

Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка інформаційно-розважального мобільного додатку для туристів»

Студентки 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

**Нікітенко Аліни
Євгеніївни**

підпис студента

Науковий керівник
кандидат технічних наук,
професор

**Пашорін Валерій
Іванович**

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент

**Цензура Микола
Олександрович**

підпис керівника

КИЇВ – 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки
Криворучко О.В.
"20" жовтня 2020 р.

Завдання

на випускню кваліфікаційну роботу студентів

Нікітенко Аліні Євгеніївні

1.Тема випускної кваліфікаційної роботи «Розробка інформаційно-розважального мобільного додатку для туристів»

Затверджена наказом ректора від «30» жовтня 2020 р. №3225

2.Строк здачі студентом закінченої роботи _____

3.Цільова установка та вихідні дані до роботи

Мета роботи: метою створення даної дипломної роботи є подальше застосування опрацьованого матеріалу для розширення кількості потенційних користувачів, та розробка даного додатку в інших середовищах.

Об'єкт дослідження: Android додаток, який буде надавати клієнту оптимальні інформаційно-розважальні місця на основі попередньо зібраних даних від користувачів додатку

Предметом дослідження є мобільні Android-додатки

4.Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5.Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧІ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОГО ДОДАТОКА ДЛЯ ТУРИСТІВ

1.1 Поняття застосунку

1.1.2 Операційна система Андроїд

1.1.3 Мобільний застосунок

1.2 Аналіз предметної області системи

1.2.1 Завдання та проблеми, які вирішує система

1.3 Дослідження існуючих застосунків

1.3.1 Додаток «Walqlike»

1.3.2 Додаток «Restaurant Guru»

1.3.3 Порівняння наявних систем

1.4 Завдання застосунку

1.5 Постановка задачі

1.5.1 Функціональні вимоги

1.5.2 Не функціональні вимоги

1.5.3 Обмеження задачі

1.6 Особливості реалізації додатку

1.6.1 Варіанти використання додатку

ВИСНОВКИ ДО РОЗДІЛУ 1

РОЗДІЛ 2

ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СПРОЩЕННЯ ОЗНАЙОМЛЕННЯ ТА РОЗМІЩЕННЯ В МІСТІ

2.1 Вибір технологій та засобів для написання додатку

2.1.1 Вибір мови програмування

2.1.2 Вибір середовища для розробки

2.2 Структура застосунку

2.3 Проєктування пакетів та класів

2.3.1 Проєктування пакетів

2.3.2 Проєктування класів

2.4 Проектування додатку

2.4.1 Сценарій створення облікового запису в додатку

2.4.2 Сценарій обрання закладу та побудова маршруту

2.4.3 Сценарій проходження квесту

ВИСНОВКИ ДО РОЗДІЛУ 2

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ

3.1 Результати розробки пакетів

3.2 Результати розробки класів

3.2.1 Класи пакету activity

3.2.2 Класи пакету adapter

3.2.3 Класи пакету fragment

3.3 Розробка інтерфейсу користувача

3.3.1 Екран входу

3.3.2 Екран головного меню

3.3.3 Екран обраної певної категорії

ВИСНОВКИ ДО РОЗДІЛУ 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	Вибір теми випускної кваліфікаційної роботи	21.09.2020	21.09.2020
2.	Вступ та перелік літературних джерел	14.12.2020	14.12.2020
3.	Розділ 1.«ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ СОЦІАЛЬНОЇ МЕРЕЖІ»	19.02.2021	19.02.2021
4.	Розділ 2.«МОДЕЛЮВАННЯ ТА АНАЛІЗ РОЗРОБКИ СОЦІАЛЬНОЇ МЕРЕЖІ»	03.03.2021	03.03.2021
5.	Розділ 3.«РОЗРОБКА ВЕБ-ДОДАТКУ СОЦІАЛЬНОЇ МЕРЕЖІ»	10.04.2021	5.04.2021
6.	Висновки	24.04.2021	23.04.2021
7.	Здача випускної кваліфікаційної роботи на кафедру(перша перевірка)	14.05.2021	14.05.2021
8.	Підготовка автореферату та презентація доповіді	17.05.2021	17.05.2021
9.	Попередній захист випускної кваліфікаційної роботи	18.05.2021-21.05.2021	21.05.2021
10.	Зовнішнє рецензування випускної кваліфікаційної роботи	01.06.2021	01.06.2021
11.	Здача прожитої випускної кваліфікаційної роботи на кафедру	02.06.2021	02.06.2021
12.	Публічний захист випускної кваліфікаційної роботи		

7. Дата видачі завдання «20» жовтня 2020 р.

8. Науковий керівник випускної кваліфікаційної роботи

Пашорін В.І.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

Цензура М. О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

Нікітенко А. Є.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист Пашорін В. І.
(ПІБ, підпис, дата)

Випускню кваліфікаційну роботу _____ Нікітенко А.Є.
(прізвище, ініціали)

Гарант освітньої програми _____ Цензура М. О.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(прізвище, ініціали, підпис)

« » 20 p.

