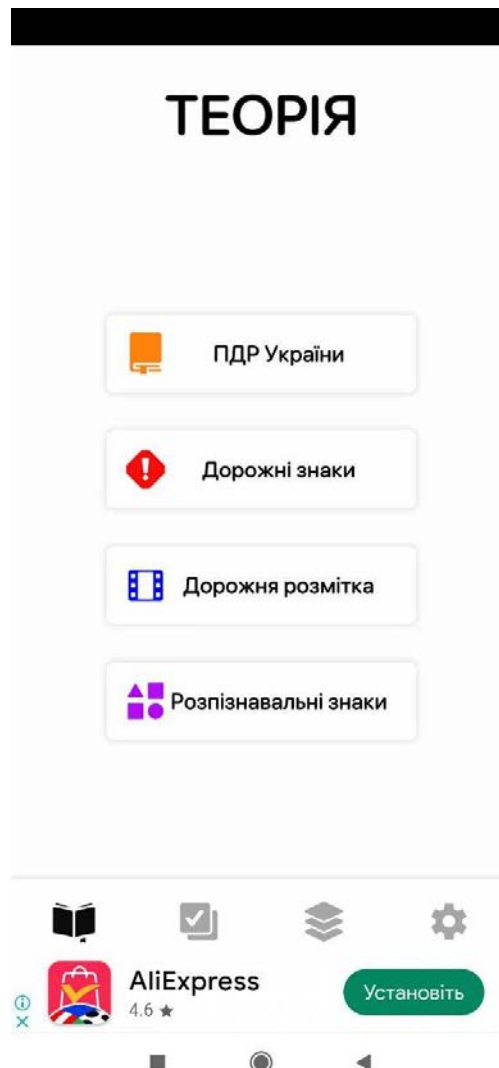


Рисунок 1.4 – Головна сторінка застосунку

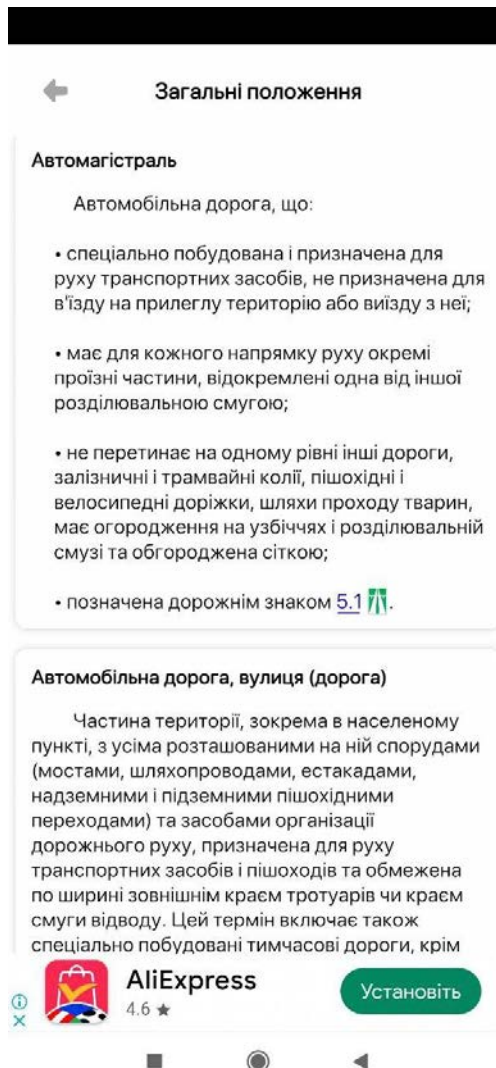


Рисунок 1.5 – Подання правил в застосунку

Переваги:

- інтуїтивний інтерфейс;
- розділення правил за темами;
- наявність тестування знань.

Недоліки:

- відсутність графічних зображень до правил ;
- наявність реклами;
- потреба у внутрішніх покупках;
- відсутність використання актуальних методів навчання;
- не можливо встановити на телефон фірми huawei.

1.4 Постановка завдання роботи

Після проведення дослідження наявних додатків для вивчення ПДР можна помітити відсутність використання актуальних методик навчання та інсталяції на телефони фірми Huawei.

Найменування додатку – «FlashPDR»

Для застосунку було виведено такі функціональні вимоги:

- відсутня необхідність авторизації;
- втілення актуального методу навчання, а саме використання флеш-карток;
- зробити додаток доступний для усіх смартфонів на основі ОС Android;
- наявність пошуку за словами;
- наявність графічних зображень.

Найменування застосунку «ПДР 2024 з флеш-картами». ОС для розробки Android.

1.5 Висновки за розділом 1

В першому розділі було проведено аналіз предметної області. Проведено огляд і порівняння існуючих застосунків з вивчення ПДР, виявлено їх переваги та недоліки. Після чого було сформульовано завдання роботи.

2 МАТЕРІАЛИ І МЕТОДИ

Мобільна революція створила привід для розвитку мобільних додатків, тим більше і сам телефон став незамінною одиницею в роботі, для деяких людей навіть замінивши використання комп'ютера чи ноутбуку. Також окрім телефонів неабияку популярність мають планшети, завдячуючи своїй мобільності та легкості в транспортуванні. Тому зі зростанням попиту на додатки, пропорційним є зростання кількості та якості мов програмування та технологій для мобільної розробки. Тому постає логічне запитання, яку мову та методи необхідно обрати для максимальної ефективності розробки.

2.1 Порівняння та аналіз мов програмування

В світі більше 3 мільярдів активних андроїд девайсів [3], сама по собі система відома та відрізняється саме гнучкістю у використанні саме тому зберігає своє лідерство серед представників мобільних операційних систем.

Одне із основних рішень в розробці мобільних застосунків полягає у виборі мов програмування, адже від цього залежить:

- швидкість розробки;
- продуктивність;
- додаткові можливості в застосовуванні розширень ;
- подальша підтримка проекту.

В цьому розділі наша мета полягатиме в аналізі та оцінці доцільності використання таких мов програмування:

- Kotlin;
- java.

2.1.1 Java

Java – це класова, об’єктно-орієнтована мова програмування, що розроблена компанією Sun Microsft, так щоб дозволити розробникам додатків написати код один раз і мати змогу запустити будь-де (WORA), що означає, що скомпільований код Java може працювати на всіх платформах, які підтримують Java, без необхідності повторної компіляції. Вона є швидкою, безпечною, надійною, багатоплатформеною та була створена для простоти використання.

Підтримує свій рівень популярності серед розробників завдяки особливим характеристикам:

а) доступність якісних ресурсів для навчання, так як java використовується доволі таки довго, її документація детально прописана, видано велику кількість книжок як для вивчення синтаксису, так із реальними прикладами застосування мови;

б) надання доступу до великої екосистеми вбудованих функцій та бібліотек, що дозволяє не писати абсолютно весь код самостійно;

в) активна підтримка, java має спільноту користувачів, що можуть допомогти при наявності проблеми, а також підтримується та регулярно оновлюється;

г) надання доступу до якісних інструментів розробника, таких як автоматичне редагування, дебагінг, тестування та інші. що дозволяє економити час;

г) безпека, java надає можливість загрузити будь-який код та запустити його в безпечному середовищі, де робота програми не завдасть шкоди комп’ютеру;

д) варіативність використання, java використовується для:

- 1) розробки ігор;
- 2) серверних технологій;

- 3) роботи з великими даними(big data);
- 4) штучного інтелекту;
- 5) інтернету речей[4].

2.1.2 Kotlin

Kotlin – це статично типізована, об'єктно орієнтована мова програмування.[10] Вона початково була створена, як покращення мови Java командою JetBrains та перший раз представлена в 2011. Це мова працює поверх платформи JVM. Завдяки взаємодії з Java, Kotlin можна використовувати для розробки:

- Android додатків(одна з ключових сфер застосування);
- серверної-частини додатків;
- веб-розробки;
- мультиплатформених додатків;
- в науці про дані[5];
- desktop-додатки[6].
- для контролю хмарних ресурсів розробленого додатку на Kotlin.

Не зважаючи на доволі таки короткий час присутності на полі Kotlin вже має прихильність серед розробників займаючи 13 місце в рейтингу популярності мов для розробки. Швидкий успіх пояснюється такими особливостями мови як:

- сумісність. сумісність з java зумовлена тим що обидві мови компілюються в однаковий байт код. тому код Kotlin може бути інтегровано в існуючі java-проекти. тобто можна додати нові функції не переписуючи проєкт з самого початку;
- безпека. Kotlin спроектовано для того аби допомагати з уникнення типових помилок наприклад пов'язаних з «null pointer», через відсутність зберігання значення null;

- чіткість. Kotlin зменшує кількість шаблонного коду, тим самим роблячи його стислими та економлячи розробнику час;
- підтримка інструментів. Kotlin має інструментальну підтримку для Android розробки;
- підтримка ком'юніті. зважаючи на молодість мови, вона має активне і зацікавлену общину яка працює над покращенням мови[5].

2.1.3 Порівняння мов програмування

Java та Kotlin є одними з найпопулярніших рішень для створення мобільних застосунків аби оприділитись з мовою для власного додатку краще всього скористуватися порівняльною таблицею 2.1

Таблиця 2.1 – Порівняння мов програмування

Параметр	Java	Kotlin
Performance	Добре оптимізована продуктивність на платформі Android завдяки Android Runtime (ART) і Just-In-Time (JIT) компіляції.	Висока продуктивність, схожа на Java.
Development Speed and Ease	Висока продуктивність на платформі Android завдяки Android Runtime (ART) і Just-In-Time (JIT) компіляції.	Підвищує продуктивність завдяки компактному синтаксису, зменшенню boilerplate коду і захисту від помилок, пов'язаних з null.
Community and Ecosystem	Величезна екосистема, численні бібліотеки, фреймворки та інструменти.	Зростаюча екосистема і спільнота, але не така велика, як у Java.

Продовження таблиці 2.1

Параметр	Java	Kotlin
Learning Curve	Велика кількість розробників знайомі з Java завдяки її поширеності в освітніх закладах і індустрії, що зменшує криву навчання і прискорює онбординг.	Потребує навчання для розробників, які звикли до Java, але синтаксис схожий і логічний.
Advantages	Стабільність, зрілість, широка підтримка, висока продуктивність, сумісність з Android SDK, великий обсяг документації та підтримки спільноти.	Сучасний синтаксис, зменшення коду, захист від помилок, сумісність з Java.
Disadvantages	Вербозність, що може призводити до більшої кількості boilerplate коду.	Менша екосистема та спільнота в порівнянні з Java, потребує навчання для Java-розробників.
Interoperability	Безшовна інтеграція з існуючими Android проектами і бібліотеками.	Повна сумісність з Java, дозволяє поступову міграцію з Java на Kotlin.

Java залишається сильною мовою для розробки програм для Android завдяки своїй зрілості, стабільності, великій екосистемі та високій продуктивності.

Незважаючи на те, що Kotlin пропонує більш сучасні функції та скорочений код, Java все ще є надійним і широко підтримуваним вибором для багатьох потреб розробки програм Android. Саме тому для виконання роботи обрано мову Java.

2.2 Аналіз середовищ розробки

Середовище розробки є одним з основних засад до успішної та ефективної розробки додатків. Адже в залежності від вибору ви отримуєте або втрачаєте додатковий доступ до розширень, які допоможуть пришвидшити та зробити більш ефективним для розробника процес тестування додатку. В цьому розділі розглянемо 3 популярні додатки для мобільної розробки: Android Studio, Eclipse IDE та IntelliJ IDEA.

Android Studio це офіційне Інтегроване середовище розробки (IDE) для розробки Android додатків. Яке спирається на потужний редактор коду та інструмент розробника IntelliJ IDEA. Андроїд студія надає більше додаткових можливостей, які підвищують продуктивність розробника під час створення додатку, наприклад:

- гнучка система побудови на основі gradle;
- швидкий та багатофункціональний емулятор;
- уніфікований простір розробки для всіх android девайсів;
- пряме оновлення компонентів в емуляторі та фізичних девайсах реальному часі;
- шаблони коду та інтеграція github для допомоги в побудови часто використовуємого функціоналу або імпортувати приклад коду;
- розширений набір тестувальних інструментів та структур;
- lint інструменти для відслідковування продуктивності, зручності використання та сумісності версій [7].

Eclipse IDE - відкрите інтегроване середовище розробки мовою програмування Java,[11] яке розповсюджується та підтримується Eclipse Foundation. [8] Ключовий функціонал за який розробники обирають саме це середовище розробки:

- покращена платформа;
- безкоштовний та відкритий ресурс релізу коду;

- покращенні підключенні інструменти розробника;
- підтримка ком'юніті.

IntelliJ IDEA – це IDE для Java та Kotlin мов, розроблена спеціально для того, аби максимізувати продуктивність розробника. Вона виконує рутинні та повторювальні завдання за розробника завдяки, розумному виконанню, статичному виконанню та рефракторингу коду. Це дозволяє розробнику зосередитися на розроблювальній системі та інноваційних підходах до вирішенні поставлених задач та дійсно отримати позитивний досвід від взаємодії. З поміж інших середовищ розробки саме IntelliJ IDEA виділяють такі риси як:

- підтримка мультиплатформеності;
- широкий вибір мов розробки не тільки для jvm (java virtual machine);
- вбудовані плагіни для інших мов;
- ергономічний та налаштовуваний під себе дизайн;
- наявність просунутих комбінацій та швидкими діями до майже всього;
- зручна навігація та пошук в проєкті;
- кодовий асистент, для базових та стандартних імен, функцій тощо;
- рефракторинг коду в усьому проєкті;
- інтегровані інструменти розробника з системою локальної історії змін [9].

Порівняння середовищ розробки зображено в таблиці 2.2

Таблиця 2.2 -Порівняння середовищ розробки

Параметр	Android Studio	IntelliJ IDEA	Eclipse
Основний фокус	Спеціалізоване середовище для розробки Android-додатків.	Універсальне середовище для розробки з підтримкою Android, веб, серверних додатків та багато іншого.	Універсальне середовище для розробки з широкою підтримкою різних мов та платформ.
Підтримка Android	Повна підтримка Android SDK, інтегровані інструменти для розробки, налагодження та тестування Android-додатків.	Повна підтримка Android SDK, інтеграція з Android Studio плагіном для повного досвіду розробки.	Підтримка Android через плагіни, але не така інтегрована як у Android Studio або IntelliJ IDEA.
Продуктивність і Швидкість	Оптимізована для розробки Android, але може бути повільною на великих проєктах або на машинах з низькою продуктивністю.	Висока продуктивність, особливо у версії Ultimate, менша вимогливість до ресурсів у порівнянні з Android Studio.	Може бути менш продуктивним і швидким через потребу встановлення численних плагінів для повної підтримки Android.
Функціональність	Вбудовані інструменти для розробки, налагодження, профілювання, тестування та емулявання Android-додатків.	Усі функції Android Studio, плюс додаткові інструменти для інших мов та платформ, розширена функціональність IDE.	Широка підтримка мов програмування та платформ, але потребує більше налаштування для Android.
Інтеграція з іншими інструментами	Широка інтеграція з Google інструментами, Firebase, та іншими Android сервісами.	Широка інтеграція з багатьма інструментами та сервісами, включаючи Android, веб і серверні технології.	Інтеграція з численними інструментами через плагіни, але менш інтуїтивна.
Підтримка Мов	Kotlin, Java, C++, та інші через плагіни.	Kotlin, Java, C++, Scala, Groovy, Python, та інші через плагіни.	Java, Kotlin, C++, і багато інших через плагіни.

Продовження таблиці 2.2

Параметр	Android Studio	IntelliJ IDEA	Eclipse
Підтримка Плагінів	Велика кількість плагінів для розширення функціональності, оптимізованих для Android.	Широка підтримка плагінів для різних мов та платформ, включаючи Android.	Велика кількість плагінів, але можуть бути проблеми зі стабільністю та продуктивністю.
Спільнота та Підтримка	Велика спільнота розробників Android, активна підтримка від Google.	Велика спільнота розробників різних мов, активна підтримка від JetBrains.	Велика спільнота, але не така спеціалізована на Android.
Навчальні Ресурси	Велика кількість ресурсів, документації, курсів та посібників для розробки Android-додатків.	Велика кількість ресурсів для різних мов і платформ, включаючи Android.	Менша кількість спеціалізованих ресурсів для Android, але багато загальних ресурсів для програмування.

Отже беручи до уваги вище перераховані переваги кожного середовища розробки, для роботи буде доцільно обрати саме IntelliJ IDEA через об'єктивну наявність комфорту для розробника та можливості додаткового використання інноваційних рішень.

2.3 Аналіз системи автоматично складання

Ручне складання проєктів на Java досить трудомісткий процес. Потрібно правильно вказати потрібні проєкту бібліотеки та фреймворки, від яких проєкт залежить[12]. Тому аби пришвидшити процес розробки, розробники пишуть скрипти для виконання завдань, що зменшує час побудови, але робить складнішим процес розробки командного проєкту.

Задля вирішення проблем були створені системи автоматичного складання систем. З актуальних на даний час систем зборки для Java виділяють Maven та Gradle.

Maven це система автоматичного складання[12] проєктів з використанням Java. Вона допомагає показати, як побудувати програму та пов'язати її з різними залежностями. Для описання проєкту для побудови використовується файл розширення .xml в якому описується проєкт для розробки, залежність проєктів від модулів та бібліотек, порядок побудови та необхідні плагіни. Maven має наперед визначені цілі для задач таких як пакування та компіляція.

Gradle це система автоматичного складання з повністю відкритим кодом та використовує концепти з Apache Maven та Apache Ant. Для конфігурації проєкту Gradle використовує предметно орієнтовану мову програмування (DSL) на основі мови програмування Groove. У 2013 Google адаптувала її як, вбудовану систему для Android проєктів. Спроектвана для підтримки суттєвих мульти-проєктних побудов, Gradle також дозволяє додаткові зміни до побудови, ідентифікуючи оновлені частини вашого проєкту. Це допомагає уникати повторного виконання частин, яких не стосувалось оновлення[10].

Порівняння характеристик двох систем зображені в таблиці 2.3.

Таблиця 2.3 – Порівняння систем автоматичного складання

Характеристика	Maven	Gradle
Динамічні та транзитивні залежності	+	+
Використання кешу сторонніх залежностей	+	+
Читання формату метаданих POM	+	+

Продовження таблиці 2.3

Характеристика	Maven	Gradle
Додаткові зміни до побудови	-	+
Підтримка приватних репозиторіїв	+	+
API	+	+, працює швидше
Імплементация залежностей	+	+, працює швидше
Безпечність одночасного використання кешу	+	+, працює швидше
Правила заміни для сумісних бібліотек	-	+
Застосування правил ReplacedBy	-	+
Розширена роздільна здатність метаданих	-	+
Можливість динамічної заміни залежностей проєкту на зовнішні	-	+
Паралельна побудова	+	+

Після аналізу систем автоматичного складання вибором для використання в розробці є Gradle адже він пропонує більше можливостей та функцій ніж Maven.

2.4 Висновок за розділом

В другому розділі було проведено аналіз матеріалів та методів для розробки програмного забезпечення. Проведено огляд і порівняння існуючих мов програмування, програмних засобів та систем складання проєктів для розробки Android додатків.

Для подальшої розробки було обрано:

- IntelliJ IDEA, як середовище розробки;
- Java, як мову програмування;
- Gradle, в якості системи автоматичного складання.

3 ПРОЄКТУВАННЯ ЗАСТОСУНКУ

3.1 Функціональні вимоги до програми та інтерфейсу

Початковий етап проєктування застосунку полягає у визначенні функцій та характеристик, які необхідно мати програмному забезпеченню з вивчення правил дорожнього руху для ефективного та комфортного вивчення:

- теоретичний матеріал – так як основна роль застосунку полягає в наданні можливості вчити матеріал, одним із основних функцій є подача матеріалу;
- сортування матеріалу за темами – оскільки теоретичного матеріалу велика кількість важливо розділити його за темами для комфорту користування;
- пошук – складно запам'ятати абсолютно усі пункти і для легкості використання додатку необхідна можливість пошуку за вмістом параграфу;
- прогрес користувача – для заохочення продовження навчання, для користувача необхідний стимул у вигляді відслідковування прогресу;
- реєстрація – для зберігання прогресу користувача, йому необхідно зареєструватися в додатку;
- авторизація – користувач має необхідність у входженні в додаток, саме тому необхідна авторизація за зареєстрованими даними.

3.2 Розробка користувацького інтерфейсу

Одним із основних компонентів в розробці додатків є графічний інтерфейс. Його ключові функції полягають в наданні користувачу зручного та ефективного досвіду взаємодії з додатком. Інтерфейс повинен мати такі властивості, як:

- інтуїтивність – інтерфейс повинен бути зрозумілим для користувача та легким у використанні;

– компактність – інтерфейс не має бути занадто обширний, аби бути доступний і не заплутувати користувача.

Для програми з вивчення правил дорожнього руху було розроблено 10 інтерфейсів.

Макет головної сторінки це інтерфейс `activity_main`, де користувач може обрати режим роботи з програмою – теорія чи картки, шукати необхідний параграф, перейти на сторінку для входу в аккаунт. Макет зображений на рис. 3.1.

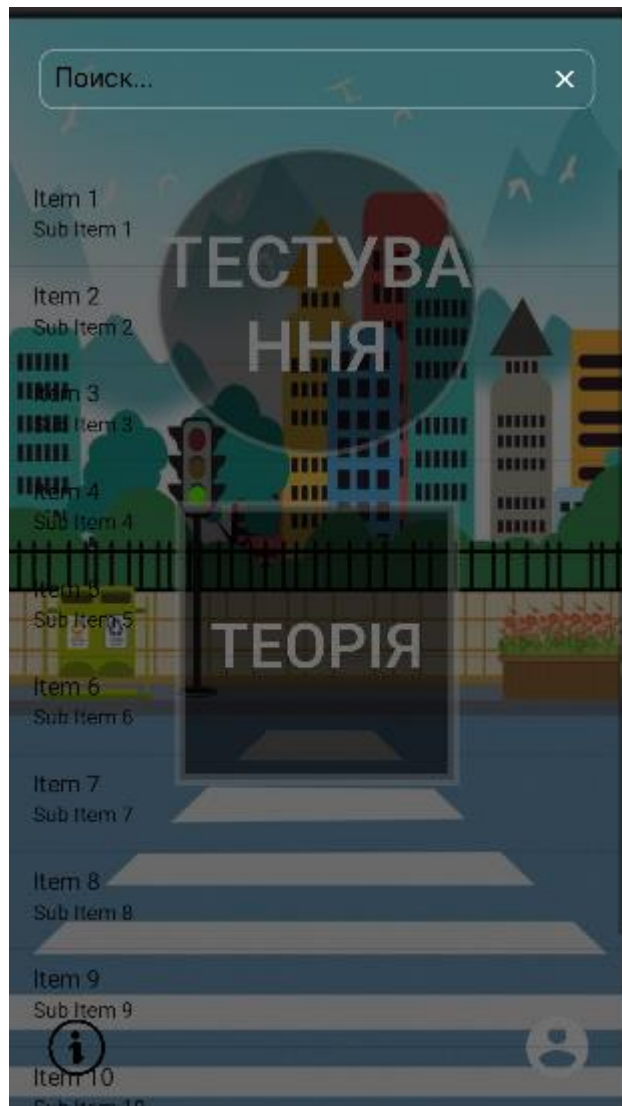


Рисунок 3.1 – Макет головної сторінки

Макет теоретичної сторінки це інтерфейс `activity_theory`, де користувач може обрати режим необхідну йому теорію: Правила дорожнього руху чи дорожні знаки. Макет зображений на рис. 3.2.

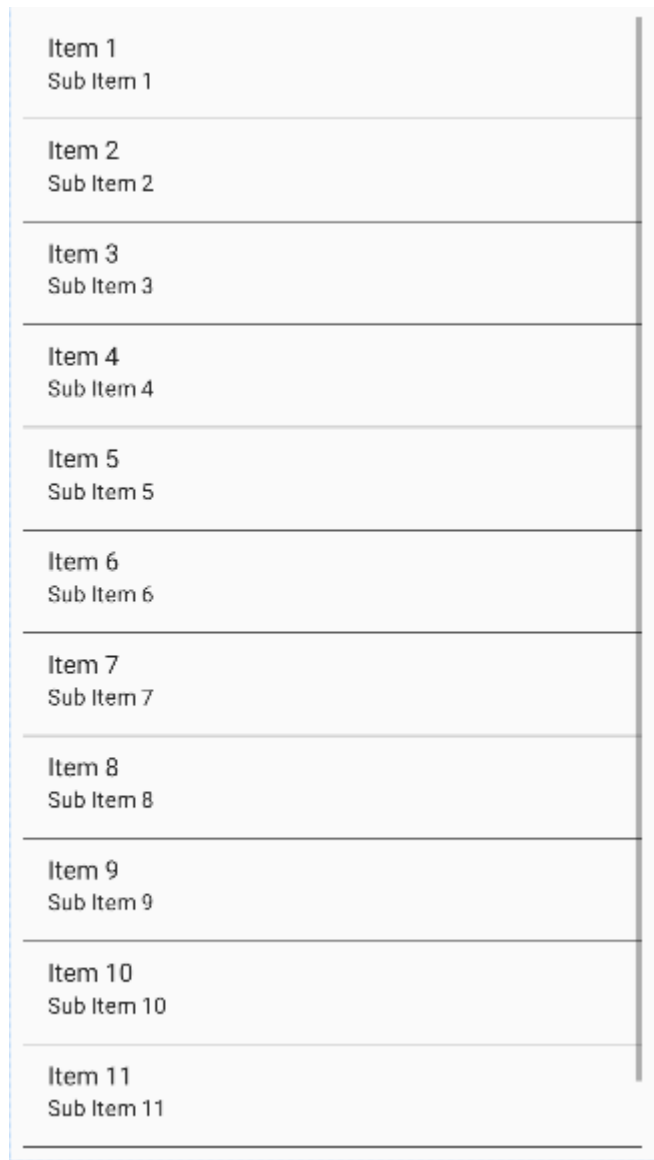


Рисунок 3.2 – Макет теоретичної сторінки

Макет сторінки з розділами це інтерфейс `activity_pdr`, де користувач може обрати розділ для вивчення теорії. Макет зображений на рис. 3.3.

Item 1 Sub Item 1
Item 2 Sub Item 2
Item 3 Sub Item 3
Item 4 Sub Item 4
Item 5 Sub Item 5
Item 6 Sub Item 6
Item 7 Sub Item 7
Item 8 Sub Item 8
Item 9 Sub Item 9
Item 10 Sub Item 10
Item 11 Sub Item 11

Рисунок 3.3 – Макет сторінки з розділами

Макет сторінки з теорією це інтерфейс `activity_information`, де користувач може прочитати та вивчити необхідну теорію. Макет зображений на рис. 3.4.

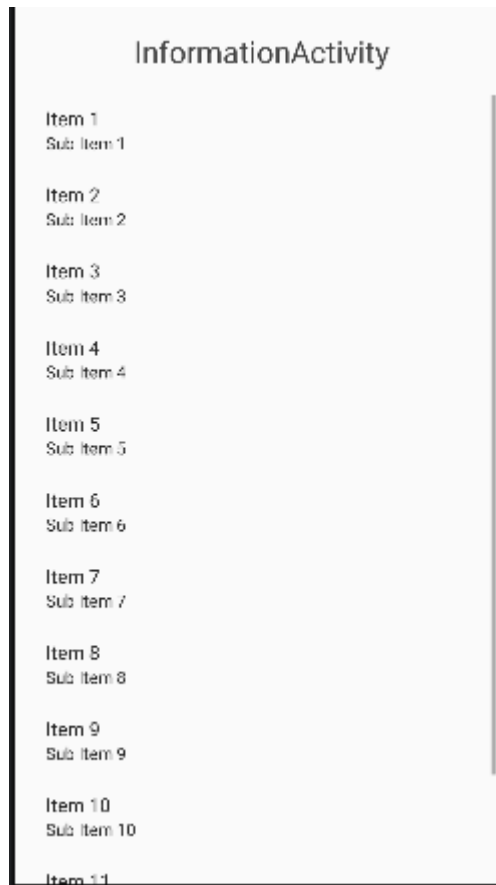


Рисунок 3.4 – Макет сторінки з теорією

Макет сторінки з дорожніми це інтерфейс `activity_Image`, де користувач може прочитати та ознайомитись з дорожніми знаками. Макет зображений на рис. 3.5.

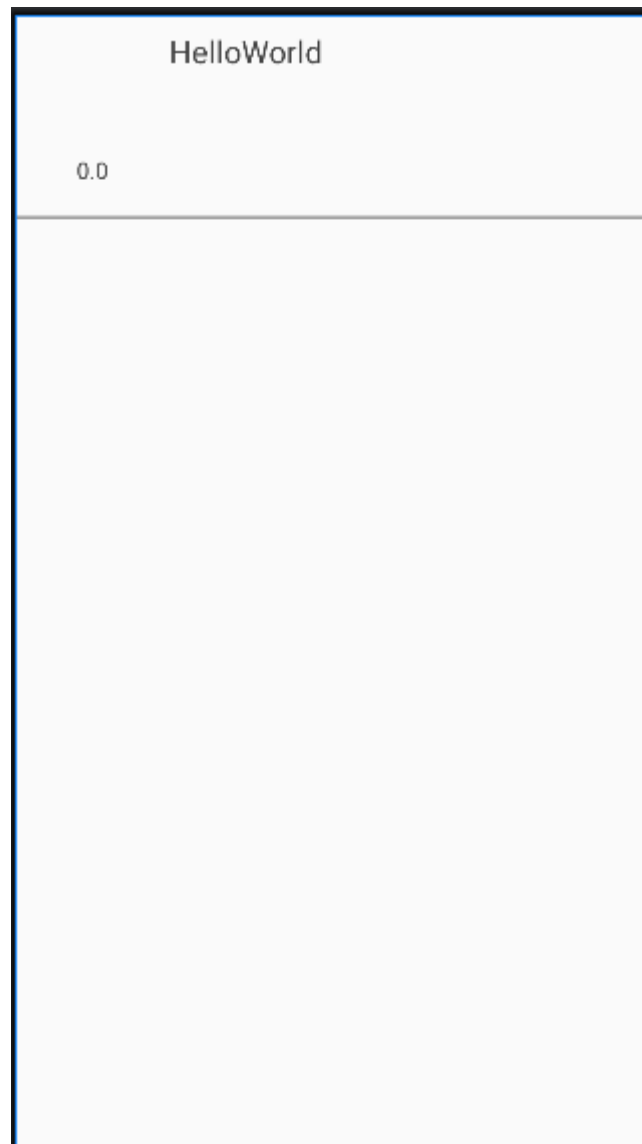


Рисунок 3.5 – Макет головної сторінки

Макет сторінки з флеш-картками це інтерфейс `activity_Cards`, де користувач може скреститися флеш-картками для вивчення дорожніх знаків. Макет зображений на рис. 3.6.

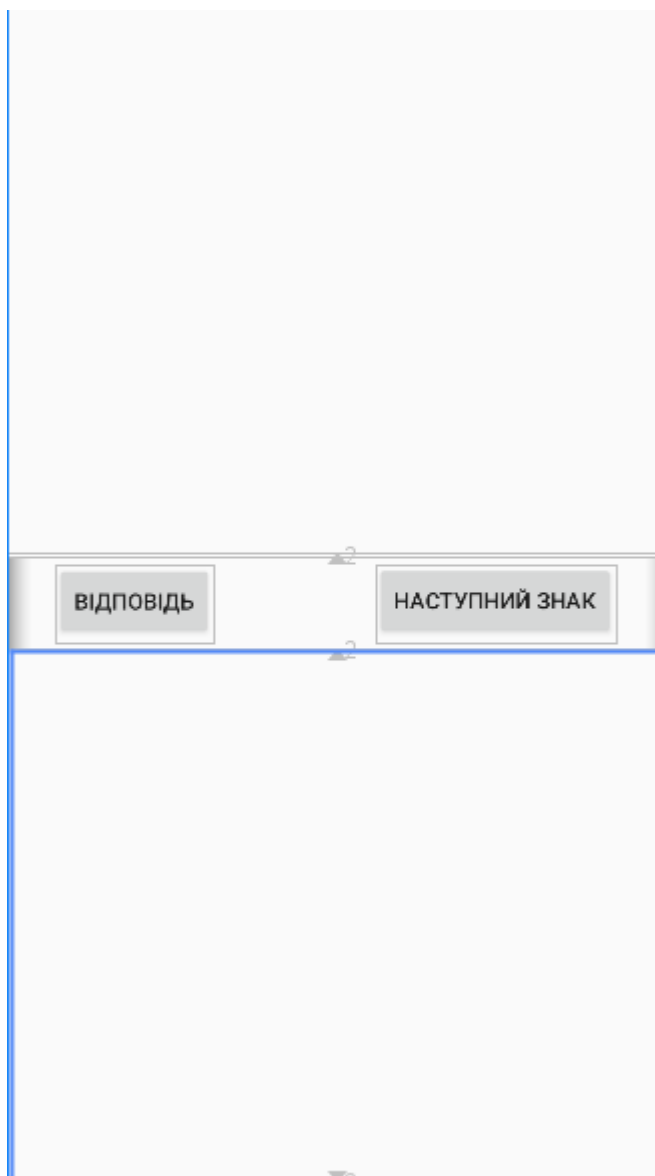
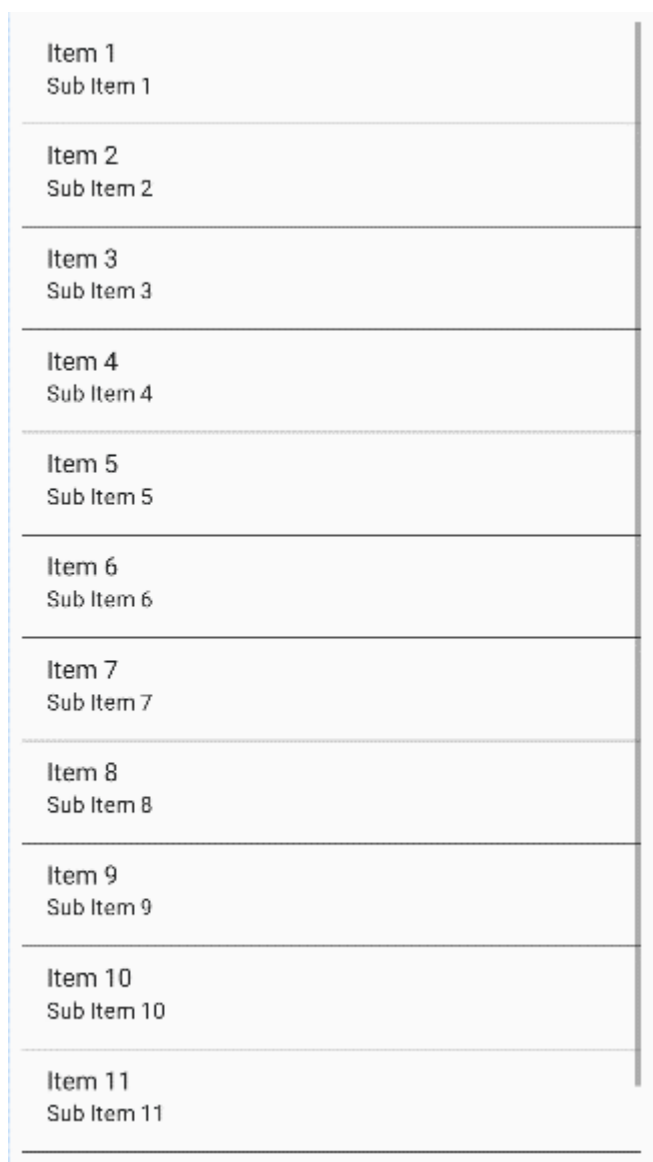


Рисунок 3.6 – Макет сторінки з флеш-картками

Макет сторінки відображення параграфу після пошуку це інтерфейс `activity_article`, де користувач може прочитати параграф який його цікавив. Макет зображений на рис. 3.7.



Item 1 Sub Item 1
Item 2 Sub Item 2
Item 3 Sub Item 3
Item 4 Sub Item 4
Item 5 Sub Item 5
Item 6 Sub Item 6
Item 7 Sub Item 7
Item 8 Sub Item 8
Item 9 Sub Item 9
Item 10 Sub Item 10
Item 11 Sub Item 11

Рисунок 3.7 – Макет сторінки відображення параграфу

Макет сторінки авторизації та реєстрації це інтерфейс `activity_login`, де користувач може авторизуватись в акаунт чи зареєструватись. Макет зображений на рис. 3.8.



Рисунок 3.8 – Макет сторінки авторизації та реєстрації

Макет сторінки профілю це інтерфейс `activity_profile`, де користувач може переглянути свій прогрес. Макет зображений на рис. 3.9.

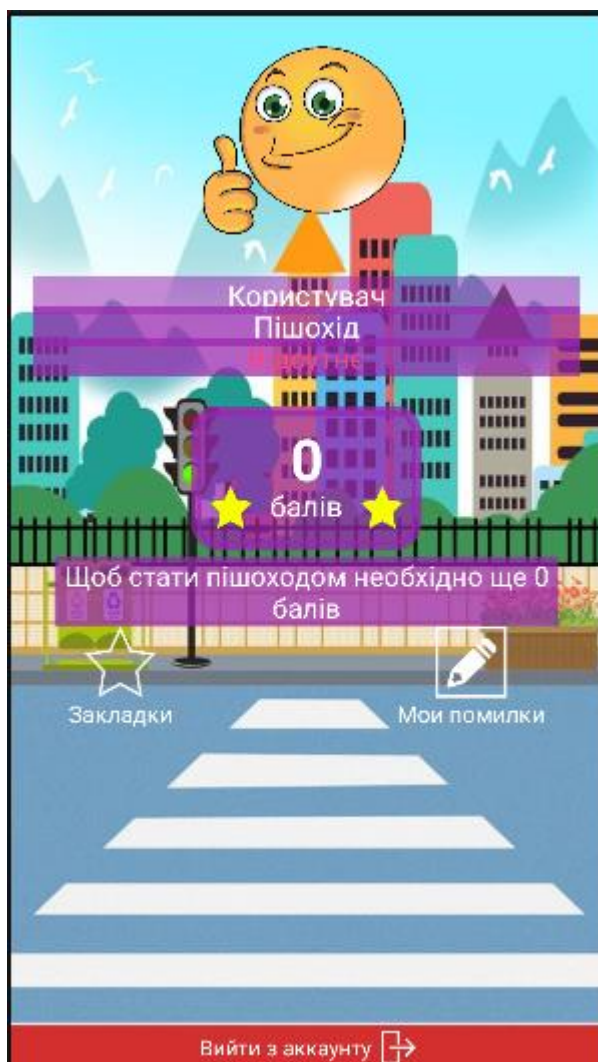


Рисунок 3.9 – Макет сторінки профілю

Макет сторінки інформації про додаток це інтерфейс `activity_info`, де користувач може переглянути свій прогрес. Макет зображений на рис. 3.10.



Рисунок 3.10 – Макет сторінки інформації

3.3 Розробка архітектури застосунку

Спосіб переміщення користувача між різними екранами зображений в діаграмі навігації по інтерфейсу на рис. 3.11.



Рисунок 3.11 – Навігація по інтерфейсу

Функціональна структура програми показує розподіл функцій системи на дрібніші підфункції. Це важливий крок у проектуванні, оскільки допомагає розробникам краще зрозуміти структуру програми та взаємозв'язок між різними її частинами. Функціональна структура зображена на рис. 3.12.

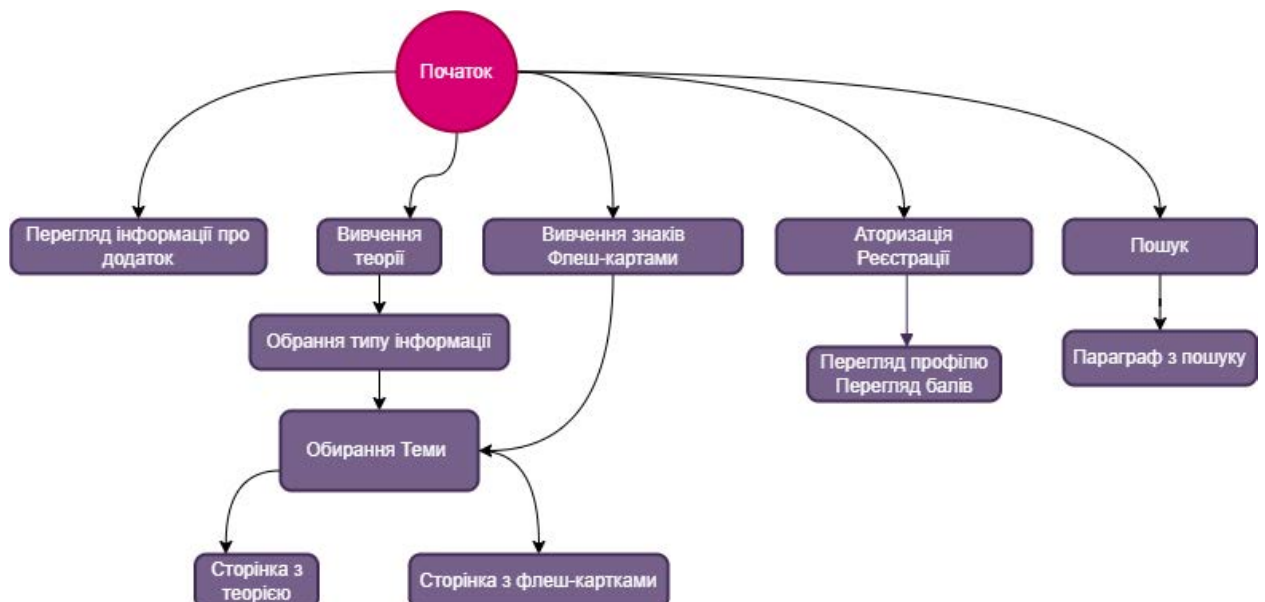


Рисунок 3.12 – Функціональна структура програми

Прецеденти визначають очікувану поведінку та відповідають на питання, що робить система. Прецедент вказує, що має трапитись, проте не

відповідає на запитання, як це має статись[14]. Прецедент застосунку для вивчення ПДР зображений на рис. 3.13.



Рисунок 3.13 – Прецедент застосунку

3.4 Висновок за розділом 3

В третьому розділі були представлені функціональні вимоги до додатку, розроблений користувацький інтерфейс, зокрема макети застосунку, а також описана навігація та функціональна структура програми, діаграма та прецедентів.

4 РОЗРОБКА ЗАСТОСУНКУ

4.1 Структурна схема програми

Структурна схеми програми – це важлива складова для розуміння вже існуючого коду. Для розробників важливо дотримуватися базових принципів організації коду, який полягає в тому аби розбити свою програму мінімальні блоки та розділити їх по відповідним директоріям.

Ці принципи є ключовими в підтриманні організації коду, оновлені модулів та архітектури програми.

Фізична модель додатку для вивчення правил дорожнього руху відображає набір таких файлів як:

- класи;
- ресурси ;
- маніфести.

Фізична модель додатку для вивчення ПДР представлено на рис. 4.1.

Загальна структура додатку для вивчення правил дорожнього руху відображає набір таких класів та файлів макетів як:

- інтерфейси;
- адаптери;
- активності;
- утиліти.

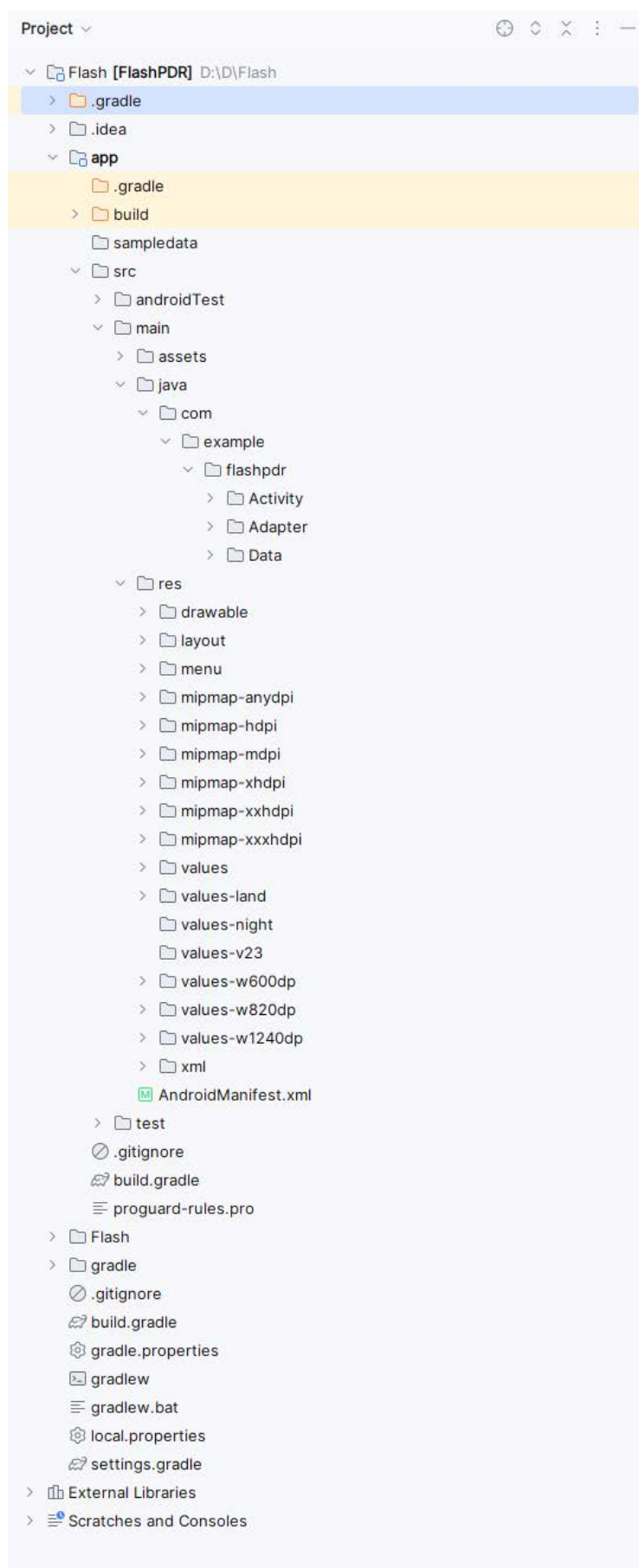


Рисунок 4.1 – Фізична структура застосунку

Загальна структурна модель застосунку для вивчення ПДР представлено на рис. 4.2

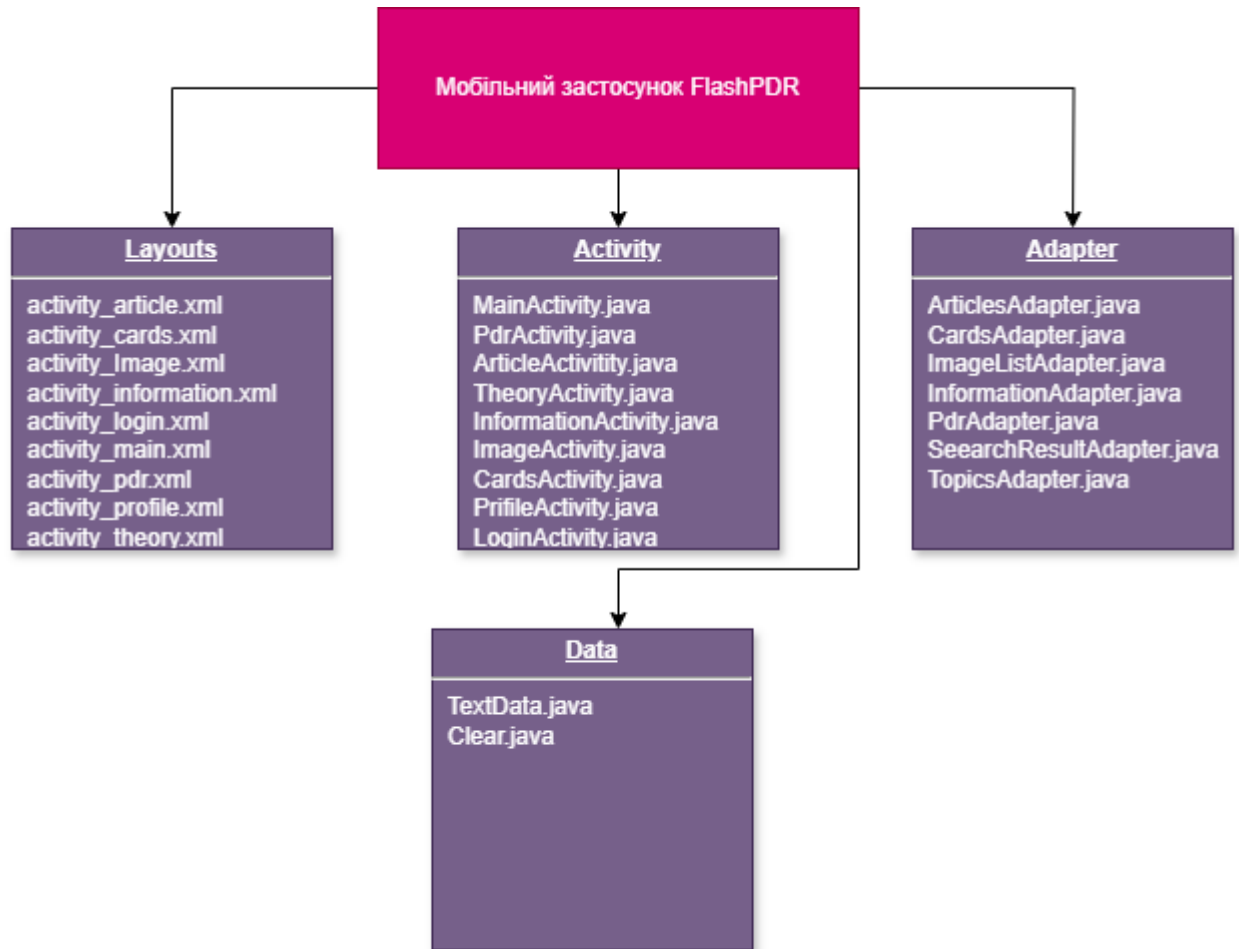


Рисунок 4.2 – Загальна структура додатку

Спираючись на рисунки 4.1 та 4.2 можна зрозуміти фізичну структуру, як спосіб організації коду та принцип його зберігання, в той час як загальна структура представляє логічну структуру програмного коду.