АНОТАЦІЯ

Кваліфікаційна робота бакалавра, с., рис., табл., джерел, 1 додаток.

Об'єктом розробки був мобільний застосунок, а саме інформаційний додаток зі списками закладів та інтерактивним квестор по місту.

Додаток розроблявся для операційної системи Android, з використанням мови програмування Java та технологій Android SDK. Розробка проводилася в середовищі розробки Android Studio 4.1.0.0.

Метою даної роботи є створення застосунку, для того щоб туристам та жителям міста було простіше знайти потрібний заклад та ознайомитись з основними пам'ятками. Для кожного закладу буде наданий номер телефону, категорія, та час робот. Завдяки пункту Квест, люди зможуть пройти невеликий маршрут з інформацією про місця та невеликим інтерактивом.

Результатом роботи є реалізація таких можливостей додатку:

- Надання списку закладів міста та інформація про них;
- Розділення закладів по категоріях;
- Побудова маршруту на карті до обраного закладу;
- Проходження невеликого пізнавального квесту;
- Функції реєстрації та входу користувача, а також можливість зміни паролю;

Подальший розвиток можливий в сторону розширення списку закладів, створення нових маршрутів, покращення інтерфейсу та розробка серверної частини.

Робота має практичну цінність. Розрахунок економічної цінності не проводиться.

ANDROID SDK, JAVA, API, АНДРОІД ДОДАТОК, FIREBASE

ANNOTATION

Qualifying work of the bachelor, s., Fig., Tab., Sources, 1 appendix.

The object of development was a mobile application, namely an information application with lists of institutions and an interactive conservator in the city.

The application was developed for the Android operating system, using the Java programming language and Android SDK technologies. The development was carried out in the development environment of Android Studio 4.1.0.0.

The purpose of this work is to create an application to make it easier for tourists and residents to find the right place and get acquainted with the main attractions. A telephone number, category, and robot time will be provided for each facility. Thanks to the Quest item, people will be able to walk a short route with information about places and a little interactivity.

The result is the implementation of the following features of the application:

- Providing a list of institutions of the city and information about them;
- Division of institutions into categories;
- Construction of the route on the map to the selected institution;
- Passing a small cognitive quest;
- User registration and login functions, as well as the ability to change the password;

Further development is possible in the direction of expanding the list of establishments, creating new routes, improving the interface and developing the server part.

The work has practical value. The calculation of economic value is not performed.

ANDROID SDK, JAVA, API, ANDROID APP, FIREBASE

					KHTEY 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		3

ЗМІСТ

ВСТУП -----	2
АНАЛІЗ ЗАДАЧІ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОГО ДОДАТОКА ДЛЯ ТУРИСТІВ -----	3
1.1 Поняття застосунку -----	3
1.1.2 Операційна система Андроїд -----	3
1.1.3 Мобільний застосунок -----	5
1.2 Аналіз предметної області системи -----	6
1.2.1 Завдання та проблеми, які вирішує система -----	6
1.3 Дослідження існуючих застосунків -----	7
1.3.1 Додаток «Walqlike» -----	7
1.3.2 Додаток «Restaurant Guru» -----	10
1.3.3 Порівняння наявних систем -----	12
1.4 Завдання застосунку -----	13
1.5.1 Функціональні вимоги -----	14
1.5.2 Не функціональні вимоги -----	14
1.5.3 Обмеження задачі -----	14
1.6 Особливості реалізації додатку -----	14
1.6.1 Варіанти використання додатку -----	14
Висновки до розділу 1 -----	16
РОЗДІЛ 2 -----	17
ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СПРОЩЕННЯ ОЗНАЙОМЛЕННЯ ТА РОЗМІЩЕННЯ В МІСТІ -----	17
2.1 Вибір технологій та засобів для написання додатку -----	17
2.1.1 Вибір мови програмування -----	17
2.1.2 Вибір середовища для розробки -----	19
2.2 Структура застосунку -----	22
2.3 Проєктування пакетів та класів -----	23
2.3.1 Проєктування пакетів -----	23
2.3.2 Проєктування класів -----	23
2.4 Проєктування додатку -----	26
2.4.1 Сценарій створення облікового запису в додатку -----	26
2.4.2 Сценарій обрання закладу та побудова маршруту -----	28
2.4.3 Сценарій проходження квесту -----	31

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		14.12.20	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		14.12.20		В	2	52
Гарант		Цензура М.О.		14.12.20		Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		14.12.20				
					Вступ			

РОЗДІЛ 3 -----	35
РОЗРОБКА ДОДАТКУ -----	35
3.1 Результати розробки пакетів -----	35
3.2 Результати розробки класів -----	36
3.2.1 Класи пакету activity -----	36
3.2.2 Класи пакету adapter -----	40
3.2.3 Класи пакету fragment -----	41
3.3 Розробка інтерфейсу користувача -----	42
3.3.1 Екран входу -----	42
3.3.2 Екран головного меню -----	44
3.3.3 Екран обраної певної категорії -----	46
Висновки до розділу 3 -----	47
ВИСНОВКИ ТА ПРОПОЗИЦІЇ -----	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ -----	49
ДОДАТКИ -----	51
Додаток А -----	51

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		3

ВСТУП

Мобільний додаток – це програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних застосунків встановлені на самому пристрої або можуть бути завантажені на нього з онлайн магазинів мобільних застосунків. Вони застосовуються для поліпшення чи прискорення тих або інших задач.