4.2 Файли, методи та функції

4.2.1 Модуль Activity

В цьому Розділі розглянемо імплементації активностей які виконуються в застосунку.

Файл MainActivity.java це головна активність застосунку. Яка запускає головний екран з вибором дії: теорія, флеш карти або авторизація. В таблиці 4.1 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.1 – Призначення методів та змінних файлу MainActivity.java

Найменування	Опис
checkForLogin()	Перевірка входу
setLayout(int layout)	Функція зміни інтерфейсу
onCardClick()	Запуск активності CardsActivity
onTheoryClick()	Запуск активності TheoryActivity
startLoginActivity()	Запуск активності LoginAvtivity
showMore()	Запуск активності InformationActivity
findWord(String s)	Пошук за словом
qsort(List<Pair<JSONObject, Integer>> entries, int l, int r)	Сортування
displayInfo(List<Pair<JSONObject, Integer>> entries)	Вивід інформації з файлу

Файл CardsActivity.java це активність, яка запускає режим флеш-карток. Де показується дорожній знак, і користувач має його назвати та перевірити свою відповідь та перейти до наступного зображення. В таблиці 4.2 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.2 – Призначення методів та змінних файлу CardsActivity.java

Найменування	Опис
Dictionary FlashCard	Зміна для поточних даних
Dictionary getFlashcard(int position)	Функція задання поточних даних
void displayInfo(List<JSONObject> data, List<Boolean> isDivider)	Вивід поточних даних
void getJSONData(JSONArray articles)	Отримання даних з файлу

Файл TheoryActivity.java це активність, яка запускає режим вибору типу теорії для вивчення. Є вибір обрати дорожні знаки чи ПДР. В таблиці 4.3 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.3 – Призначення методів та змінних файлу TheoryActivity.java

Найменування	Опис
void runActivity(Class activity)	Запуск активності PdrActivity
void setListViewAdapter()	Налаштування вибору режиму

Файл PdrActivity.java це активність, яка запускає режим вибору теми з теорії чи знаків дорожнього руху та перехід до самої теорії. В таблиці 4.4 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.4 – Призначення методів та змінних файлу PdrActivity.java

Найменування	Опис
String TAG = "PdrActivity";	Маркер для позначення, яку інформації підгружати
void onTitleClick(int i)	Запуск активності InformationActivity
displayInfo()	Вивід тем

Файл `InformationActivity.java` це активність, яка запускає режим вибору типу теорії для вивчення. Є вибір обрати дорожні знаки чи ПДР. В таблиці 4.5 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.5 – Призначення методів та змінних файлу `InformationActivity.java`

Найменування	Опис
<code>void getJSONData(JSONArray articles)</code>	Отримання теорії з файлу
<code>void displayInfo(List<JSONObject> data, List<Boolean> isDivider)</code>	Вивід теорії

Файл `LoginActivity.java` це активність, яка запускає входу в свій аккаунт або реєстрації. В таблиці 4.6 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.6 – Призначення методів та змінних файлу `LoginActivity.java`

Найменування	Опис
<code>void onRegistrationClick()</code>	Реєстрація нового користувача
<code>void onLoginClick()</code>	Вхід в аккаунт
<code>void startProfileActivity()</code>	Запуск активності <code>ProfileActivity</code>
<code>void setLayout(int layout)</code>	Обирання режиму інтерфейсу

Файл `ProfileActivity.java` це активність, яка запускає Твій акаунт. В таблиці 4.7 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.7 – Призначення методів та змінних файлу `ProfileActivity.java`

Найменування	Опис
<code>private void onLogoutClick()</code>	Реєстрація нового користувача

Файл `ArticleActivity.java` це активність, яка запускає відображення одного пункту теорії. В таблиці 4.8 наведено опис призначення методів та змінних програмного коду.

Таблиця 4.8 – Призначення методів та змінних файлу `ArticleActivity.java`

Найменування	Опис
<code>void displayInfo()</code>	Відображення одного пункту теорії

Отже наскільки зрозуміло з імплементації класів модуля `Activity` розширює можливості бібліотеки “`AppCompatActivity`” та кожен клас відповідає за окрему активність.

4.2.2 Модуль `Adapter`

В цьому Розділі розглянемо імплементації адаптерів які виконуються в застосунку. Оскільки дані розміщені у файлі а зображення зберігаються в ресурсах для коректного відображення застосовуються власні адаптери. Усі адаптери мають одні методи та змінні, які наведені в таблиці 4.9.

Таблиця 4.9 – Призначення методів та змінних файлу адаптерів

Найменування	Опис
<code>List<JSONObject> paragraphs</code>	Зберігання даних з файлу
<code>List<Boolean> isDivider</code>	Додання розділювача
<code>String fromActivity;</code>	Інформація з якої активності
<code>int getCount()</code>	Кількість даних
<code>Object getItem(int i)</code>	Отримання даних
<code>long getItemId(int i)</code>	Отримання номеру даних

Продовження таблиці 4.9

Найменування	Опис
View getView(int i, View view, ViewGroup viewGroup)	Вивід даних у вигляді ListView

Файл ArticleAdapter.java це адаптер що описує відображення одного пункту теорії з пошуку.

Файл CardsAdapte.java це адаптер що описує відображення карток зі знаками.

Файл ImageListAdapter.java це адаптер що описує відображення дорожніх знаків у ListView та підписи до них.

Файл InformationAdapter.java це адаптер що описує відображення теорії.

Файл PdrAdapter.java це адаптер що описує відображення розділів теорії.

Файл SearchAdapter.java це адаптер що описує відображення результатів пошуку.

Файл TopicsAdapter.java це адаптер що відображує який розділ теорії вас цікавить.

Отже наскільки зрозуміло з імплементації класів модуля Adapter вони розширюють можливості віджету “BaseAdapter” та кожен клас відповідає за коректне відображення даних.

4.2.3 Модуль Data

В цьому розділі розглянемо імплементації збереження даних.

Файл TextData.java це клас що зберігає усі необхідні дані для роботи додатку. В таблиці 4.10 наведено опис методів та змінних програмного коду.

Таблиця 4.10 – Призначення методів та змінних файлу TextData.java

Найменування	Опис
private static TextData textData;	Зберігання об'єкту за Singleton Design Pattern
private static Context context;	Глобальної інформації про додаток
void setContext(Context contextValue)	Оголошення даних, щоб не допустити Memory Leak
TextData get TextData()	Гетер для TextData
boolean isLoggedIn;	Перевра входу
List<JSONObject> pdr;	Зберігає об'єкти з файлу pdr
List<JSONObject> signs;	Зберігає об'єкти з файлу trafficSign
TextData()	Конструктор класу
void getData(String fileName)	Вигрузка даних з файлу
void chooser(JSONArray data, String fileName)	Обозначення в який List додавати данні

Отже клас TextData.java імплементує Singleton Design Pattern – цей патерн запевняє, що лише один об'єкт такого типу існує і надає одну точку входу в нього для всього коду.

Файл Clear.java це клас що очищує кеш додатку.

4.2.4 Модуль Layout

В цьому розділі розглянемо імплементатії інтерфейсу. Інтерфейс було імплементовано у форматі .xml файлів, В таблиці 4.11 наведено опис інтерфейсів та її призначення.

Таблиця 4.11 – Опис файлів інтерфейсу

Найменування	Опис
activity_article.xml	Інтерфейс для пункту теорії з пошуку
activity_cards.xml	Інтерфейс для відображення карток
activity_Image.xml	Інтерфейс розділів зі знаками
activity_information.xml	Інтерфейс подачі теорії
activity_login.xml	Інтерфейс входження в аккаунт
activity_main.xml	Інтерфейс головного меню
activity_pdr.xml	Інтерфейс вибору типу теорії
activity_profile.xml	Інтерфейс профілю
activity_theory.xml	Інтерфейс розділів з теорією

4.3 Висновок за розділом

В четвертому розділі наведено детальний опис розроблених модулів застосунку.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

В даному розділі розглянуто призначення програми, умови і процес її виконання та тестування.

5.1 Призначення та умови застосування програми

Застосунок «FlashPdr» призначений для ефективного та доступного вивчення Правил Дорожнього Руху. Для виконання поставлених функцій і вимог було спроектовано:

- функцію пошуку;
- зберігання поточного результату;
- навчання завдяки флеш карткам;
- робота застосунку за відсутності інтернету.

Для коректної роботи, застосунку необхідно дотриматися наступних апаратних вимог:

- операційна система пристрою – android;
- мінімальна версія операційної системи android – 8.0 (android oreo);
- максимальна версія операційної системи android – 14.0 (android upside down cake);
- 10 мб вільного місця на пам'яті.

5.2 Опис роботи програми

Для запуску програми необхідно скачати файл з розширенням .apk на смартфон, що задовольняє апаратні вимоги та встановити його. Процес встановлення зображено на рис. 5.1.

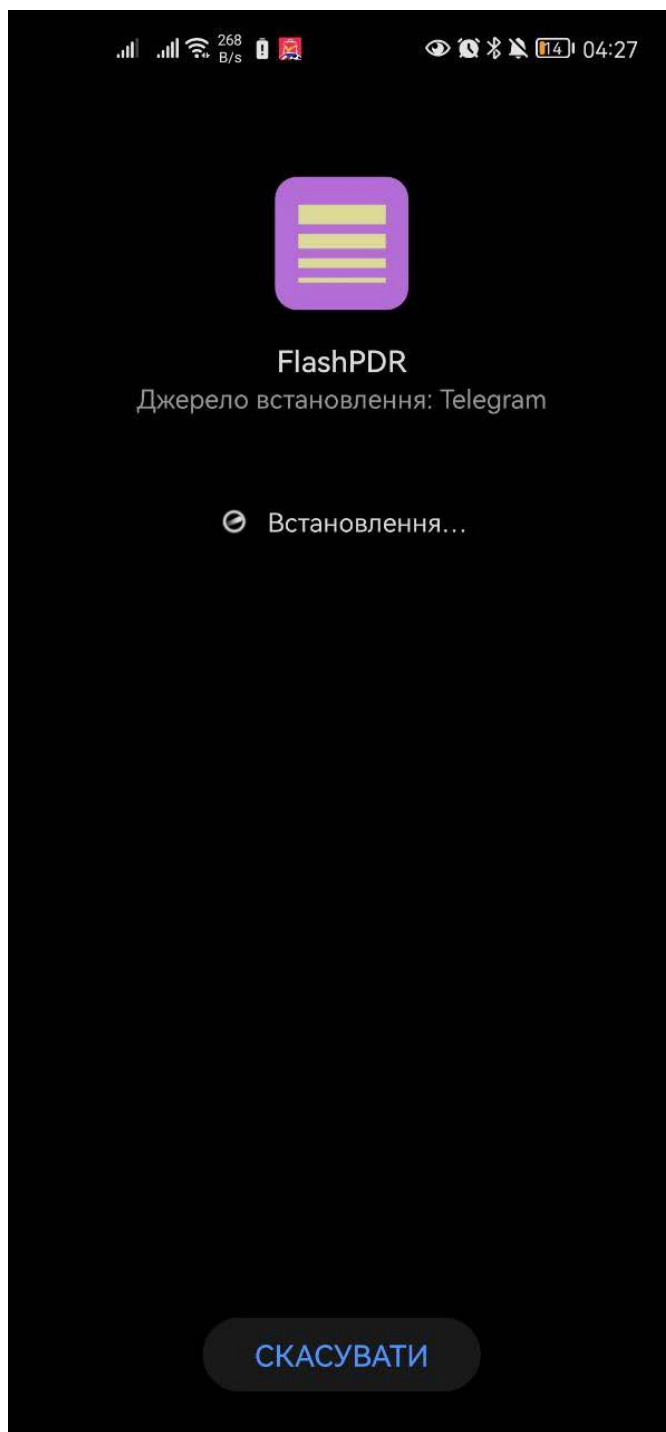


Рисунок 5.1 – Процес установки додатку

Після запуску програми ми потрапляємо на головну сторінку, яку зображено на рис. 5.2. де можемо обрати режими роботи які нас цікавлять.



а) кнопка «Теорія»;

б) кнопка «Тестування»;

в) кнопка «Профіль»;

г) кнопка «Інформація»

д) поле «Пошук»

Рисунок 5.2 – Головний екран додатку

При натисканні кнопки теорія, користувач потрапляє в меню вибору типу теорії, що зображено на рис. 5.3

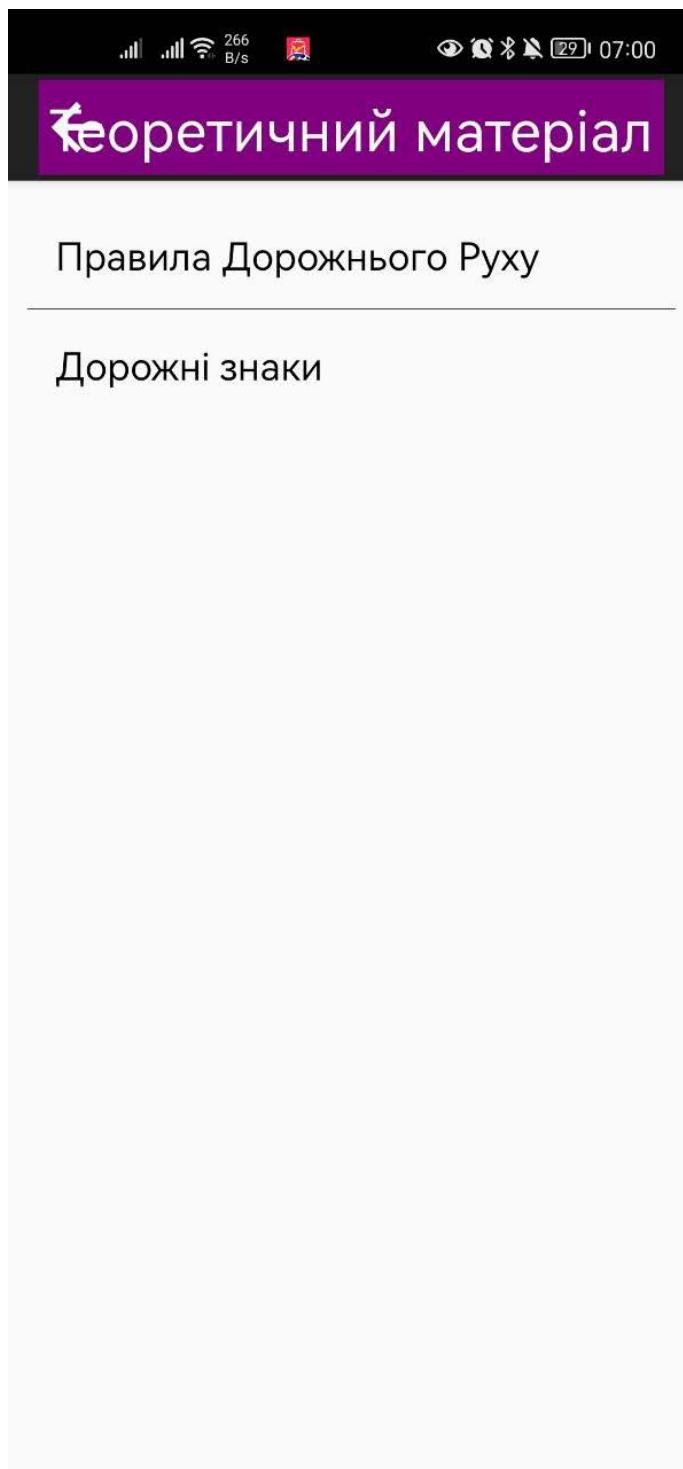


Рисунок 5.3 – Меню вибору типу теорій

При натисканні типу «Правила Дорожнього Руху» відкривається вікно з назвами розділів теоретичного матеріалу, що зображено на рис. 5.4.

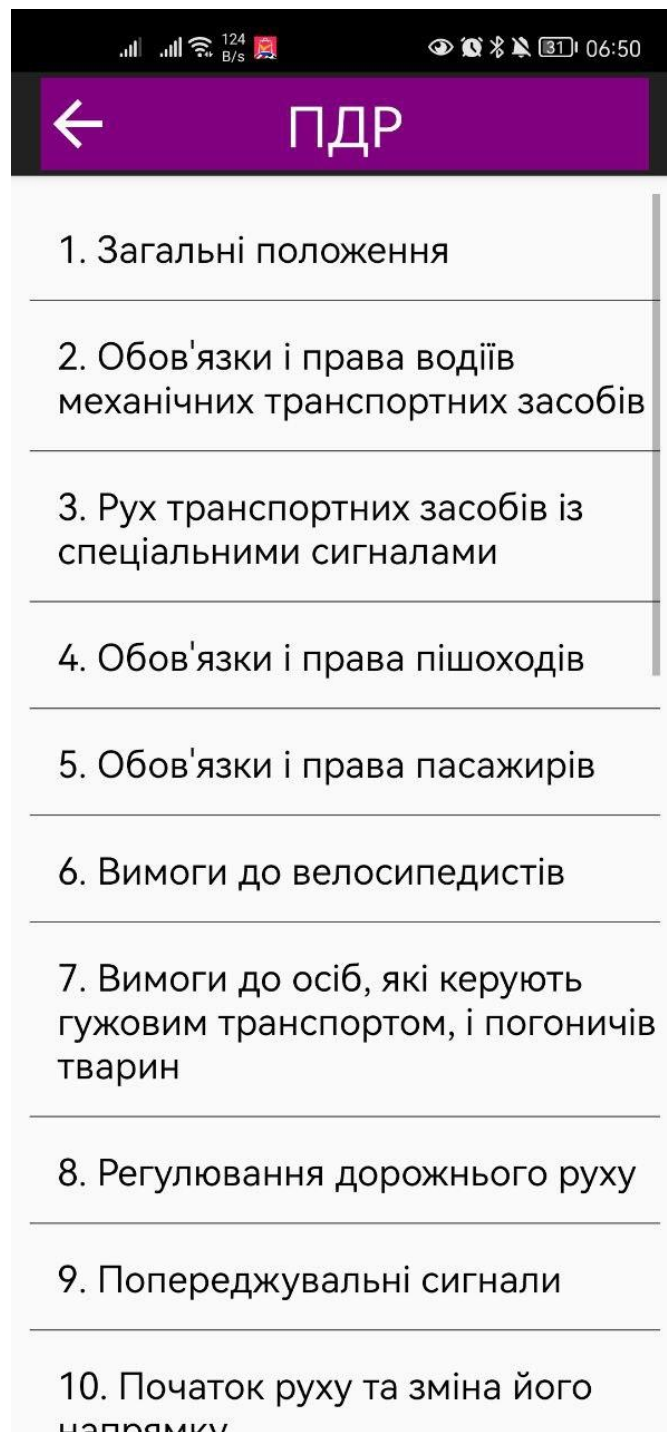


Рисунок 5.4 – Меню назв розділів теорії

Після вибору розділу відкривається теорія саме з цього розділу, зображено на рис. 5.5.

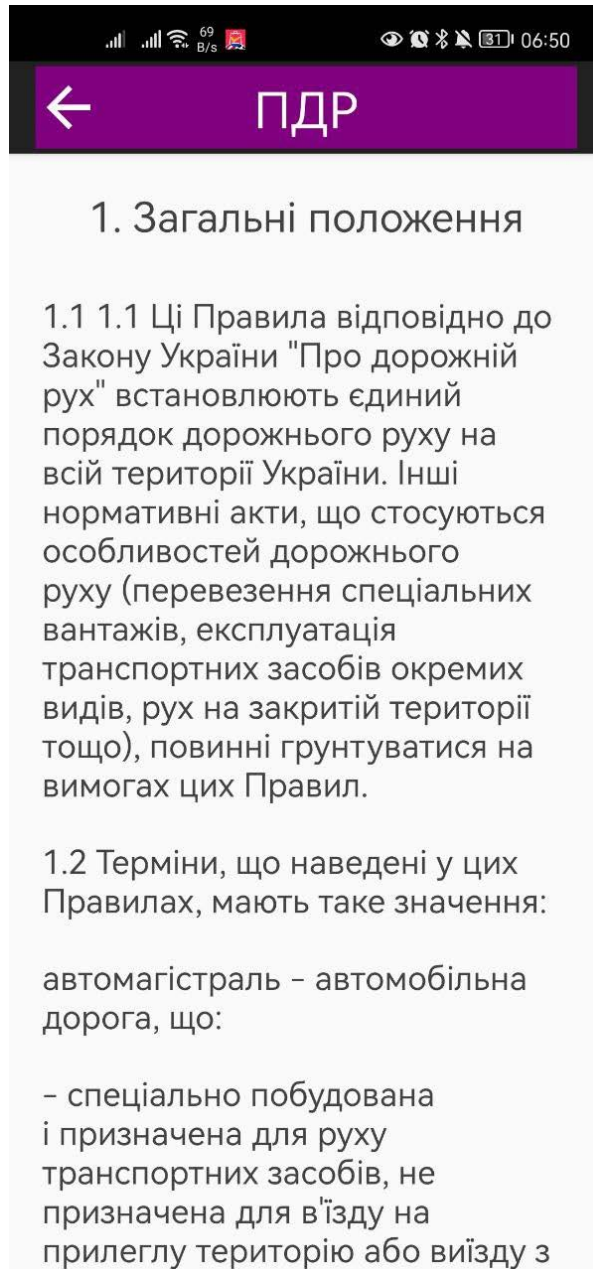


Рисунок 5.5 – Теорія 1 розділу

При обираї в меню вибору типу теорії обравши тип «Дорожні знаки», відкривається вікно з назвами розділів дорожніх знаків ілюстровано на рис. 5.6.



Рисунок 5.6 – Меню назв розділів дорожніх знаків

Після вибору розділу відкривається теорія саме з цього розділу, зображено на рис. 5.7.



Рисунок 5.7 – Теорія 1 розділу Дорожніх знаків

Якщо в головному меню натиснути кнопку тестування, відкривається вікно вибору розділів (рис. 5.6). Після вибору розділу відкривається вікно з картками для тестування (рис. 5.8):

- де можна або дізнатись відповідь;
- або перейти на наступний знак.



Рисунок 5.8 – Вікно карток для тестування

Приклад виведення відповіді зображено на рис 5.9.



Рисунок 5.9 – Відповідь до картки тестування

Приклад виведення наступного зображення зображено на рис 5.10.



Рисунок 5.10 – Наступна картка

Щоб знайти параграф за текстом в ньому необхідно:

- а) скористуватися полем пошуку;
- б) вести саме те ключове слово, що ви шукаєте (рис. 5.11) ;
- в) натиснути на необхідний для вас параграф після чого відкриється вікно з одним необхідним параграфом.

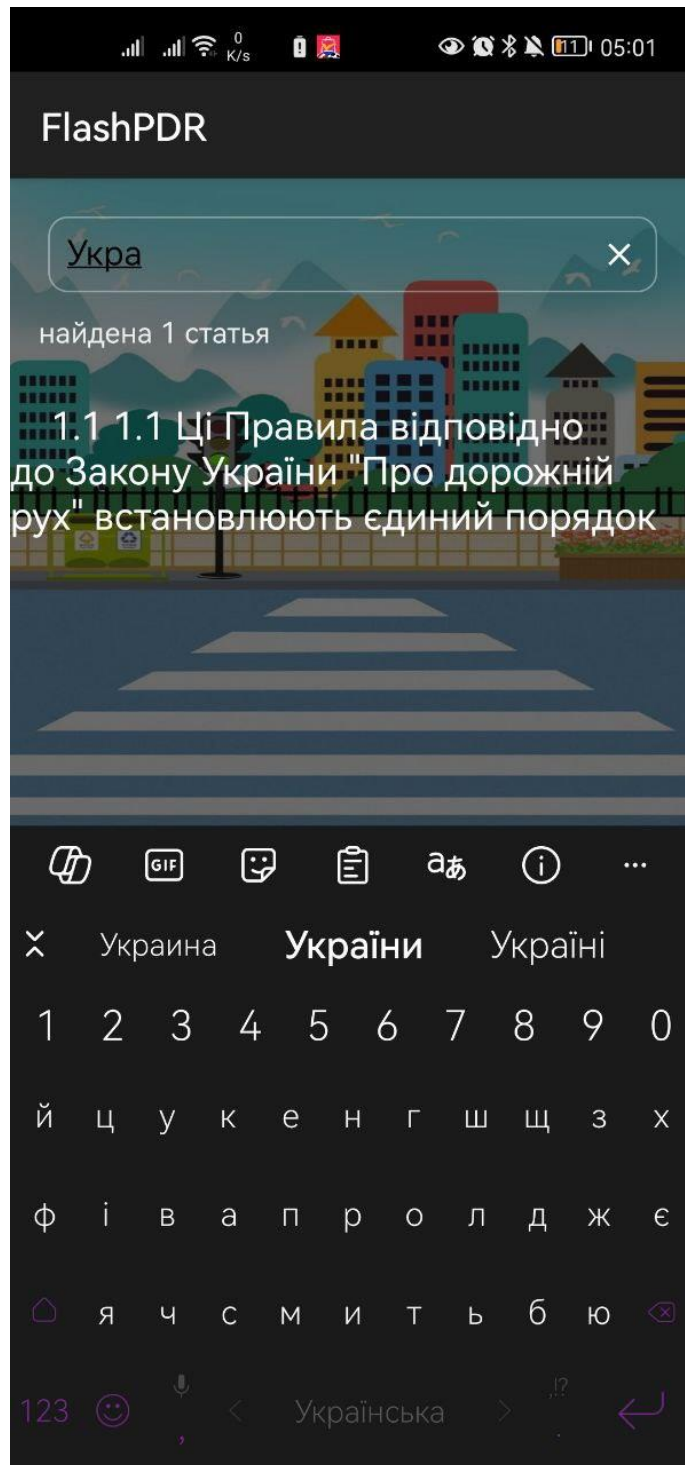


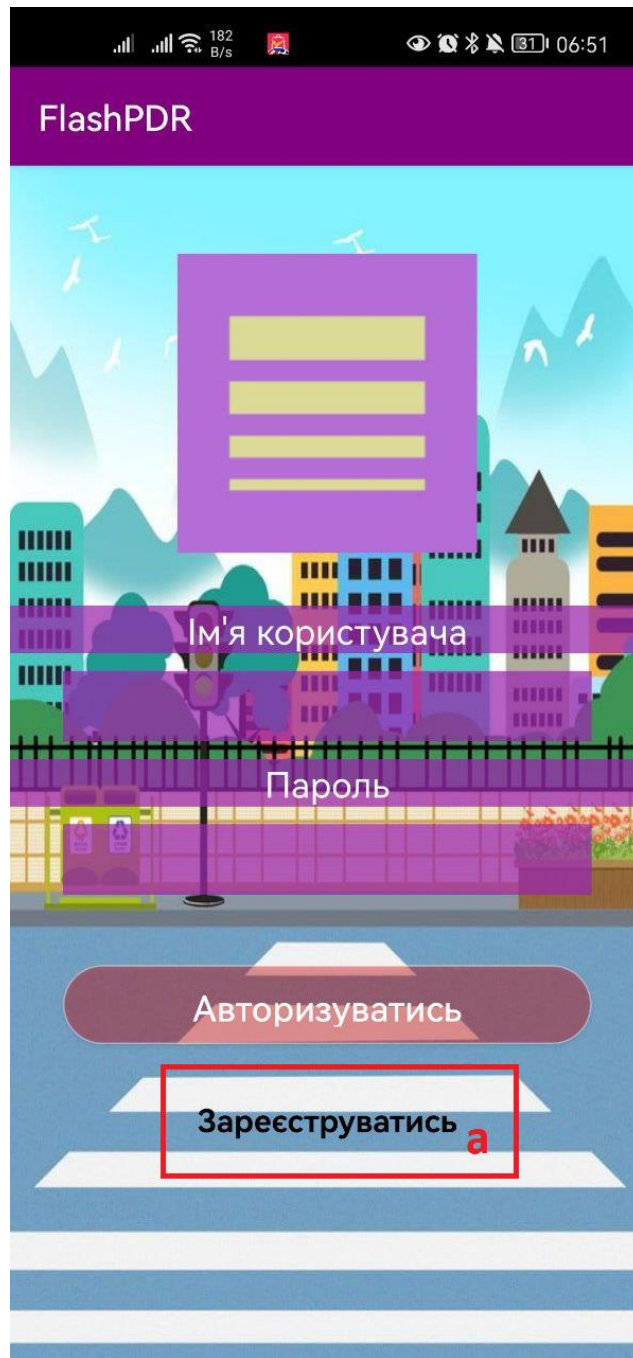
Рисунок 5.11 – Пошук по слову

Приклад виведення необхідного параграфу після пошуку зображено на рис 5.12.



Рисунок 5.12 –Знайдений параграф

Щоб увійти або зареєструвати акаунт необхідно натиснути кнопку профілю, далі відкривається форма авторизації (рис.5.13), якщо ж користувач не зареєстрований необхідно натиснути зареєструватись (рис.5.13а).



а) кнопка «Зареєструватись»

Рисунок 5.13 –Форма авторизації

В оновленому інтерфейсі (рис 5.14) необхідно вести свої дані та натиснути кнопку реєстрація. Після чого ви зможете авторизуватися в додатку.

FlashPDR

← Реєстрація

Ім'я

Почта

Пароль

Повторіть пароль

Реєстрація

Рисунок 5.14 – Форма реєстрації

Після авторизації користувач потрапляє у вікно профілю (рис.5.15) де зображені його бали та рівень прогресу.

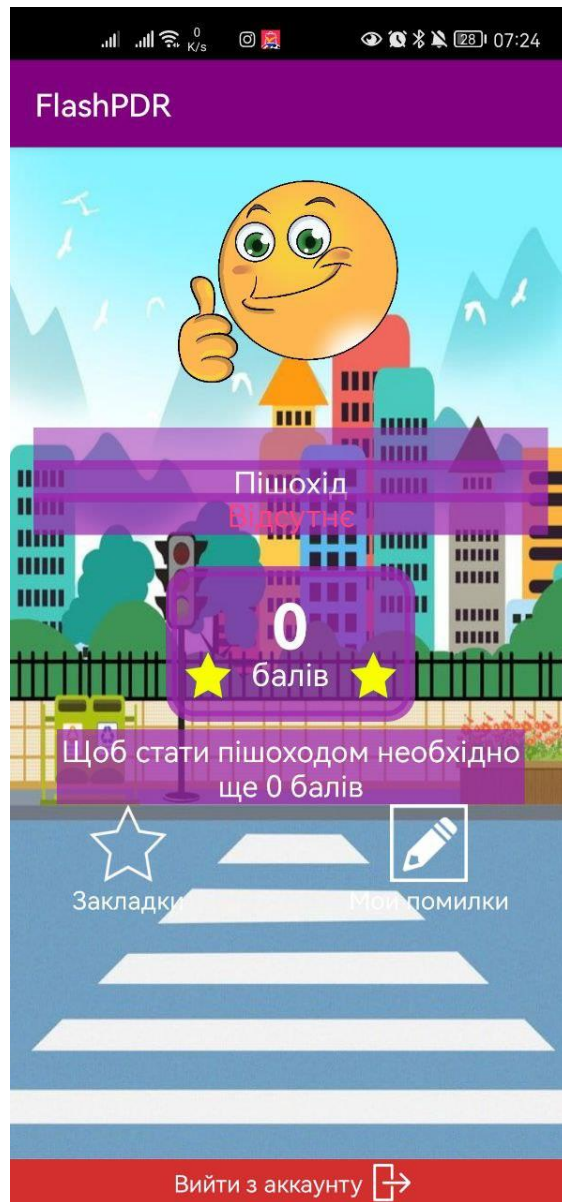


Рисунок 5.15 – Форма профілю

Щоб переглянути інформацію про додаток необхідно натиснути кнопку інформація в головному меню, після чого відкриється вікно з інформацією про додаток та розробника(рис. 5.16)

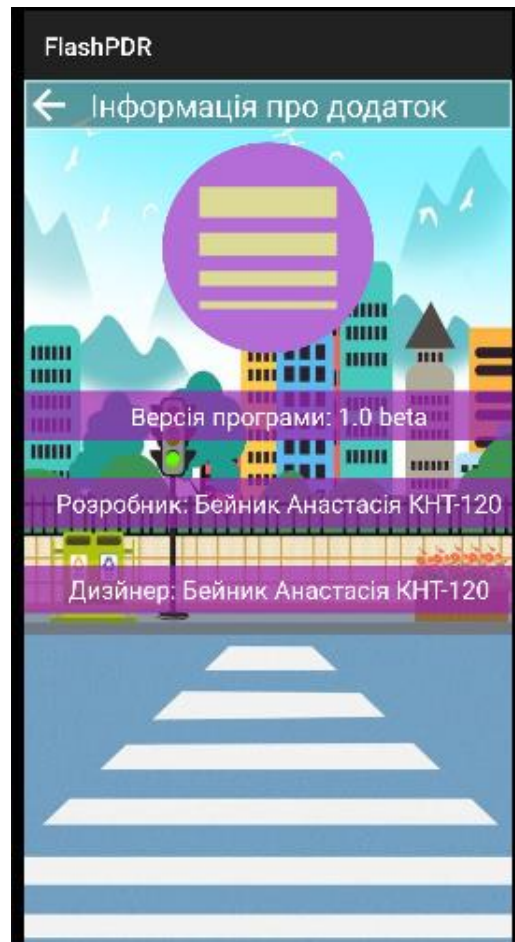


Рисунок 5.16 – Форма інформації

5.3 Тестування програми

Невід’ємною частиною процесу розробки є тестування. Воно необхідне аби в точності перевірити правильність роботи застосунку та реалізацій дій.

Перше тестування відносилось до перевірки коректності реєстрації, за умови введення валідних даних результат зображено на рис. 5.17, реєстрація вже існуючого користувача (рис 5.18).

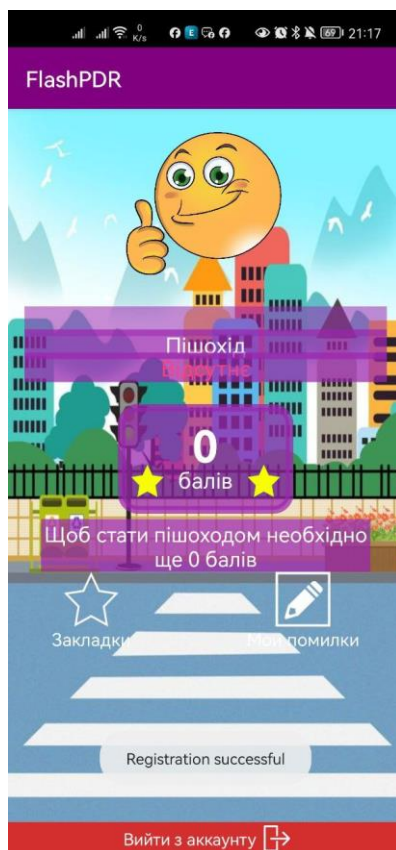


Рисунок 5.17 –Введенні валідні данні

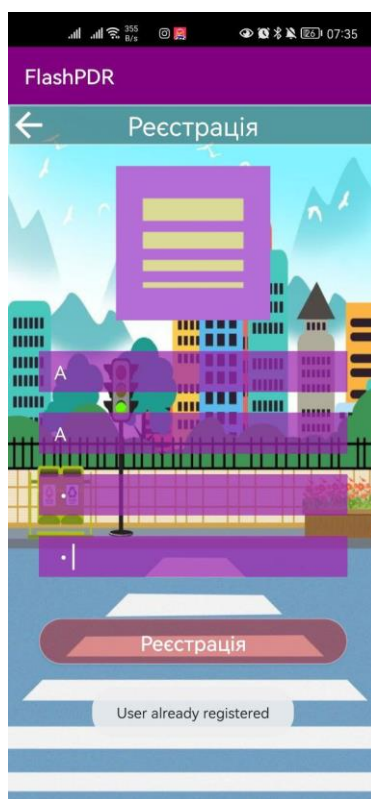


Рисунок 5.18 –Реєстрація вже існуючого користувача

На рис. 5.19 зображено тестування реєстрації при введенні не однакових паролів.



Рисунок 5.19 –Введенні не однакові паролі

Друге тестування полягало в перевірці коректної авторизації, для перевірки використовувались валідні (рис. 5.14) дані та невалідні (рис.5.20)

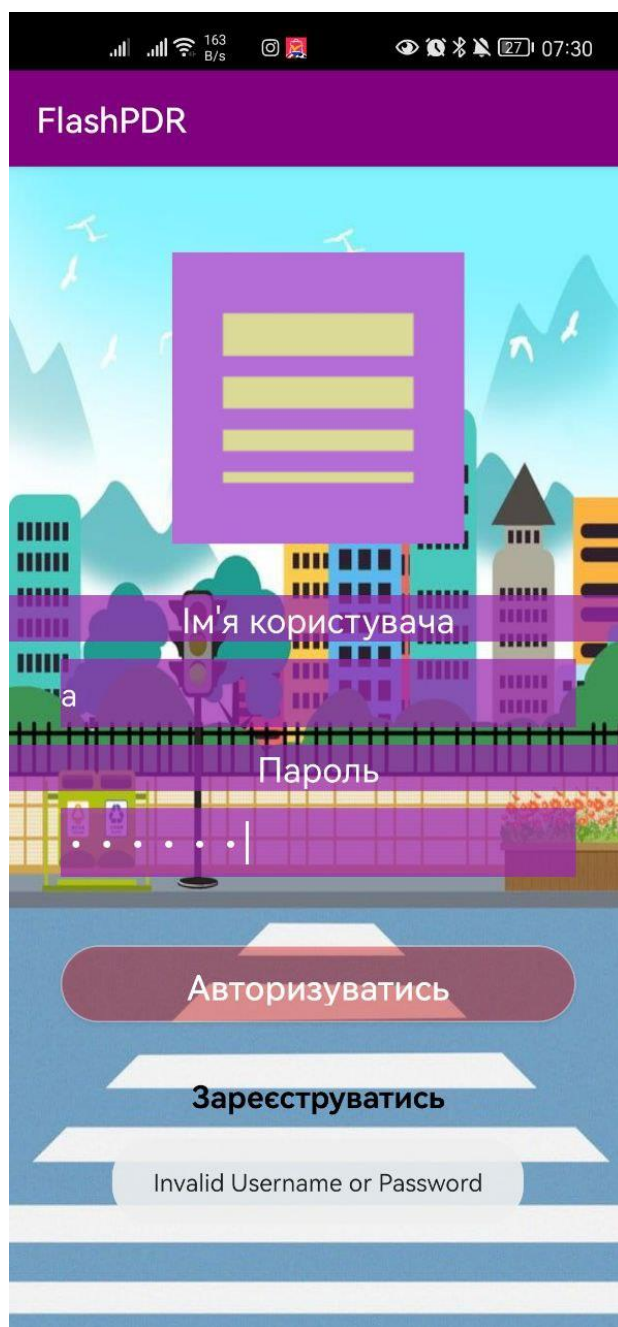


Рисунок 5.20 –Введенні невалідних даних

Наступна перевірка полягала в правильному відображенню результатів пошуку з існуючими даними (рис.5.11) та невалідними (рис.5.21).



Рисунок 5.21 –Пошук невалідних даних

Отже додаток успішно пройшов усі необхідні тести та підтвердив свою працездатність.

5.4 Висновки до 5 розділу

Після розглянутих пунктів в цьому розділі, програмний продукт підтвердив свою працездатність, відповідність потребам та виконання усіх необхідних попередньо поставлених завдань.

Тестування надало впевненість в коректному функціонуванні.

ВИСНОВКИ

Під час виконання бакалаврської роботи було отримано, як необхідний досвід для подальшого розвитку в сфері ІТ, так і розроблено якісний продукт – мобільний застосунок для вивчення ПДР з імплементацією методу ефективного навчання.

Для досягнення мети роботи було проведено дослідження таких факторів:

- аналіз актуальності теми вивчення ПДР;
- дослідження ефективних методів навчання;
- аналіз існуючих додатків та визначення недоліків для виправлення.

Розроблено архітектуру та функціональні можливості мобільного застосунку.

Програмна реалізація була проведена мовою програмування Java з використанням файлів формату .xml для опису інтерфейсів програми. Для складання додатку була використана система Gradle, що дозволить в подальшому додавати нові модулі і функції без перескладання застосунку.

Дані та імплементації функцій розташовані окремо у власному класі, що також дає простір для подальшої актуалізації програми.

Головна вимога в легкості використанні та мобільності була реалізована, додаток неперевантажений даними і містить лише базові найнеобхідніші пункти. Повноцінний доступ до системи має лише користувач застосунку.

Подальше розширення додатку може включати в себе додавання режиму тестів та функції створення власних флеш карток до теорії.

Досвід отриманий під час виконання дипломної роботи, показав, що розробка інтуїтивного та ефективного додатку для навчання є актуальним напрямком руху, можливістю подальшого розширення функціоналу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Трач, О. Вивчати правила дорожнього руху тепер можна самостійно [Електрон. ресурс]/ О. Трач – Режим доступу: <https://zz.te.ua/vyvchaty-pravyla-dorozhn-oho-rukhu-teper-mozhna-samostiyno/>
2. Zosym, M. Навчання (Learning) [Електрон. ресурс]/ M.Zosym – Режим доступу: <https://www.maxzosim.com/learning/> .
3. Living in a multi-device world with Android [Electronic resource]. – Access mode: <https://blog.google/products/android/io22-multideviceworld/> .
4. What is Java? [Electronic resource]. – Access mode: https://aws.amazon.com/what-is/java/?nc1=h_ls .
5. Kotlin [Electronic resource]. – Access mode: <https://www.techtarget.com/whatis/definition/Kotlin> .
6. Мова програмування Kotlin [Електрон. ресурс] – Режим доступу: <https://lemon.school/blog/mova-programuvannya-Kotlin> .
7. Meet Android Studio [Electronic resource]. – Access mode: <https://developer.android.com/studio/intro> .
8. Eclipse Java: середовище розробки “під себе” [Електрон. ресурс] – Режим доступу: <https://javarush.com/ua/groups/posts/uk.2359.eclipse-java-seredovijshe-rozrobki-pd-sebe> .
9. IntelliJ IDEA overview [Electronic resource]. – Access mode: <https://www.jetbrains.com/help/idea/discover-intellij-idea.html#next-steps> .
10. Gradle vs. Maven: Performance, Compatibility, Builds, & More [Electronic resource]. – Access mode: <https://stackify.com/gradle-vs-maven/> .
11. Аналіз сучасних досягнень програмної інженерії для створення універсального алгоритму ASCII-ART конвертації [Електрон. ресурс] – Режим доступу: <https://ojs.kntu.net.ua/index.php/visnyk/article/view/237> .
12. Омельчук, Л. Л. Програмування: теорія та практика. Збірник матеріалів за результатами ІТ-проєкту міждисциплінарної інтеграції / За

редакцією Л.Л. Омельчук, О.М. Ткаченка, О.В. Шишацької. – Одеса: Айс Принт, 2021. – 161 с.

13. Коротке знайомство з Gradle [Електрон. ресурс] – Режим доступу: <https://javarush.com/ua/groups/posts/uk.2126.korotke-znayomstvo-z-gradle> .

14. Каграманова, Ю. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти [Електрон. ресурс] ./ Ю. Каграманова – Режим доступу: <https://dou.ua/forums/topic/40575/> .

ДОДАТОК А
Технічне завдання

Вступ

Мобільний застосунок для вивчення ПДР, у сфері освіти або для особистого користування, який також використовується для вивчення правил дорожнього руху в будь яку вільну хвилинку.

A.1 Підстава для розробки

Розробка здійснювалася на основі завдання на дипломну кваліфікаційну роботу за темою «Розробка мобільного застосунку для вивчення правил дорожнього руху», у відповідності з наказом №168 від 24 квітня 2024р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Мобільний застосунок призначений для персонального використання для підвищення обізнаності користувачів про правила дорожнього руху.

A.2.1 Функціональне призначення

Функціональне призначення застосунку полягає в тому аби надати користувачу такі можливості:

- постійний доступ до теоретичного матеріалу ;
- поділ матеріалу на теми;
- можливість пошуку необхідної інформації ;
- мати можливість бачити свій прогрес;
- реєстрація;
- авторизація;
- додання інтерактивних методів у навчання.

A.2.2 Експлуатаційне призначення

Програма призначена, як для особистого використання в учбових цілях, так і використання в навчальному процесі.

A3 Основні вимоги до розроблюваної програми

A.3.1 Вимоги до надійності

Додаток повинен забезпечити обробку винятків та стабільну підтримку Android систем.

A.3.2 Умови експлуатації

Для мобільного застосунку необхідно:

- мати мобільний пристрій на операційній системі android;
- для повноціного досвіду використання необхідно знати дані для входу в свій профіль;
- взаємодіяти з додатком можна в будь якому просторі.

A.3.3 Вимоги до складу й параметрів технічних засобів

Для коректного функціонування застосунку необхідна:

- операційна система пристрою – Android;
- мінімальна версія операційної системи Android – 8.0 (Android Oreo);
- максимальна версія операційної системи Android – 14.0 (Android Upside Down Cake);
- 10 МБ вільного місця на пам'яті.

A.4 Вимоги до програмного засобу

A.4.1. Вимоги до складу виконуваних функцій

Додаток має виконувати зазначені нижче функції:

- а) функція подачі даних;
- б) функція відстеження прогресу;
- в) функція використання актуальних методів навчання;
- г) функція реєстрації. Користувач буде вводити свої дані та підтверджувати обліковий запис за допомогою введеної пошти;
- д) функція авторизації. Користувач буде входити у свій обліковий запис через пошту;
- ж) функція пошуку даних.

A.4.2 Вимоги до організації вхідних даних

Під поняттям вхідні дані маються на увазі дані, які надходять в програму з додаткових ресурсів та вводиться користувачем.

Вхідні файли з текстом мають бути у форматі .json. та розділені на:

- Articles;
- tile – назви тем;
- paragraphs – параграфи з обширним вмістом;
- pharagra – строка тексту.

Вхідні дані від користувача це ім'я та логін.

A.4.3 Вимоги до організації вихідних даних

Вихідні дані це вже опрацьовані вхідні дані, які виводяться на екран для користувача. Вихідні дані до програми мають подаватися у вигляді відсортованих списків з воводом лише тих тем, що необхідні користувачу.

A.5 Вимоги до програмної документації

Програмна документація має складатись з:

- технічне завдання відповідно до аналізу предметної області та функціональних завдань;
- звіт з виконаної роботи, що включає: опис етапу проєктування, функціонально-структурної схеми, діаграми структури, опис класів та їх функцій;
- інструкція з експлуатації мобільного застосунку.

A.6 Техніко-економічні показники

Економічна ефективність додатку полягає в зекономлених грошах на теоретичну школу.

Ефективність програмного забезпечення має оцінюватися в зекономленому користувачем часу, користуючись додатком в будь який момент без потреби відділяти окремо час на навчання.

A.7 Стадії та етапи розробки

Розробка ведеться у 4 етапи:

- а). розробка технічного завдання (1 тиждень);
- б) розробка архітектури програмного забезпечення (2 тиждень);

- в) розробка програмного забезпечення (3-4 тиждень);
- г) тестування програмного забезпечення (5 тиждень).

A.8 Порядок контролю та приймання

Розробник повинен провести випробовування функціоналу додатку. Тільки після переконання в повній працездатності застосунок передається замовнику.

ДОДАТОК Б
Текст програми

MainActivity.java

```
package com.example.flashpdr.Activity;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.util.Pair;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.backendless.Backendless;
import com.backendless.BackendlessUser;
import com.backendless.async.callback.AsyncCallback;
import com.backendless.exceptions.BackendlessFault;
import com.backendless.persistence.local.UserTokenStorageFactory;
import com.example.flashpdr.Adapter.SearchAdapter;
import com.example.flashpdr.Data.Const;
import com.example.flashpdr.Data.TextData;
import com.example.flashpdr.Data.global;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
```

```

import java.util.List;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";

    private Button ButtonTheory;
    private Button ButtonTest;
    private ImageButton ButtonProfile;

    private ProgressBar ProgressBarMy;
    private LinearLayout CoverLinearLayoutSearch;
    private LinearLayout LinearLayoutMain;
    private RelativeLayout preferencesRelativeLayout;
    private RelativeLayout searchRelativeLayout;

    private EditText EditTextSearch;
    private ListView ResultListViewSearch;
    private List<Pair<JSONObject, Integer>> DJson;
    private TextView TextViewCoincidence;
    private ImageButton deleteButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Const.context = getApplicationContext();

        ButtonTheory = (Button) findViewById(R.id.ButtonTheory);
        ButtonTest = (Button) findViewById(R.id.ButtonTest);
        ButtonProfile = (ImageButton) findViewById(R.id.ButtonProfile);
        ButtonTheory.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onTheoryClick();
            }
        });
        ButtonTest.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onTheoryClick();
            }
        });
    }
}

```

```

    }
});
ButtonProfile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (! TextData. getTextData().isLoggedIn)
            startLoginActivity();
        else
            startProfileActivity();
    }
});

ButtonProfile.setEnabled(false);
ButtonProfile.setVisibility(View.INVISIBLE);
checkForLogin();

CoverLinearLayoutSearch = (LinearLayout) findViewById(R.id.CoverLinearLayoutSearch);
LinearLayoutMain = (LinearLayout) findViewById(R.id.LinearLayoutMain);
preferencesRelativeLayout = (RelativeLayout) findViewById(R.id.preferencesRelativeLayout);
searchRelativeLayout = (RelativeLayout) findViewById(R.id.searchRelativeLayout);

coincidenceTextView = (TextView) findViewById(R.id.coincidenceTextView);

ProgressBarMy = (ProgressBar) findViewById(R.id.loading_spinner);
ProgressBarMy.setVisibility(View.INVISIBLE);

deleteButton = (ImageButton) findViewById(R.id.deleteButton);
deleteButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        EditTextSearch.setText("");
        displayInfo(new ArrayList<Pair<JSONObject, Integer>>());
        coincidenceTextView.setText("");
        setLayout(1);
    }
});

ResultListViewSearch = (ListView) findViewById(R.id.ResultListViewSearch);
ResultListViewSearch.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

```

```

        TextData.getTextData().chosenArticle = DJson.get(i).first;
        showMore();
    }
});

EditTextSearch = (EditText) findViewById(R.id.EditTextSearch);
EditTextSearch.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
        if (charSequence.length() != 0) {
            setLayout(2);
            ProgressBarMy.setVisibility(View.VISIBLE);

            try {
                findWord(charSequence.toString());
            } catch (JSONException e) {
                e.printStackTrace();
            }
        } else {
            displayInfo(new ArrayList<Pair<JSONObject, Integer>>());
            coincidenceTextView.setText("");
            setLayout(1);
        }
    }
});

@Override
public void afterTextChanged(Editable editable) {
}

});

setLayout(1);
}

private void checkForLogin() {

    String userToken = UserTokenStorageFactory.instance().getStorage().get();

```

```

        if (userToken != null && !userToken.equals("")) {
            String currentUserObjectId = Backendless.UserService.loggedInUser();
            Backendless.UserService.findById(currentUserObjectId,
AsyncCallback<BackendlessUser>() {
                @Override
                public void handleResponse(BackendlessUser response) {
                    Backendless.UserService.setCurrentUser(response);
                    TextData.getTextData().isLoggedIn = true;
                    ButtonProfile.setEnabled(true);
                    ButtonProfile.setVisibility(View.VISIBLE);
                }

                @Override
                public void handleFault(BackendlessFault fault) {
                    Log.e(TAG, "Failed to find loggedIn User: " + fault.getMessage());
                    ButtonProfile.setEnabled(true);
                    ButtonProfile.setVisibility(View.VISIBLE);
                }
            });
        } else {
            Log.e(TAG, "User is not logged in");
            ButtonProfile.setEnabled(true);
            ButtonProfile.setVisibility(View.VISIBLE);
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (EditTextSearch.getText().length() == 0)
            setLayout(1);
        else
            setLayout(2);
    }

    private void setLayout(int layout) {
        if (layout == 1) {
            CoverLinearLayoutSearch.setEnabled(false);
            CoverLinearLayoutSearch.setVisibility(View.INVISIBLE);

            LinearLayoutMain.setEnabled(true);

```

```

LinearLayoutMain.setVisibility(View.VISIBLE);

deleteButton.setEnabled(false);
deleteButton.setVisibility(View.INVISIBLE);

preferencesRelativeLayout.setEnabled(true);
preferencesRelativeLayout.setVisibility(View.VISIBLE);

searchRelativeLayout.setEnabled(true);
searchRelativeLayout.setVisibility(View.VISIBLE);
} else if (layout == 2) {
    CoverLinearLayoutSearch.setEnabled(true);
    CoverLinearLayoutSearch.setVisibility(View.VISIBLE);

    LinearLayoutMain.setEnabled(false);
    LinearLayoutMain.setVisibility(View.INVISIBLE);

    deleteButton.setEnabled(true);
    deleteButton.setVisibility(View.VISIBLE);

    preferencesRelativeLayout.setEnabled(false);
    preferencesRelativeLayout.setVisibility(View.INVISIBLE);

    searchRelativeLayout.setEnabled(true);
    searchRelativeLayout.setVisibility(View.VISIBLE);
} else if (layout == 3) {
    CoverLinearLayoutSearch.setEnabled(false);
    CoverLinearLayoutSearch.setVisibility(View.INVISIBLE);

    LinearLayoutMain.setEnabled(false);
    LinearLayoutMain.setVisibility(View.INVISIBLE);

    deleteButton.setEnabled(false);
    deleteButton.setVisibility(View.INVISIBLE);

    preferencesRelativeLayout.setEnabled(false);
    preferencesRelativeLayout.setVisibility(View.INVISIBLE);

    searchRelativeLayout.setEnabled(false);
    searchRelativeLayout.setVisibility(View.INVISIBLE);
}

```

```
}
```

```
private void onTestClick() {
    global globalVar = new global();
    globalVar.setChecker(true);
    Intent intent = new Intent(this, TrafficSignsActivity.class);
    startActivity(intent);
}
```

```
private void onTheoryClick() {
    global globalVar = new global();
    globalVar.setChecker(false);
    Intent intent = new Intent(this, TheoryActivity.class);
    startActivity(intent);
}
```

```
private void startLoginActivity() {
    Intent intent = new Intent(this, LoginActivity.class);
    startActivity(intent);
}
```

```
private void startProfileActivity() {
    Intent intent = new Intent(this, ProfileActivity.class);
    startActivity(intent);
}
```

```
private void showMore() {
    Intent intent = new Intent(this, ShowTheArticleActivity.class);
    startActivity(intent);
}
```

```
private void findWord(String s) throws JSONException {
    DJson = new ArrayList<>();

    List<JSONObject> pdd = TextData. getTextData().pdd;

    for (int i = 0; i < pdd.size(); i++) {
        JSONArray articles = pdd.get(i).getJSONArray("articles");

        for (int j = 0; j < articles.length(); j++) {
            JSONArray paragraphs = articles.getJSONObject(j).getJSONArray("paragraphs");
```

```

int counter = 0;

for (int k = 0; k < paragraphs.length(); k++) {
    String paragraph = paragraphs.getJSONObject(k).getString("paragraph");
    counter += kmp(s.toLowerCase(), paragraph.toLowerCase());
}

DJson.add(Pair.create(articles.getJSONObject(j), counter));
}
}

List<JSONObject> signs = TextData.getTextData().signs;

for (int i = 0; i < signs.size(); i++) {
    JSONArray articles = signs.get(i).getJSONArray("articles");

    for (int j = 0; j < articles.length(); j++) {
        JSONArray paragraphs = articles.getJSONObject(j).getJSONArray("paragraphs");
        int counter = 0;

        for (int k = 0; k < paragraphs.length(); k++) {
            String paragraph = paragraphs.getJSONObject(k).getString("paragraph");
            Log.e(TAG, paragraph + "!!!!!!!!!!!!!!!!!!!!");
            String imageURL = paragraphs.getJSONObject(k).getString("imageURL").replace('_', '!');
            counter += kmp(s.toLowerCase(), paragraph.toLowerCase() + '@' + imageURL);
        }

        DJson.add(Pair.create(articles.getJSONObject(j), counter));
    }
}

qsort(DJson, 0, DJson.size() - 1);

for (int i = DJson.size() - 1; i >= 0; i--)
    if (DJson.get(i).second == 0)
        DJson.remove(i);

displayInfo(DJson);
}

private int kmp(String s, String paragraph) {

```

```

int ans = 0;
String str = '@' + s + '@' + paragraph;
List<Integer> arr = new ArrayList<>();
arr.add(0);
arr.add(0);

for (int i = 2; i < str.length(); i++) {

    int j = arr.get(i - 1);

    while (j > 0 && str.charAt(j + 1) != str.charAt(i))
        j = arr.get(j);

    if (str.charAt(j + 1) == str.charAt(i))
        arr.add(j + 1);
    else
        arr.add(0);

    if (arr.get(i) == s.length())
        ans++;
}

return ans;
}

private void qsort(List<Pair<JSONObject, Integer>> DJson, int l, int r) {
    int i = l;
    int j = r;
    int s = DJson.get(l + (r - l) / 2).second;

    while (i <= j) {
        while (DJson.get(i).second > s) i++;
        while (DJson.get(j).second < s) j--;

        if (i <= j) {
            Pair<JSONObject, Integer> d = DJson.get(i);
            DJson.set(i, DJson.get(j));
            DJson.set(j, d);
            i++; j--;
        }
    }
}

```

```

    }

    if (l < j) qsort(DJson, l, j);
    if (i < r) qsort(DJson, i, r);
}

private void displayInfo(List<Pair<JSONObject, Integer>> DJson) {
    SearchAdapter adapter = new SearchAdapter(this, DJson);
    ResultListViewSearch.setAdapter(adapter);
    if (DJson.size() > 0 && (DJson.size() % 10 > 4 || DJson.size() % 10 == 0 || (DJson.size() % 100 >= 11
&& DJson.size() % 100 <= 14)))
        coincidenceTextView.setText("Знайдено " + DJson.size() + " топіків");
    else if (DJson.size() % 10 > 1)
        coincidenceTextView.setText("Знайдено " + DJson.size() + " топіків ");
    else if (DJson.size() % 10 == 1)
        coincidenceTextView.setText("найдена " + DJson.size() + " топік");
    else
        coincidenceTextView.setText("Нічого не знайдено");

    ProgressBarMy.setVisibility(View.INVISIBLE);
}

@Override
protected void onDestroy() {
    ClearCache.CacheDelete(this);
    super.onDestroy();
}
}

```

ArticleActivity.java

```

package com.example.flashpdr.Activity;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.widget.ImageButton;
import android.widget.ListView;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Adapter.ArticleAdapterSingle;
import com.example.flashpdr.Data.TextData;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

import org.json.JSONArray;
import org.json.JSONException;

```

```

public class ArticleActivity extends AppCompatActivity {

    private ListView listView;
    private ImageButton backButton;

    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_article);

        getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
        getSupportActionBar().setCustomView(R.layout.pdr_action_bar);

        backButton = (ImageButton) findViewById(R.id.backButton);

        listView = (ListView) findViewById(R.id.articleListView);

        try {
            displayInfo( TextData. getTextData().chosenArticle.getJSONArray("paragraphs"));
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    private void displayInfo(JSONArray paragraphs) {
        ArticleAdapterSingle adapter = new ArticleAdapterSingle(this, paragraphs);
        listView.setAdapter(adapter);
    }

    @Override
    protected void onDestroy() {
        ClearCache.CacheDelete(this);
        super.onDestroy();
    }
}

```

CardsActivity.java

```

package com.example.flashpdr.Activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Adapter.CardsAdapter;
import com.example.flashpdr.Adapter.TrafficSignsChaptersListAdapter;

```

```

import com.example.flashpdr.Data. TextData;
import com.example.flashpdr.R;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class CardsActivity extends AppCompatActivity {
    private ImageView questionImageView;
    private TextView answerTextView;
    private Button showAnswerButton;
    private ImageButton backButton;
    private Button nextCardButton;
    private TrafficSignsChaptersListAdapter adapter;
    private List<Flashcard> flashcards;
    private CardsAdapter cardsAdapter;

    // Assuming you have a list of flashcards
    //private ArrayList<Flashcard> flashcards;
    private int currentIndex = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cards);
        getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
        getSupportActionBar().setCustomView(R.layout.image_action_bar);
        flashcards = new ArrayList<>();
        int position = getIntent().getIntExtra("position", 0);

        try {
            JSONArray signs = TextData. getTextData().signs.get(position).getJSONArray("articles");

            for (int i = 0; i < signs.length(); i++) {
                JSONObject sign = signs.getJSONObject(i);

                int imageResId = getResources().getIdentifier(sign.getString("title"), "drawable",
getPackageName());
                String paragraph = sign.getString("paragraph");

                flashcards.add(new Flashcard(imageResId, paragraph));
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }

        backButton = (ImageButton) findViewById(R.id.backButton);
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });

        questionImageView = findViewById(R.id.question_image_view);
        answerTextView = findViewById(R.id.answer_text_view);
        showAnswerButton = findViewById(R.id.show_answer_button);

```

```

nextCardButton = findViewById(R.id.next_card_button);

showAnswerButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        answerTextView.setVisibility(View.VISIBLE);
        showAnswerButton.setVisibility(View.INVISIBLE);
    }
});

Intent intent = getIntent();
position = getIntent().getIntExtra("position", 0);

// Get flashcard from position
Flashcard currentCard = getFlashcard(position);

if(currentCard != null) {
    // Update ImageView with the title image from the current card
    questionImageView.setImageResource(currentCard.getImageResId());
    // Update Answer TextView with answer from the current card
    answerTextView.setText(currentCard.getParagraph());
    answerTextView.setVisibility(View.INVISIBLE); // initially hide the answer
}

nextCardButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        currentIndex = (currentIndex + 1) % flashcards.size();
        displayCard(flashcards.get(currentIndex));
    }
});
}
class Flashcard {
    private final int imageResId;
    private final String paragraph;

    Flashcard(int imageResId, String paragraph) {
        this.imageResId = imageResId;
        this.paragraph = paragraph;
    }

    public int getImageResId() {
        return imageResId;
    }

    public String getParagraph() {
        return paragraph;
    }
}

private void displayCard(Flashcard flashcard) {
    if (flashcard == null) return;
    questionImageView.setImageResource(flashcard.getImageResId());
    answerTextView.setText(flashcard.getParagraph());
    answerTextView.setVisibility(View.INVISIBLE); // Initially hide the answer
}

private Flashcard getFlashcard(int position) {
    try {
        JSONArray signs = TextData.getTextData().signs.get(position).getJSONArray("articles");
        if (currentIndex < signs.length()) {

```

```

        JSONObject sign = signs.getJSONObject(currentIndex);
        int imageResId = getResources().getIdentifier(sign.getString("title"), "drawable",
getPackageName());
        String paragraph = sign.getString("paragraph");
        return new Flashcard(imageResId, paragraph);
    }
} catch (JSONException e) {
    e.printStackTrace();
}
return null;
}

private void displayInfo(List<JSONObject> data, List<Boolean> isDivider) {
    StringBuilder stringBuilder = new StringBuilder();
    for (JSONObject jsonObject : data) {
        try {
            stringBuilder.append(jsonObject.getString("paragraph"));
            stringBuilder.append("\n");
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    answerTextView.setText(stringBuilder.toString());
}

private void getJSONData(JSONArray articles) throws JSONException {

    List<JSONObject> data = new LinkedList<>();
    List<Boolean> isDivider = new LinkedList<>();

    for (int i = 0; i < articles.length(); i++) {
        JSONArray paragraphs = articles.getJSONObject(i).getJSONArray("paragraphs");
        int length = paragraphs.length();

        for (int j = 0; j < length; j++) {
            data.add(paragraphs.getJSONObject(j));

            if (j < length - 1)
                isDivider.add(false);
            else
                isDivider.add(true);
        }
    }

    displayInfo(data, isDivider);
}
}

```

ImageActivity.java

```

package com.example.flashpdr.Activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.ListView;
import android.widget.ProgressBar;

```

```

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Adapter.ImageListAdapter;
import com.example.flashpdr.Data.TextData;
//import com.example.flashpdr.Data.global;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

public class ImageActivity extends AppCompatActivity {

    private static final String TAG = "ImageActivity";

    private ImageButton backButton;
    private ProgressBar ProgressBarMy;
    private ListView listView;
    private ImageListAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_image);

        getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
        getSupportActionBar().setCustomView(R.layout.image_action_bar);

        backButton = (ImageButton) findViewById(R.id.backButton);
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });

        ProgressBarMy = (ProgressBar) findViewById(R.id.loading_spinner);
        ProgressBarMy.setVisibility(View.VISIBLE);

        listView = (ListView) findViewById(R.id.listView);
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                onTitleClick(i);
            }
        });

        TextData.setContext(this);

        displayInfo();
    }

    private void onTitleClick(int i) {
        global globalVar = new global();
        if(globalVar.isChecker()){
            Intent intent = new Intent(this, InformationActivity.class);
            intent.putExtra("position", i);
            intent.putExtra("from", "ImageActivity");
            startActivity(intent);
        }
        else{

```

```

        Intent intent = new Intent(this, CardsActivity.class);
        intent.putExtra("position", i);
        intent.putExtra("from", "ImageActivity");
        startActivity(intent);
    }
}

private void displayInfo() {
    adapter = new ImageListAdapter(this);
    listView.setAdapter(adapter);

    ProgressBarMy.setVisibility(View.INVISIBLE);
}

@Override
protected void onDestroy() {
    ClearCache.CacheDelete(this);
    super.onDestroy();
}
}

```

InformationActivity.java
package com.example.flashpdr.Activity;

```

import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.TextView;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Adapter.ArticlesAdapter;
import com.example.flashpdr.Data.TextData;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.LinkedList;
import java.util.List;

public class InformationActivity extends AppCompatActivity {

    private static final String TAG = "InformationActivity";
    private TextView testTextView;
    private ListView listView;
    private ImageButton backButton;
    private String fromActivity;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_information);

        int position = getIntent().getExtras().getInt("position");
        fromActivity = getIntent().getExtras().getString("from");
    }
}

```

```

if (fromActivity.equals("PdrActivity")) {
    getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
    getSupportActionBar().setCustomView(R.layout.pdr_action_bar);
} else if (fromActivity.equals("ImageActivity")) {
    getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
    getSupportActionBar().setCustomView(R.layout.image_action_bar);
}
backButton = (ImageButton) findViewById(R.id.backButton);
backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();
    }
});

testTextView = (TextView) findViewById(R.id.testTextView);

try {
    if (fromActivity.equals("PdrActivity"))
        testTextView.setText( TextData. getTextData().pdd.get(position).getString("title"));
    else if (fromActivity.equals("ImageActivity"))
        testTextView.setText( TextData. getTextData().signs.get(position).getString("title"));
} catch (JSONException e) {
    e.printStackTrace();
}

listView = (ListView) findViewById(R.id.articlesListView);

try {
    if (fromActivity.equals("PdrActivity"))
        getJSONData( TextData. getTextData().pdd.get(position).getJSONArray("articles"));
    else if (fromActivity.equals("ImageActivity"))
        getJSONData( TextData. getTextData().signs.get(position).getJSONArray("articles"));
} catch (JSONException e) {
    e.printStackTrace();
}
}

private void getJSONData(JSONArray articles) throws JSONException {

    List<JSONObject> data = new LinkedList<>();
    List<Boolean> isDivider = new LinkedList<>();

    for (int i = 0; i < articles.length(); i++) {
        JSONArray paragraphs = articles.getJSONObject(i).getJSONArray("paragraphs");
        int length = paragraphs.length();

        for (int j = 0; j < length; j++) {
            data.add(paragraphs.getJSONObject(j));

            if (j < length - 1)
                isDivider.add(false);
            else
                isDivider.add(true);
        }
    }

    displayInfo(data, isDivider);
}

private void displayInfo(List<JSONObject> data, List<Boolean> isDivider) {

```

```

        ArticlesAdapter adapter = new ArticlesAdapter(this, data, isDivider, fromActivity);
        listView.setAdapter(adapter);
    }

    @Override
    protected void onDestroy() {
        ClearCache.CacheDelete(this);
        super.onDestroy();
    }
}

```

LoginActivity.java

```

package com.example.flashpdr.Activity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

public class LoginActivity extends AppCompatActivity {

    private RelativeLayout RelativeLayoutLogin;
    private RelativeLayout RelativeLayoutRegistration;

    private ImageView ImageViewApp;
    private EditText EditTextUsername;
    private EditText EditTextPassword;
    private ImageButton backButton;
    private Button ButtonLogin;
    private TextView TextViewRegistration;

    private String email;
    private String username;
    private String password;
    private String retryPassword;

    private ImageButton backImageButton;
    private Button registrationButton;
    private EditText usernameRegEditText;
    private EditText emailRegEditText;
    private EditText passwordRegEditText;
    private EditText retryPasswordRegEditText;
    private ImageView appRegImageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
    }
}

```

```

//Backendless.initApp(this, Const.APP_ID, Const.ANDROID_KEY);

RelativeLayoutLogin = (RelativeLayout) findViewById(R.id.RelativeLayoutLogin);
RelativeLayoutRegistration = (RelativeLayout) findViewById(R.id.registrationRelativeLayout);
backButton = (ImageButton) findViewById(R.id.backImageButton);
backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();}
});

setLayout(1);

EditTextUsername = (EditText) findViewById(R.id.EditTextUsername);
EditTextPassword = (EditText) findViewById(R.id.EditTextPassword);

ButtonLogin = (Button) findViewById(R.id.ButtonLogin);
ButtonLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        username = EditTextUsername.getText().toString();
        password = EditTextPassword.getText().toString();
        if (username == "" || password == "")
            Toast.makeText(getApplicationContext(), "Заполните все поля",
Toast.LENGTH_SHORT).show();
        else
            onLoginClick();
    }
});

TextViewRegistration = (TextView) findViewById(R.id.TextViewRegistration);
TextViewRegistration.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setLayout(2);
    }
});

backImageButton = (ImageButton) findViewById(R.id.backImageButton);
backImageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setLayout(1);
    }
});

usernameRegEditText = (EditText) findViewById(R.id.usernameRegEditText);
emailRegEditText = (EditText) findViewById(R.id.emailRegEditText);
passwordRegEditText = (EditText) findViewById(R.id.passwordRegEditText);
retryPasswordRegEditText = (EditText) findViewById(R.id.retryPasswordRegEditText);

registrationButton = (Button) findViewById(R.id.registrationButton);
registrationButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        username = usernameRegEditText.getText().toString();

```

```

        email = emailRegEditText.getText().toString();
        password = passwordRegEditText.getText().toString();
        retryPassword = retryPasswordRegEditText.getText().toString();
        if (username == "" || email == "" || password == "" || retryPassword == "")
            Toast.makeText(getApplicationContext(), "Заповніть всі поля",
Toast.LENGTH_SHORT).show();
        else if (!password.equals(retryPassword))
            Toast.makeText(getApplicationContext(), "Пароли не однакові",
Toast.LENGTH_SHORT).show();
        else
            onRegistrationClick();
    }
    });
}

private void onRegistrationClick() {

    SharedPreferences prefs = getSharedPreferences("UserPrefs", MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();

    // Checks if user is already registered
    if (prefs.getString(username, null) != null) {
        Toast.makeText(getApplicationContext(), "User already registered",
Toast.LENGTH_SHORT).show();
        return;
    }

    // Registers new user
    editor.putString(username, password);
    editor.apply();

    Toast.makeText(getApplicationContext(), "Registration successful",
Toast.LENGTH_SHORT).show();
    startProfileActivity();
    finish();
}

private void onLoginClick() {
    EditText EditTextUsername = findViewById(R.id.EditTextUsername); // Not usernameRegEditText
    EditText EditTextPassword = findViewById(R.id.EditTextPassword);

    String enteredUsername = EditTextUsername.getText().toString();
    String enteredPassword = EditTextPassword.getText().toString();

    SharedPreferences prefs = getSharedPreferences("UserPrefs", MODE_PRIVATE);

    String storedPassword = prefs.getString(enteredUsername, null);

    if (storedPassword != null) {
        if (storedPassword.equals(enteredPassword)) {
            startProfileActivity();
            finish();
        } else {
            // Wrong password for the entered username
            Toast.makeText(getApplicationContext(), "Invalid Username or Password",
Toast.LENGTH_SHORT).show();
        }
    } else {
        // User does not exist or wrong username

```

```

        Toast.makeText(getApplicationContext(), "Invalid Username or Password",
        Toast.LENGTH_SHORT).show();
    }
}

private void startProfileActivity() {
    Intent intent = new Intent(this, ProfileActivity.class);
    startActivity(intent);
}

private void setLayout(int layout) {
    if (layout == 1) {
        RelativeLayoutLogin.setEnabled(true);
        RelativeLayoutLogin.setVisibility(View.VISIBLE);

        RelativeLayoutRegistration.setEnabled(false);
        RelativeLayoutRegistration.setVisibility(View.INVISIBLE);
    } else if (layout == 2) {
        RelativeLayoutLogin.setEnabled(false);
        RelativeLayoutLogin.setVisibility(View.INVISIBLE);

        RelativeLayoutRegistration.setEnabled(true);
        RelativeLayoutRegistration.setVisibility(View.VISIBLE);
    }
}

@Override
protected void onDestroy() {
    ClearCache.CacheDelete(this);
    super.onDestroy();
}
}

```

PdrActivity.java

```

package com.example.flashpdr.Activity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.ProgressBar;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Adapter.ListAdapterPdrChapters;
import com.example.flashpdr.Data.TextData;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

public class PdrActivity extends AppCompatActivity {

    private static final String TAG = "PdrActivity";
    private ListView listView;
    private ListAdapterPdrChapters adapter;
    private ProgressBar ProgressBarMy;

```

```

private ImageButton backButton;

@SuppressLint("MissingInflatedId")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_pdr);

    getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
    getSupportActionBar().setCustomView(R.layout.pdr_action_bar);

    backButton = (ImageButton) findViewById(R.id.backButton);
    backButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    });

    listView = (ListView) findViewById(R.id.listView);
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            onTitleClick(i);
        }
    });

    TextData.setContext(this);
    displayInfo();
}

private void onTitleClick(int i) {
    Intent intent = new Intent(this, InformationActivity.class);
    intent.putExtra("position", i);
    intent.putExtra("from", "PdrActivity");
    startActivity(intent);
}

private void displayInfo() {
    adapter = new ListAdapterPdrChapters(this);
    listView.setAdapter(adapter);

    ProgressBarMy.setVisibility(View.INVISIBLE);
}

@Override
protected void onDestroy() {
    ClearCache.CacheDelete(this);
    super.onDestroy();
}
}

```

```

ProfileActivity.java
package com.example.flashpdr.Activity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;

```

```

import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.backendless.Backendless;
import com.backendless.async.callback.AsyncCallback;
import com.backendless.exceptions.BackendlessFault;
import com.example.flashpdr.Data.TextData;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

import java.util.ArrayList;

public class ProfileActivity extends AppCompatActivity {

    private static final String TAG = "ProfileActivity";

    private TextView TextViewUsername;
    private TextView TextViewTitle;
    private TextView TextViewCar;

    private TextView TextViewExperience;
    private TextView TextViewScoreWord;

    private TextView TextViewExperienceLeft;

    private LinearLayout logoutLinearLayout;
    private int totalScore;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);

        logoutLinearLayout = (LinearLayout) findViewById(R.id.logoutLinearLayout);
        logoutLinearLayout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onLogoutClick();
            }
        });

        TextViewUsername = (TextView) findViewById(R.id.TextViewUsername);

        TextViewUsername.setText(Backendless.UserService.CurrentUser().getProperty("name").toString());

        totalScore = (Integer) Backendless.UserService.CurrentUser().getProperty("totalScore");
        TextViewExperience = (TextView) findViewById(R.id.TextViewExperience);
        TextViewExperience.setText("" + totalScore);

        TextViewScoreWord = (TextView) findViewById(R.id.TextViewScoreWord);
        if (totalScore % 100 != 11 && totalScore % 10 == 1)
            TextViewScoreWord.setText("очко");
        else if ((totalScore % 100 < 12 || totalScore % 100 > 14) && (totalScore % 10 >= 2 && totalScore %
10 <= 4))
            TextViewScoreWord.setText("очка");
        else

```

```

        TextViewScoreWord.setText("очков");
    }

    private void onLogoutClick() {

        Log.d(TAG, "Logging out started!!!");

        Backendless.UserService.logout(new AsyncCallback<Void>() {
            @Override
            public void handleResponse(Void response) {
                Log.d(TAG, "Logged out successfully!!!");
                TextData.getTextData().isLoggedIn = false;
                TextData.getTextData().maxCorrect = new ArrayList<Integer>();
                Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
            }

            @Override
            public void handleFault(BackendlessFault fault) {
                Log.e(TAG, "Logging out failed! " + fault.getMessage());
                Toast.makeText(getApplicationContext(), "Logging Out Failed!!!", Toast.LENGTH_SHORT);
            }
        });
    }

    @Override
    protected void onDestroy() {
        ClearCache.CacheDelete(this);
        super.onDestroy();
    }
}

TheoryActivity.java
package com.example.flashpdr.Activity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListView;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.example.flashpdr.Adapter.TopicsListAdapter;
import com.example.flashpdr.Helper.ClearCache;
import com.example.flashpdr.R;

import java.util.LinkedList;
import java.util.List;

public class TheoryActivity extends AppCompatActivity {

    private ListView listView;
    private ImageButton backButton;

    @SuppressWarnings("MissingInflatedId")

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_theory);

    getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
    getSupportActionBar().setCustomView(R.layout.theory_action_bar);

    backButton = (ImageButton) findViewById(R.id.backButton);
    backButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    });

    listView = (ListView) findViewById(R.id.listView);
    setListAdapter();
}

private void setListAdapter() {
    List<String> topics = new LinkedList<>();
    topics.add("Правила Дорожнього Руху");
    topics.add("Дорожні знаки");

    TopicsListAdapter adapter = new TopicsListAdapter(this, topics);
    listView.setAdapter(adapter);

    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            if (i == 0) {
                runActivity(PdrActivity.class);
            } else if (i == 1) {
                runActivity(Image.class);
            }
        }
    });
}

private void runActivity(Class activity) {
    Intent intent = new Intent(this, activity);
    startActivity(intent);
}

@Override
protected void onDestroy() {
    ClearCache.CacheDelete(this);
    super.onDestroy();
}
}

```

ArticleAdapter.java

```
package com.example.flashpdr.Adapter;
```

```
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
```

```

import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.bumptech.glide.Glide;
import com.example.flashpdr.R;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

public class ArticlesAdapter extends BaseAdapter {

    private List<JSONObject> paragraphs;
    private List<Boolean> isDivider;
    private String fromActivity;
    private Context context;
    private LayoutInflater inflater;

    public ArticlesAdapter(Context context, List<JSONObject> paragraphs, List<Boolean> isDivider,
String fromActivity) {
        this.context = context;
        this.paragraphs = paragraphs;
        this.isDivider = isDivider;
        this.fromActivity = fromActivity;
        this.inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return paragraphs.size();
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

```

```

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {

    String image = null;

    try {
        if (fromActivity.equals("ImageActivity"))
            image = paragraphs.get(i).getString("imageUrl");
    } catch (JSONException e) {
        e.printStackTrace();
    }

    ViewHolder viewHolder;

    if (view == null) {
        if (image == null)
            view = inflater.inflate(R.layout.row_article_item, viewGroup, false);
        else
            view = inflater.inflate(R.layout.row_article_image_item, viewGroup, false);
        viewHolder = new ViewHolder(view, image != null);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }

    try {
        viewHolder.articleTextView.setText(paragraphs.get(i).getString("paragraph"));

        if (image != null) {
            Glide.with(context).load(context.getResources().getIdentifier("@drawable/" + image, null,
context.getPackageName())).into(viewHolder.imageView);

            String str = "";
            for (int j = 1; j < image.length(); j++)
                if (image.charAt(j) == '_')

```

```

        str += '!';
    else
        str += image.charAt(j);

    viewHolder.imageView.setText(str);

    if (!isDivider.get(i) || i == isDivider.size() - 1)
        viewHolder.ViewSeparator.setVisibility(View.INVISIBLE);
    else
        viewHolder.ViewSeparator.setVisibility(View.VISIBLE);
    }
} catch (JSONException e) {
    e.printStackTrace();
}

return view;
}

private class ViewHolder {
    TextView articleTextView;
    ImageView imageView;
    TextView imageTextView;
    View ViewSeparator;

    public ViewHolder(View v, boolean isImage) {
        articleTextView = (TextView) v.findViewById(R.id.articleTextView);
        if (isImage) {
            imageView = (ImageView) v.findViewById(R.id.imageView);
            imageTextView = (TextView) v.findViewById(R.id.imageTextView);
            ViewSeparator = v.findViewById(R.id.separatorView);
        }
    }
}
}

```

CardsAdapter.java

```

package com.example.flashpdr.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.bumptech.glide.Glide;
import com.example.flashpdr.R;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

public class CardsAdapter extends BaseAdapter {
    private List<JSONObject> paragraphs;
    private List<Boolean> isDivider;
    private String fromActivity;
    private Context context;
    private LayoutInflater inflater;

    public CardsAdapter(Context context, List<JSONObject> paragraphs, List<Boolean> isDivider, String
fromActivity) {
        this.context = context;
        this.paragraphs = paragraphs;
        this.isDivider = isDivider;
        this.fromActivity = fromActivity;
        this.inflater =
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

    }

    @Override
    public int getCount() {
        return paragraphs.size();
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

```

```

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {

    String image = null;

    try {
        if (fromActivity.equals("ImageActivity"))
            image = paragraphs.get(i).getString("imageUrl");
    } catch (JSONException e) {
        e.printStackTrace();
    }

    CardsAdapter.ViewHolder viewHolder;

    if (view == null) {
        if (image == null)
            view = inflater.inflate(R.layout.row_article_item, viewGroup, false);
        else
            view = inflater.inflate(R.layout.row_article_image_item, viewGroup, false);
        viewHolder = new CardsAdapter.ViewHolder(view, image != null);
        view.setTag(viewHolder);
    } else {
        viewHolder = (CardsAdapter.ViewHolder) view.getTag();
    }

    try {
        viewHolder.articleTextView.setText(paragraphs.get(i).getString("paragraph"));

        if (image != null) {
            Glide.with(context).load(context.getResources().getIdentifier("@drawable/" + image, null,
context.getPackageName())).into(viewHolder.imageView);

            String str = "";
            for (int j = 1; j < image.length(); j++)
                if (image.charAt(j) == '_')
                    str += '!';

```

```

        else
            str += image.charAt(j);

        viewHolder.imageTextView.setText(str);

        if (!isDivider.get(i) || i == isDivider.size() - 1)
            viewHolder.separatorView.setVisibility(View.INVISIBLE);
        else
            viewHolder.separatorView.setVisibility(View.VISIBLE);
    }
} catch (JSONException e) {
    e.printStackTrace();
}

return view;
}

private class ViewHolder {
    TextView articleTextView;
    ImageView imageView;
    TextView imageTextView;
    View separatorView;

    public ViewHolder(View v, boolean isImage) {
        articleTextView = (TextView) v.findViewById(R.id.articleTextView);
        if (isImage) {
            imageView = (ImageView) v.findViewById(R.id.imageView);
            imageTextView = (TextView) v.findViewById(R.id.imageTextView);
            separatorView = v.findViewById(R.id.separatorView);
        }
    }
}

}

ImageAdapter.com

package com.example.flashpdr.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

```

import android.widget.BaseAdapter;
import android.widget.TextView;

import com.example.flashpdr.Data. TextData;
import com.example.flashpdr.R;

import org.json.JSONException;

public class ImageAdapter extends BaseAdapter {

    private Context context;
    private LayoutInflater inflater;

    public ImageAdapter(Context context) {
        this.context = context;
        this.inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return TextData. getTextData().signs.size();
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {

        ViewHolder viewHolder = null;

        if (view == null) {

```

```

        view = inflater.inflate(R.layout.row_chapter_item, viewGroup, false);
        viewHolder = new ViewHolder(view);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }

    try {
        viewHolder.TextViewTitle.setText( TextData. getTextData().signs.get(i).getString("title"));
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return view;
}

private class ViewHolder {
    TextView TextViewTitle;

    public ViewHolder(View v) {
        TextViewTitle = (TextView) v.findViewById(R.id.TextViewTitle);
    }
}
}

```

informationAdapter.java

```
package com.example.flashpdr.Adapter;
```

```

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

```

```

import com.bumptech.glide.Glide;
import com.example.flashpdr.R;

```

```
import org.json.JSONArray;
```

```

import org.json.JSONException;

public class ArticleAdapterSingle extends BaseAdapter {

    private JSONArray paragraphs;
    private Context context;
    private LayoutInflater inflater;

    public ArticleAdapterSingle(Context context, JSONArray paragraphs) {
        this.context = context;
        this.paragraphs = paragraphs;
        this.inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return paragraphs.length();
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {

        String image = null;

        try {
            image = paragraphs.getJSONObject(i).getString("imageUrl");
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

```

```

ViewHolder viewHolder;

if (view == null) {
    if (image == null)
        view = inflater.inflate(R.layout.row_article_item, viewGroup, false);
    else
        view = inflater.inflate(R.layout.row_article_image_item, viewGroup, false);
    viewHolder = new ViewHolder(view, image != null);
    view.setTag(viewHolder);
} else {
    viewHolder = (ViewHolder) view.getTag();
}

try {
    viewHolder.TextViewArticle.setText(paragraphs.getJSONObject(i).getString("paragraph"));

    if (image != null) {
        Glide.with(context).load(context.getResources().getIdentifier("@drawable/" + image, null,
context.getPackageName())).into(viewHolder.ViewImage);

        String str = "";
        for (int j = 1; j < image.length(); j++)
            if (image.charAt(j) == '_')
                str += '!';
            else
                str += image.charAt(j);

        viewHolder.TextViewImage.setText(str);

        viewHolder.ViewSeparator.setVisibility(View.INVISIBLE);
    }
} catch (JSONException e) {
    e.printStackTrace();
}

return view;
}

private class ViewHolder {
    TextView TextViewArticle;

```

```

    ImageView ViewImage;
    TextView TextViewImage;
    View ViewSeparator;

    public ViewHolder(View v, boolean isImage) {
        TextViewArticle = (TextView) v.findViewById(R.id.TextViewArticle);
        if (isImage) {
            ViewImage = (ImageView) v.findViewById(R.id.ViewImage);
            TextViewImage = (TextView) v.findViewById(R.id.TextViewImage);
            ViewSeparator = v.findViewById(R.id.ViewSeparator);
        }
    }
}

```

PdrAdapter.java

```

package com.example.flashpdr.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import com.example.flashpdr.Data. TextData;
import com.example.flashpdr.R;

import org.json.JSONException;

public class ListAdapterPdrChapters extends BaseAdapter {

    private Context context;
    private LayoutInflater inflater;

    public ListAdapterPdrChapters(Context context) {
        this.context = context;
        this.inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }
}

```

```

@Override
public int getCount() {
    return TextData.getTextData().pdd.size();
}

@Override
public Object getItem(int i) {
    return null;
}

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {

    ViewHolder viewHolder = null;

    if (view == null) {
        view = inflater.inflate(R.layout.row_chapter_item, viewGroup, false);
        viewHolder = new ViewHolder(view);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }

    try {
        viewHolder.TextViewTitle.setText( TextData. getTextData().pdd.get(i).getString("title"));
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return view;
}

private class ViewHolder {
    TextView TextViewTitle;

```

```

    public ViewHolder(View v) {
        TextViewTitle = (TextView) v.findViewById(R.id.TextViewTitle);
    }
}

```

SearchAdapter.java

```
package com.example.flashpdr.Adapter;
```

```

import android.content.Context;
import android.util.Pair;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

```

```
import com.example.flashpdr.R;
```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

```

```
import java.util.List;
```

```
public class SearchAdapter extends BaseAdapter {
```

```

    private Context context;
    private LayoutInflater inflater;
    private List<Pair<JSONObject, Integer>> DJson;

```

```

    public SearchAdapter(Context context, List<Pair<JSONObject, Integer>> DJson) {
        this.context = context;
        this.inflater = context.getSystemService(Context.LAYOUT_INFLATER_SERVICE); // (LayoutInflater)
        this.DJson = DJson;
    }

```

```

@Override
public int getCount() {
    return DJson.size();
}

@Override
public Object getItem(int i) {
    return null;
}

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {

    ViewHolder viewHolder = null;

    if (view == null) {
        view = inflater.inflate(R.layout.row_search_result_item, viewGroup, false);
        viewHolder = new ViewHolder(view);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }

    try {
        JSONArray paragraphs = DJson.get(i).first.getJSONArray("paragraphs");

        int j = 0;
        while (paragraphs.getJSONObject(j).getString("paragraph").length() == 0 && j <
paragraphs.length())
            j++;

        viewHolder.TextViewTitle.setText("
+
paragraphs.getJSONObject(j).getString("paragraph").toString());
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

```

        return view;
    }

    private class ViewHolder {
        TextView TextViewTitle;

        public ViewHolder(View v) {
            TextViewTitle = (TextView) v.findViewById(R.id.TextViewTitle);
        }
    }
}

```

TopicsAdapter.java

```

package com.example.flashpdr.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import com.example.flashpdr.R;

import java.util.List;

public class TopicsAdapter extends BaseAdapter {

    private Context context;
    private LayoutInflater inflater;
    private List<String> topics;

    public TopicsAdapter(Context context, List<String> topics) {
        this.context = context;
        this.inflater = context.getSystemService(Context.LAYOUT_INFLATER_SERVICE); // (LayoutInflater)
        this.topics = topics;
    }

    @Override

```

```

public int getCount() {
    return topics.size();
}

@Override
public Object getItem(int i) {
    return null;
}

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {

    ViewHolder viewHolder = null;

    if (view == null) {
        view = inflater.inflate(R.layout.row_topic_item, viewGroup, false);
        viewHolder = new ViewHolder(view);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }

    viewHolder.TextViewTitle.setText(topics.get(i));

    return view;
}

private class ViewHolder {
    TextView TextViewTitle;

    public ViewHolder(View v) {
        TextViewTitle = (TextView) v.findViewById(R.id.TextViewTitle);
    }
}
}
Clear.java

```



```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class TextData {

    private static TextData textData;
    private static Context context;

    public static void setContext(Context contextValue) {
        context = contextValue.getApplicationContext();
    }

    public static TextData getTextData() {
        if (textData == null)
            textData = new TextData();
        return textData;
    }

    public boolean isLoggedIn;

    public List<JSONObject> pdr;
    public List<JSONObject> signs;

    public JSONObject chosenArticle;

    public TextData() {

        Context context;

        isLoggedIn = false;

```

```

maxCorrect = new ArrayList<>();

pdr = new LinkedList<>();
getData("pdr");

signs = new LinkedList<>();
getData("signs");

}

private void getData(String fileName) {

    String json;

    AssetManager assetManager = context.getAssets();

    try {
        InputStream inputStream = assetManager.open(fileName + ".json",
AssetManager.ACCESS_BUFFER);
        int size = inputStream.available();

        byte[] buffer = new byte[size];
        inputStream.read(buffer);

        inputStream.close();

        json = new String(buffer, "UTF-8");
        JSONArray data = new JSONArray(json);

        chooser(data, fileName);
    } catch (IOException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}

```

```
private void chooser(JSONArray data, String fileName) {
```

```
    for (int i = 0; i < data.length(); i++)
        try {
            if (fileName.equals("pdd"))
                pdr.add(data.getJSONObject(i));
            else if (fileName.equals("signs"))
                signs.add(data.getJSONObject(i));
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

```
AndroidManifest.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.FlashPDR"
        tools:targetApi="34">
        <activity
            android:name=".Activity.MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Activity.TheoryActivity"
            android:exported="true">
        </activity>
```

```

<activity android:name=".Activity.PdrActivity" />
<activity
    android:name=".Activity.TestActivity"
    android:screenOrientation="portrait" />
<activity android:name=".Activity.InformationActivity" />

```

```

<activity
    android:name=".Activity.ArticleActivity"
    android:screenOrientation="portrait" />
<activity android:name=".Activity.ImageActivity" />
<activity android:name=".Activity.LoginActivity"
    android:theme="@style/MainActivity"
    android:screenOrientation="portrait"
/>
<activity android:name=".Activity.ProfileActivity"
    android:theme="@style/MainActivity"
    android:screenOrientation="portrait"
/>
<activity android:name=".Activity.CardsActivity" />

```

```

</application>

```

```

</manifest>

```

Activity_cards.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Activity.CardsActivity">

```

```

<ImageView
    android:id="@+id/question_image_view"
    android:layout_width="match_parent"
    android:layout_height="339dp"
    android:contentDescription="@string/question_image"
    android:scaleType="centerCrop" />

```

```

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="57dp"

```

```

android:layout_below="@id/question_image_view"
android:layout_marginTop="2dp"
android:orientation="horizontal">

```

```

<Button
    android:id="@+id/show_answer_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/question_image_view"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="4.5dp"
    android:text="Відповідь" />

```

```

<Button
    android:id="@+id/next_card_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:layout_marginTop="4.5dp"
    android:layout_marginBottom="4.5dp"
    android:text="Наступний знак" />

```

```

</LinearLayout>

```

```

<TextView
    android:id="@+id/answer_text_view"
    android:layout_width="450dp"
    android:layout_height="343dp"
    android:layout_below="@+id/linearLayout"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_margin="2dp"
    android:layout_marginTop="-11dp"
    android:layout_marginBottom="388dp"
    android:text="Answer"
    android:textSize="22sp"
    android:visibility="invisible" />

```

```

</RelativeLayout>

```

Activity_info.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/infoTRelativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/main_background"
    tools:context="com.example.flashpdr.Activity.InformationActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        >

        <ImageView
            android:id="@+id/appImageView"
            android:layout_width="170dp"
            android:layout_height="170dp"
            android:layout_gravity="center_horizontal"
            android:layout_marginLeft="50dp"
            android:layout_marginTop="50dp"
            android:layout_marginRight="50dp"
            android:layout_marginBottom="30dp"
            android:adjustViewBounds="true"
            android:background="#00000000"
            android:foregroundGravity="center"
            android:scaleType="fitCenter"
            android:src="@mipmap/ic_launcher" />

        <TextView
            android:id="@+id/textV"
            android:layout_width="410dp"
            android:layout_height="40dp"
            android:background="#BA9C27B0"
            android:gravity="center"
            android:text="Версія програми: 1.0 beta"
            android:textColor="@android:color/white"
```

```

        android:textSize="20dp"
        android:layout_marginBottom="30dp"/>

```

```

<TextView
    android:id="@+id/textP"
    android:layout_height="40dp"
    android:layout_width="410dp"
    android:background="#BA9C27B0"
    android:gravity="center"
    android:text="Розробник: Бейник Анастасія КНТ-120"
    android:textColor="@android:color/white"
    android:textSize="20dp"
    android:layout_marginBottom="30dp"/>

```

```

<TextView
    android:id="@+id/textD"
    android:layout_width="410dp"
    android:layout_height="39dp"
    android:background="#BA9C27B0"
    android:gravity="center"
    android:text="Дизйнер: Бейник Анастасія КНТ-120"
    android:textColor="@android:color/white"
    android:textSize="20dp"
    android:layout_marginBottom="30dp" />

```

```

</LinearLayout>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/infoRelativeLayout"
>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:gravity="center"
        android:background="@drawable/textview_design_registration_title"
        android:text="Інформація про додаток"
        android:textSize="25dp"
        android:textColor="#FFFFFF"
    />

    <ImageButton
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"
        android:src="@drawable/ic_arrow_back_white_48dp"
        android:background="#00000000"
        android:scaleType="fitCenter"
        android:adjustViewBounds="true"
        android:id="@+id/backImageButton"
    />
</RelativeLayout>

```

```

</LinearLayout>

```

```

</RelativeLayout>

```

```

</RelativeLayout>

```

Activity_login.xml

```

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

```

```

android:layout_height="match_parent"
android:background="@drawable/main_background"
tools:context="com.example.flashpdr.Activity.LoginActivity">

```

```

<RelativeLayout

```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/loginRelativeLayout"
>

```

```

<LinearLayout

```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>

```

```

<ImageView

```

```

    android:id="@+id/appImageView"
    android:layout_width="170dp"
    android:layout_height="170dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginBottom="30dp"
    android:adjustViewBounds="true"
    android:background="#00000000"
    android:foregroundGravity="center"
    android:scaleType="fitCenter"
    android:src="@mipmap/ic_launcher" />

```

```

<TextView

```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#BA9C27B0"
    android:gravity="center"
    android:text="Ім'я користувача"
    android:textColor="@android:color/white"
    android:textSize="20dp" />

```

```

<EditText

```

```

    android:id="@+id/usernameEditText"

```

```

android:layout_width="match_parent"
android:layout_height="40dp"
android:layout_marginLeft="30dp"
android:layout_marginTop="10dp"
android:layout_marginRight="30dp"
android:layout_marginBottom="10dp"
android:background="#BA9C27B0"
android:textColor="#FFFFFF"
android:textCursorDrawable="@drawable/cursor_design_color" />

```

```
<TextView
```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#BA9C27B0"
    android:gravity="center"
    android:text="Паполь"
    android:textColor="@android:color/white"
    android:textSize="20dp" />

```

```
<EditText
```

```

    android:id="@+id/passwordEditText"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="10dp"
    android:layout_marginRight="30dp"
    android:layout_marginBottom="10dp"
    android:background="#BA9C27B0"
    android:inputType="textPassword"
    android:textColor="#FFFFFF"
    android:textCursorDrawable="@drawable/cursor_design_color" />

```

```
<Button
```

```

    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:layout_marginTop="30dp"
    android:layout_marginBottom="10dp"
    android:background="@drawable/button_design_login"
    android:gravity="center"

```

```

        android:text="Авторизуватись"
        android:textSize="20dp"
        android:textColor="#FFFFFF"
        android:textAllCaps="false"
        android:id="@+id/loginButton"
    />

```

```

<TextView
    android:id="@+id/registrationTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:gravity="center"
    android:text="Зареєструватись"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="18dp"
    android:textStyle="bold" />

```

```

</LinearLayout>

```

```

</RelativeLayout>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/registrationRelativeLayout"
>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:gravity="center"

```

```

        android:background="@drawable/textview_design_registration_title"
        android:text="Рєєєтрацїя"
        android:textSize="25dp"
        android:textColor="#FFFFFF"
    />

```

```

<ImageButton
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_alignParentLeft="true"
    android:layout_centerVertical="true"
    android:src="@drawable/ic_arrow_back_white_48dp"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:adjustViewBounds="true"
    android:id="@+id/backImageButton"
/>

```

```

</RelativeLayout>

```

```

<ImageView
    android:id="@+id/appRegImageView"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="20dp"
    android:adjustViewBounds="true"
    android:background="#00000000"
    android:foregroundGravity="center"
    android:scaleType="fitCenter"
    android:src="@mipmap/ic_launcher" />

```

```

<EditText
    android:id="@+id/usernameRegEditText"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="10dp"
    android:layout_marginRight="30dp"

```

```

android:layout_marginBottom="10dp"
android:background="#BA9C27B0"
android:hint="Имя"
android:paddingLeft="15dp"
android:textColor="@android:color/white"
android:textColorHint="@color/design_default_color_background"
android:textCursorDrawable="@drawable/cursor_design_color" />

```

```
<EditText
```

```

    android:id="@+id/emailRegEditText"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="10dp"
    android:layout_marginRight="30dp"
    android:layout_marginBottom="10dp"
    android:background="#BA9C27B0"
    android:hint="Почта"
    android:paddingLeft="15dp"
    android:textColor="#FFFFFF"
    android:textColorHint="#FFFFFF"
    android:textCursorDrawable="@drawable/cursor_design_color" />

```

```
<EditText
```

```

    android:id="@+id/passwordRegEditText"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="10dp"
    android:layout_marginRight="30dp"
    android:layout_marginBottom="10dp"
    android:background="#BA9C27B0"
    android:hint="Пароль"
    android:inputType="textPassword"
    android:paddingLeft="15dp"
    android:textColor="#FFFFFF"
    android:textColorHint="#FFFFFF"
    android:textCursorDrawable="@drawable/cursor_design_color" />

```

```
<EditText
```

```

    android:id="@+id/retryPasswordRegEditText"

```

```

        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="10dp"
        android:layout_marginRight="30dp"
        android:layout_marginBottom="10dp"
        android:background="#BA9C27B0"
        android:hint="Повторіть пароль"
        android:inputType="textPassword"
        android:paddingLeft="15dp"
        android:textColor="#FFFFFF"
        android:textColorHint="#FFFFFF"
        android:textCursorDrawable="@drawable/cursor_design_color" />

```

```

<Button
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:layout_marginTop="30dp"
    android:layout_marginBottom="10dp"
    android:background="@drawable/button_design_login"
    android:gravity="center"
    android:text="Рєєєтрація"
    android:textSize="20dp"
    android:textColor="#FFFFFF"
    android:textAllCaps="false"
    android:id="@+id/registrationButton"
/>

```

```
</LinearLayout>
```

```
</RelativeLayout>
```

```
</RelativeLayout>
```

Activity_main.xml

```

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

```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/main_background"
tools:context=".Activity.MainActivity">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginTop="70dp"
    android:id="@+id/mainLinearLayout"
>

```

```

<Button
    android:id="@+id/testButton"
    android:layout_width="216dp"
    android:layout_height="207dp"
    android:layout_gravity="center_horizontal"
    android:layout_margin="15dp"
    android:background="@drawable/button_design_main_test"
    android:gravity="center"
    android:text="Тестування"
    android:textColor="@android:color/white"
    android:textSize="48dp" />

```

```

<Button
    android:layout_width="190dp"
    android:layout_height="190dp"
    android:background="@drawable/button_design_main_pdd"
    android:gravity="center"
    android:layout_gravity="center_horizontal"
    android:layout_margin="15dp"
    android:text="ТЕОPIЯ"
    android:textSize="42dp"
    android:textColor="@android:color/white"
    android:id="@+id/theoryButton"
/>

```

```

</LinearLayout>

```

```

<RelativeLayout

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/preferencesRelativeLayout"
    >

```

```

<ImageButton
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_margin="20dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:src="@drawable/ic_account_circle_white_48dp"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:adjustViewBounds="true"
    android:id="@+id/profileButton"
/>

```

```

<ImageButton
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_margin="20dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="false"
    android:src="@drawable/i"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:adjustViewBounds="true"
    android:id="@+id/informationButton"
/>

```

```

</RelativeLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#90000000"
    android:paddingTop="70dp"
    android:orientation="vertical"
    android:id="@+id/searchCoverLinearLayout"
>

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="#EEEEEE"
    android:textSize="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginBottom="10dp"
    android:id="@+id/coincidenceTextView"
/>

```

```

<ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/searchResultListView"
/>

```

```

</LinearLayout>

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:id="@+id/searchRelativeLayout"
>

```

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:background="@drawable/edittext_design_search"
    android:hint="Поиск..."
    android:textColor="@color/colorPrimaryDark"
    android:textColorHint="@color/colorPrimaryDark"
    android:id="@+id/searchEditText"
/>

```

```

<ImageButton
    android:layout_width="40dp"
    android:layout_height="40dp"

```

```

        android:layout_alignParentRight="true"
        android:src="@drawable/ic_close_white_48dp"
        android:background="#00000000"
        android:padding="10dp"
        android:scaleType="fitCenter"
        android:adjustViewBounds="true"
        android:id="@+id/deleteButton"
    />
</RelativeLayout>

```

```

<ProgressBar
    android:id="@+id/loading_spinner"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:indeterminateTintMode="src_atop"
    android:indeterminateTint="@android:color/white"
    android:layout_centerInParent="true"
/>

```

```

</RelativeLayout>

```

Ativity_information.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Activity.InformationActivity">

```

```

    <TextView
        android:id="@+id/testTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:gravity="center"
        android:text="InformationActivity"
        android:textColor="#424242"
        android:textSize="25dp" />

```

```

    <ListView
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:id="@+id/articlesListView"
        android:divider="@android:color/white"
    />

```

```
</LinearLayout>
```

Activity_pdr.xml

```

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

```

```
<ProgressBar
```

```

    android:id="@+id/loading_spinner"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:indeterminateTintMode="src_atop"
    android:indeterminateTint="@android:color/white"
    android:layout_centerInParent="true"
/>

```

```
<ListView
```

```

    android:id="@+id/listView"
    android:layout_width="407dp"
    android:layout_height="721dp"
    android:layout_alignParentStart="true"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:divider="#424242"
    android:dividerHeight="2px" />

```

```
</RelativeLayout>
```

Activity_profile.xml

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

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="0dp"
    android:background="@drawable/main_background"
    android:orientation="vertical"
    tools:context="com.example.flashpdr.Activity.ProfileActivity">

```

```

<ImageView
    android:id="@+id/carImageView"
    android:layout_width="match_parent"
    android:layout_height="180dp"
    android:layout_margin="0dp"
    android:adjustViewBounds="true"
    android:background="#00000000"
    android:padding="0dp"
    android:scaleType="fitCenter"
    android:src="@drawable/thumbs" />

```

```

<TextView
    android:id="@+id/usernameTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginBottom="-3dp"
    android:background="#BA9C27B0"
    android:backgroundTint="#BA9C27B0"
    android:gravity="center"
    android:text="Користувач"
    android:textColor="#FFFFFF"
    android:textSize="20dp" />

```

```

<TextView
    android:id="@+id/titleTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="-3dp"
    android:layout_marginRight="15dp"

```

```

        android:layout_marginBottom="-3dp"
        android:background="#BA9C27B0"
        android:gravity="center"
        android:text="Пімохід"
        android:textColor="@color/design_default_color_background"
        android:textSize="20dp" />

```

```

<TextView
    android:id="@+id/carTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="-3dp"
    android:layout_marginRight="15dp"
    android:background="#BA9C27B0"
    android:color="@color/colorAccent"
    android:gravity="center"
    android:text="Відсутнє"
    android:textColor="@color/colorAccent"
    android:textSize="20dp" />

```

```

<RelativeLayout
    android:layout_width="160dp"
    android:layout_height="100dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="20dp"
    android:background="@drawable/layout_design_score"
    android:backgroundTint="#BA9C27B0"
    android:padding="0dp">

```

```

<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentBottom="true"
    android:layout_marginLeft="12dp"
    android:layout_marginBottom="15dp"
    android:adjustViewBounds="true"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:src="@drawable/yellow_star" />

```

```

<ImageView
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginRight="12dp"
    android:layout_marginBottom="15dp"
    android:adjustViewBounds="true"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:src="@drawable/yellow_star" />

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_centerInParent="true"
    android:gravity="center"
    android:orientation="vertical"
    android:paddingTop="-8dp">

```

```

<TextView
    android:id="@+id/experienceTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="-8dp"
    android:layout_marginBottom="-4dp"
    android:gravity="center"
    android:text="0"
    android:textColor="#FFFFFF"
    android:textSize="40dp"
    android:textStyle="bold" />

```

```

<TextView
    android:id="@+id/scoreWordTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="-4dp"
    android:gravity="center"
    android:text="6a11B"
    android:textColor="#FFFFFF"

```

```
        android:textSize="20dp" />
```

```
    </LinearLayout>
```

```
</RelativeLayout>
```

```
<TextView
```

```
    android:id="@+id/experienceLeftTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="5dp"
    android:layout_marginRight="30dp"
    android:background="#BA9C27B0"
    android:gravity="center"
    android:text="Щоб стати пішоходом необхідно ще 0 балів"
    android:textColor="#FFFFFF"
    android:textSize="19dp" />
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
```

```
<LinearLayout
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="40dp"
    android:orientation="vertical"
    android:gravity="center"
    android:id="@+id/favouriteLinearLayout"
    >
```

```
<ImageButton
```

```
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src="@drawable/empty_star"
    android:background="#00000000"
    android:scaleType="fitCenter"
```

```

        android:adjustViewBounds="true"
    />

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Закладки"
    android:textColor="#FFFFFF"
    android:textSize="16dp" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="40dp"
    android:orientation="vertical"
    android:gravity="center"
    android:id="@+id/mistakesLinearLayout"
>

```

```

<ImageButton
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src="@drawable/pencil"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:adjustViewBounds="true"
    />

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Мои помилки"
    android:textColor="#FFFFFF"
    android:textSize="16dp" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:background="#d32f2f"
    android:gravity="center"
    android:orientation="horizontal"
    android:id="@+id/logoutLinearLayout"
>

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="2dp"
    android:gravity="center"
    android:text="Вийти з аккаунту"
    android:textColor="#FFFFFF"
    android:textSize="15dp" />

```

```

<ImageView
    android:layout_width="24dp"
    android:layout_height="24dp"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="3dp"
    android:layout_marginBottom="3dp"
    android:src="@drawable/logout"
    android:background="#00000000"
    android:scaleType="fitCenter"
    android:adjustViewBounds="true"
/>

```

```

</LinearLayout>

```

```

</RelativeLayout>

```

```

</LinearLayout>

```

Activity_article.xml

```

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

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".Activity.ShowTheArticleActivity">

```

```

<ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:divider="@android:color/white"
    android:id="@+id/articleListView"
/>

```

```

</LinearLayout>

```

Activity_theory.xml

```

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

```

```

<ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:divider="#424242"
    android:dividerHeight="2px"
    android:id="@+id/listView"
/>

```

```

</LinearLayout>

```

Activity_image.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Activity.TrafficSignsActivity">

    <ProgressBar
        android:id="@+id/loading_spinner"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:indeterminateTintMode="src_atop"
        android:indeterminateTint="@android:color/white"
        android:layout_centerInParent="true"
    />

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="10dp"
        android:divider="#424242"
        android:dividerHeight="2px"
        android:id="@+id/listView"
    />

</RelativeLayout>
```

ДОДАТОК В
Слайди презентації

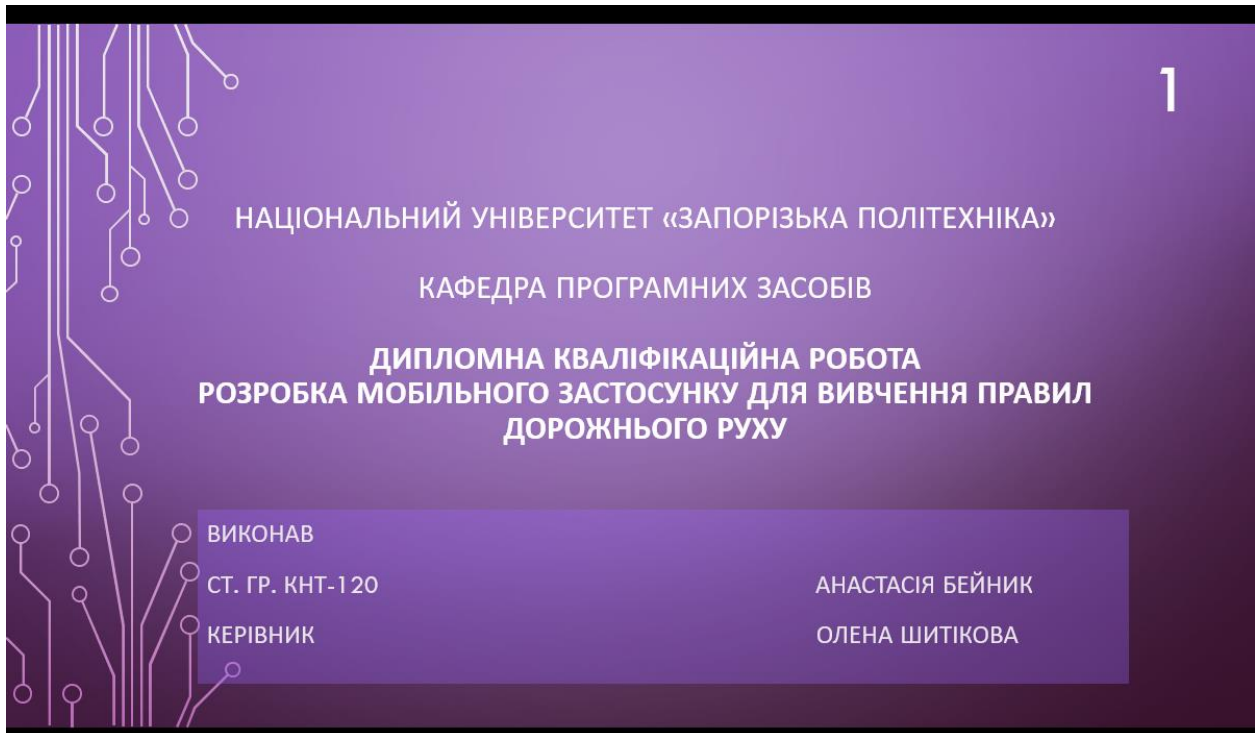


Рисунок В.1 – 1 слайд

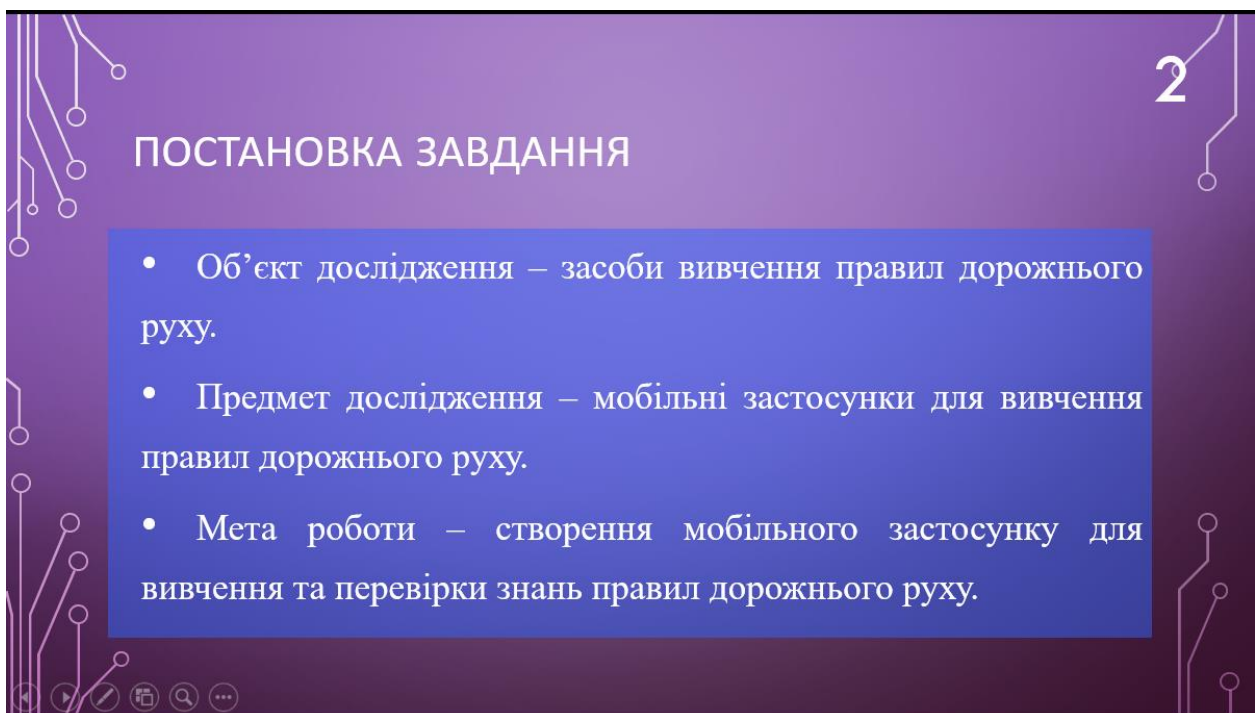


Рисунок В.2 – 2 слайд

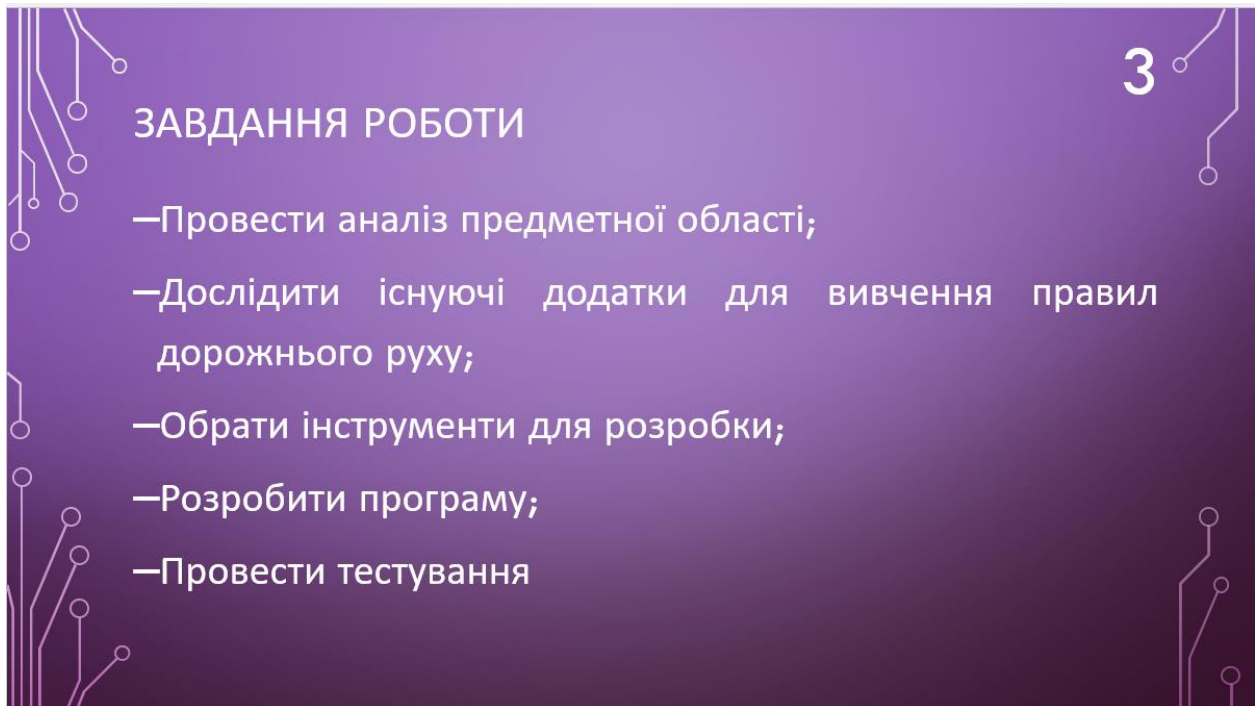


Рисунок В.3 – 3 слайд

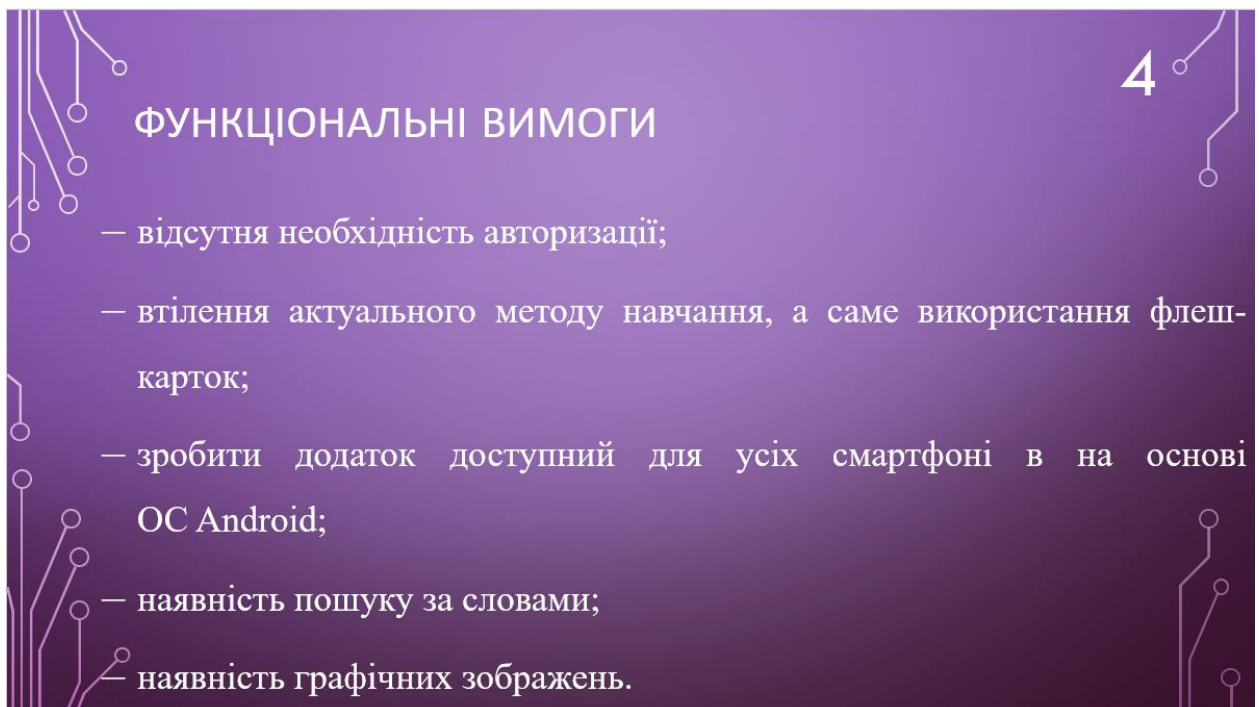


Рисунок В.4 – 4 слайд

ВИБІР МОВИ ПРОГРАМУВАННЯ			5
Параметр	Java	Kotlin	
Performance	Добре оптимізована продуктивність на платформі Android завдяки Android Runtime (ART) і Just-In-Time (JIT) компіляції.	Висока продуктивність, схожа на Java.	
Development Speed and Ease	Висока продуктивність на платформі Android завдяки Android Runtime (ART) і Just-In-Time (JIT) компіляції.	Підвищує продуктивність завдяки компактному синтаксису, зменшенню boilerplate коду і захисту від помилок, пов'язаних з null.	
Community and Ecosystem	Величезна екосистема, численні бібліотеки, фреймворки та інструменти.	Зростаюча екосистема і спільнота, але не така велика, як у Java.	
Interoperability	Безшовна інтеграція з існуючими Android проектами і бібліотеками.	Повна сумісність з Java, дозволяє поступову міграцію з Java на Kotlin.	
Learning Curve	Велика кількість розробників знайомі з Java завдяки її поширеності в освітніх закладах і індустрії, що зменшує криву навчання і прискорює онбординг.	Потребує навчання для розробників, які звикли до Java, але синтаксис схожий і логічний.	
Advantages	Стабільність, зрілість, широка підтримка, висока продуктивність, сумісність з Android SDK, великий обсяг документації та підтримки спільноти.	Сучасний синтаксис, зменшення коду, захист від помилок, сумісність з Java.	
Disadvantages	Вербозність, що може призводити до більшої кількості boilerplate коду.	Менша екосистема та спільнота в порівнянні з Java, потребує навчання для Java-розробників.	

Рисунок В.5 – 5 слайд

ВИБІР ІНТЕГРОВАНОГО СЕРЕДОВИЩА РОЗРОБКИ				6
Параметр	Android Studio	IntelliJ IDEA	Eclipse	
Основний фокус	Спеціалізоване середовище для розробки Android-додатків.	Універсальне середовище для розробки з підтримкою Android, веб, серверних додатків та багато іншого.	Універсальне середовище для розробки з широкою підтримкою різних мов та платформ.	
Підтримка Android	Повна підтримка Android SDK, інтегровані інструменти для розробки, налагодження та тестування Android-додатків.	Повна підтримка Android SDK, інтеграція з Android Studio плагіном для повного досвіду розробки.	Підтримка Android через плагіни, але не така інтегрована як у Android Studio або IntelliJ IDEA.	
Продуктивність і Швидкість	Оптимізована для розробки Android, але може бути повільною на великих проектах або на машинах з низькою продуктивністю.	Висока продуктивність, особливо у версії Ultimate, менша вимогливість до ресурсів у порівнянні з Android Studio.	Може бути менш продуктивним і швидким через потребу встановлення численних плагінів для повної підтримки Android.	
Функціональність	Вбудовані інструменти для розробки, налагодження, профілювання, тестування та емулявання Android-додатків.	Усі функції Android Studio, плюс додаткові інструменти для інших мов та платформ, розширена функціональність IDE.	Широка підтримка мов програмування та платформ, але потребує більше налаштування для Android.	

Рисунок В.6 – 6 слайд

7

ВИБІР СИСТЕМИ АВТОМАТИЧНОГО СКЛАДАННЯ

Характеристика	Maven	Gradle
API	+	+, працює швидше
Імплементація залежностей	+	+, працює швидше
Безпечність одночасного використання кешу	+	+, працює швидше
Правила заміни для сумісних бібліотек	-	+
Застосування правил ReplacedBy	-	+
Розширена роздільна здатність метаданих	-	+
Можливість динамічної заміни залежностей проєкту на зовнішні	-	+

Рисунок В.7 – 7 слайд



Рисунок В.8 – 8 слайд

9

ДІАГРАМА ПРЕЦЕДЕНТІВ

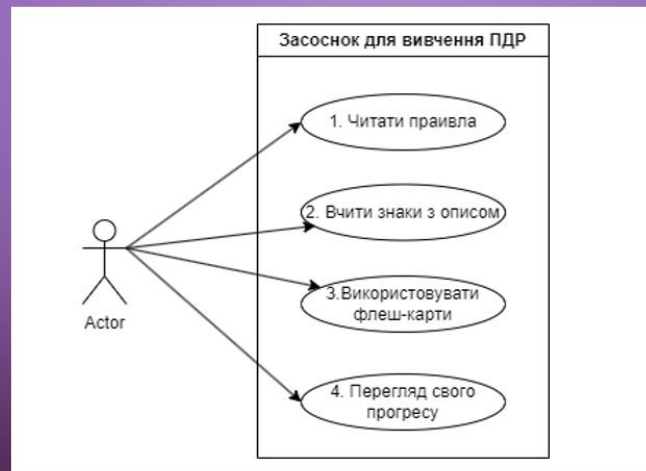


Рисунок В.9 – 9 слайд

10

ФУНКЦІОНАЛЬНА СТРУКТУРА ПРОГРАМИ

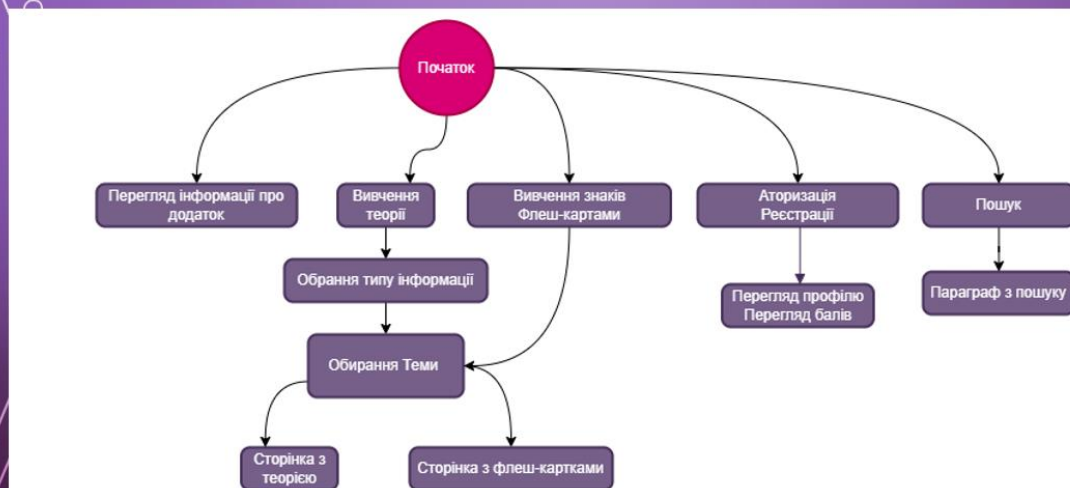


Рисунок В.10 – 10 слайд



Рисунок В.11 – 11 слайд

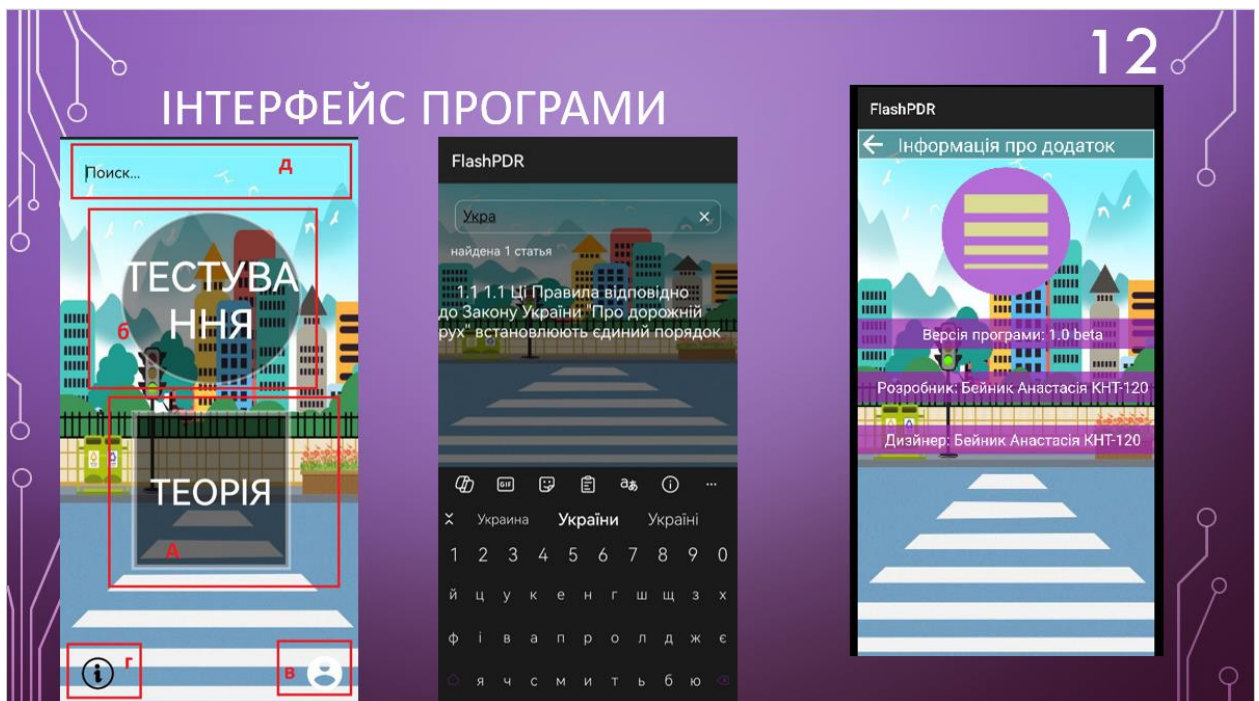


Рисунок В.12 – 12 слайд

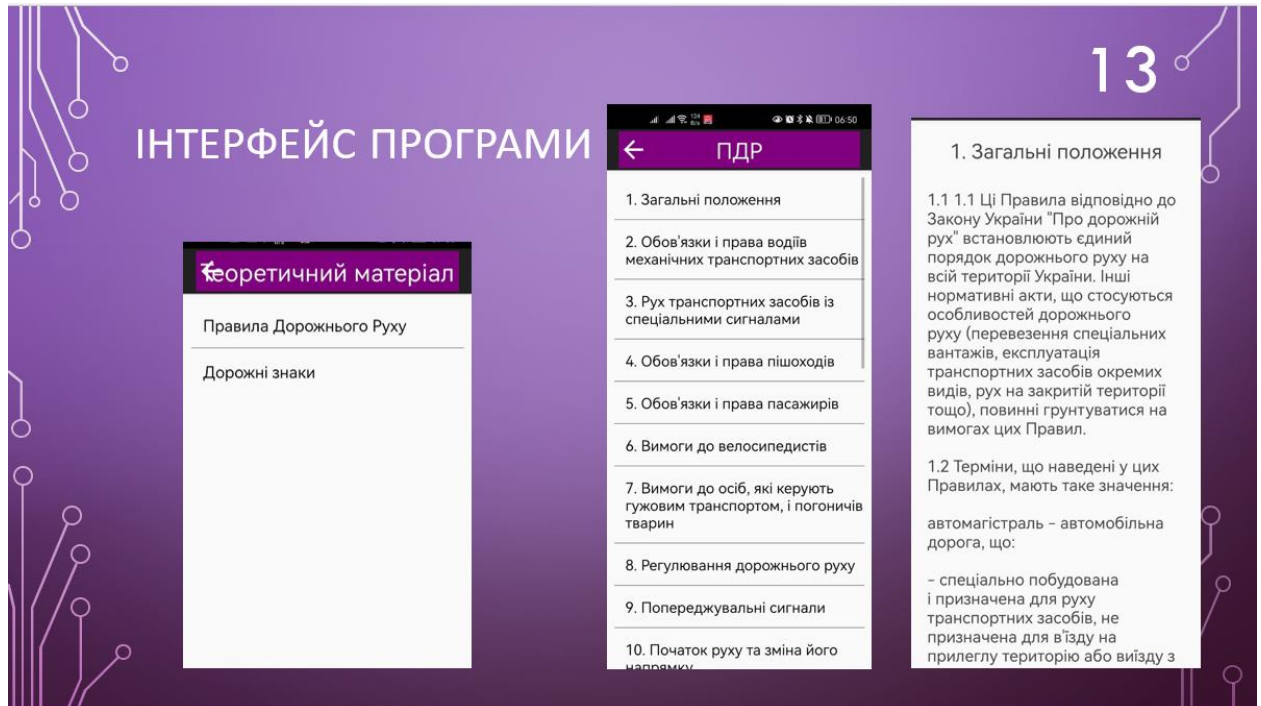


Рисунок В.13 – 13 слайд

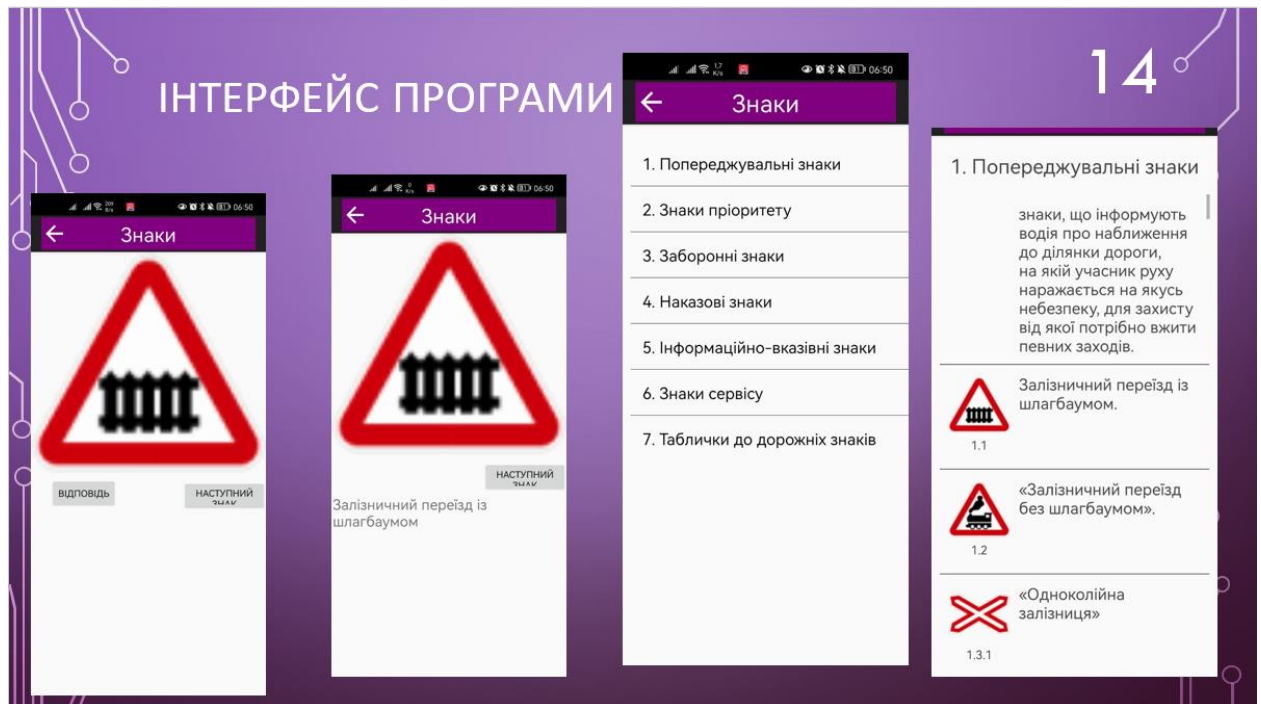


Рисунок В.14 – 14 слайд

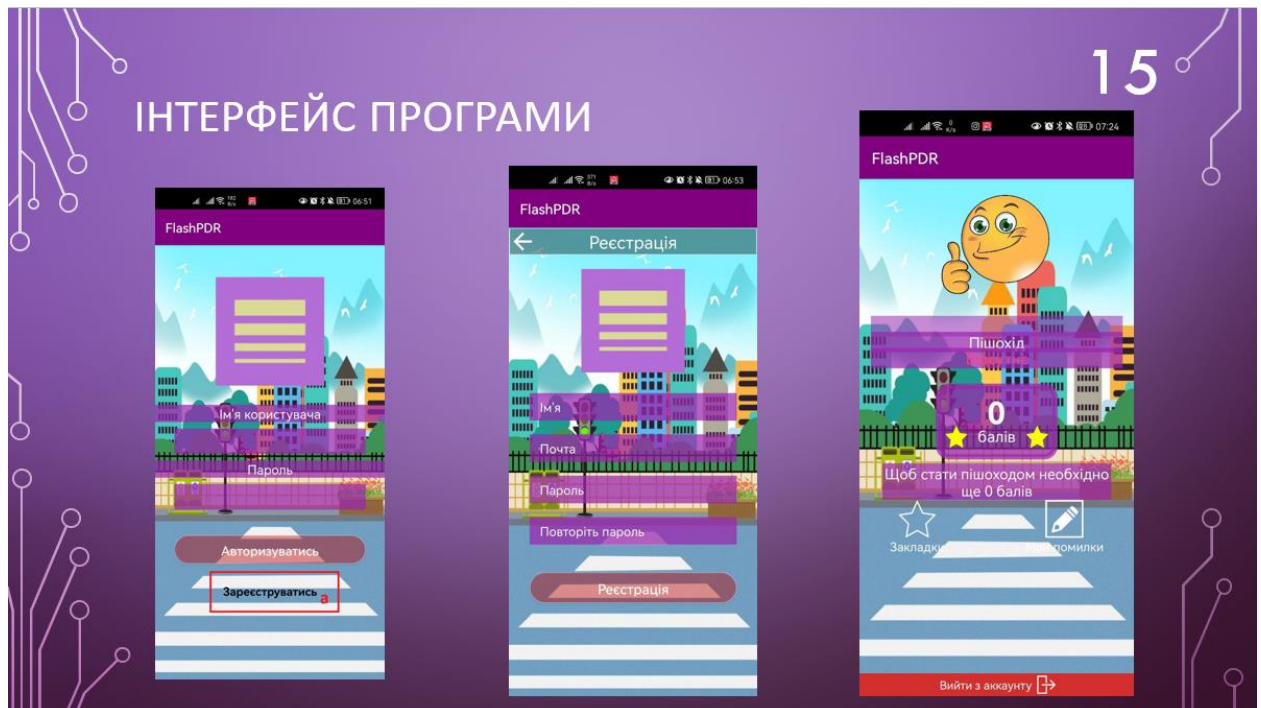


Рисунок В.15 – 15 слайд

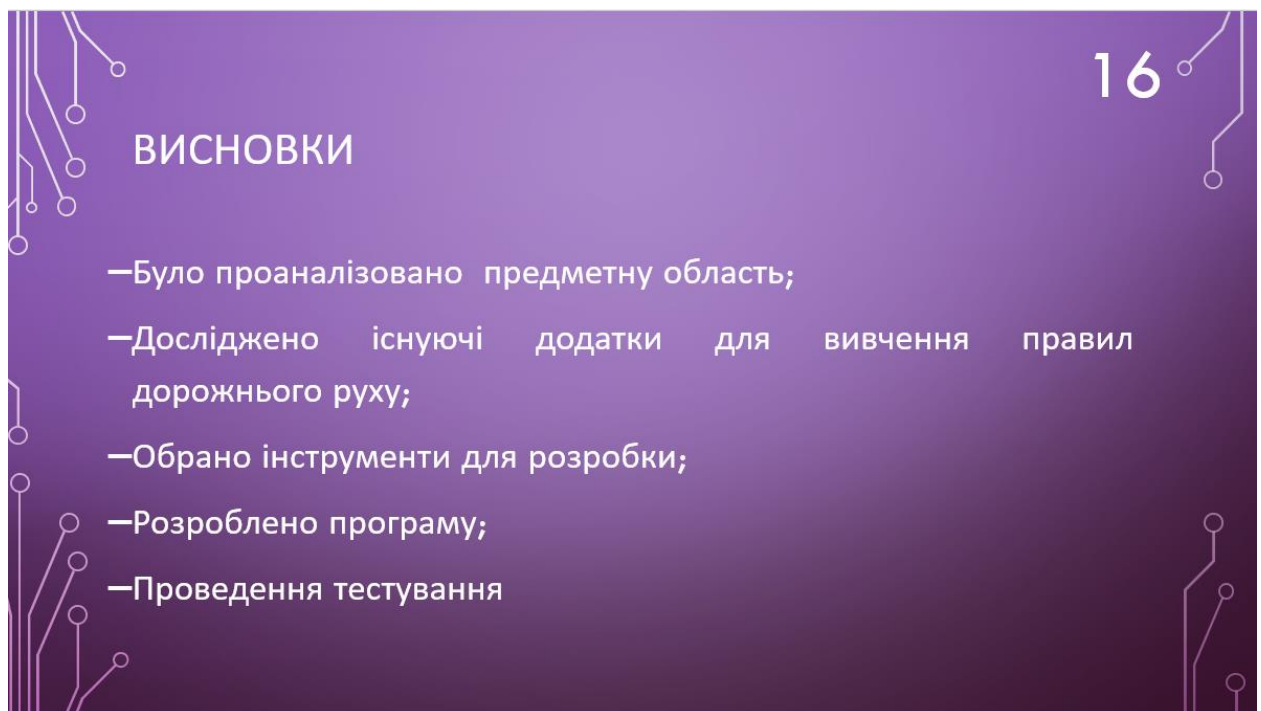


Рисунок В.16 – 16 слайд