Об'єктом розробки є Android додаток, який буде надавати клієнту оптимальні інформаційно-розважальні місця на основі попередньо зібраних даних від користувачів додатку. Додаток буде взаємодіяти з головним сервісом, який саме і буде займатися розрахунками підходящих місць, цей сервіс, в свою чергу, буде взаємодіяти з іншими сервісами, які будуть виступати в ролі допоміжних. Дана розробка буде актуальною для користувачів Android-додатків.

Android додаток дозволить повністю відмовитися від потреби розробки унікальних пристроїв, які будуть виконувати лише специфічні функції, це дає можливість повторного використання даного проекту по всьому світі та в інших галузях без необхідності реалізації індивідуальних апаратних та програмних продуктів.

Метою створення даної дипломної роботи є подальше застосування опрацьованого матеріалу для розширення кількості потенційних користувачів, та розробка даного додатку в інших середовищах.

Завданням є власне створення актуального, змістовного Android-додатку, що буде успішно використаний у майбутньому.

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		14.12.20	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		14.12.20		В	2	52
Гарант		Цензура М.О.		14.12.20		Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		14.12.20				
					Вступ			

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧІ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОГО ДОДАТОКА ДЛЯ ТУРИСТІВ

1.1 Поняття застосунку

Мобільний застосунок – програмне забезпечення, яке використовується на смартфонах та інших мобільних пристроях. Більша частина пристроїв продаються з вже встановленим комплектом програм. В наш час, розробник може залучати у свій проєкт функції попередньо встановлених застосунків. Є можливість додати Google maps, календар, браузер або поштовий клієнт.

Це значно спрощує процес розробки власних додатків і спрощує адаптацію інформаційних продуктів під певні потреби замовника.

1.1.2 Операційна система Андроїд

Андроїд – операційна система створена компанією Google на базі ядра Linux. Представлена платформа використовується для мобільних телефонів та планшетних комп'ютерів. Не зважаючи на ядро Linux, всі застосунки спираються на мову програмування Java та її віртуальну машину. Великим плюсом даної операційної системи є її розповсюдженість. На цей момент, у світі, з усіх мобільних пристроїв 84% з такою ОС.

Плюси операційної системи Андроїд:

					КНТЕУ 121 07-15.БР				
Зм	Аркуш	№ докум	Підпис	Дата					
Зав.каф.		Криворучко О.В.			«Розробка інформаційно-розважального мобільного додатку для туристів»		Стадія	Аркуш	Аркушів
Керівник		Паширін В.І.					Р1	3	52
Гарант		Цензура М.О.					Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.							
					Аналіз задачі інформаційного-розважального додатку для туристів				

- велика розповсюдженість на мобільних пристроях ;
- легке пристосування до двовимірної та тривимірної графіки та використання VGA, такими можливостями можна створювати гарні та зрозумілі інтерфейси, а також цю властивість можна використовувати в різних галузях виробництва;
- легка в використанні база даних SQLite, вона дає користувачу можливість, отримати доступ до інформації навіть в середовищі без будь-якого зв'язку, виконавши попереднє кешування;
- підтримує багато технологій для зв'язку та велику кількість форматів для аудіо та відео даних;
- має підтримку нестандартного обладнання, такого як камера, компас, акселерометр, GPS. У своїх застосунках є можливість об'єднувати перелічені пристрої;
- зручне і адаптоване середовище розробки додатків під Андроїд платформу. Android Studio і в доповнення емулятор пристроїв, можливості відладження та тестування на швидкодію;

Мінуси операційної системи Андроїд:

- можливості ядра Linux практично не використовуються через базування на мові Java;
- ваш за стосунок може бути в будь-який час видалений з систем користувачів, якщо адміністрація Google вирішить, що їх правила порушені;
- велика вразливість до зломів через відкритість ОС;
- оновлення версії платформи більш залежить від розробника пристрою, ніж від розробників операційної системи.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		4

1.1.3 Мобільний застосунок

Мобільний застосунок – це програмне забезпечення розроблене під певні функціональні можливості сучасних гаджетів. Наявні додатки призначені для спрощення роботи з сервісами, для магазинів, для розваг, для вдосконалення процесу виробництва та багато іншого. Використовувати ПО можна з доступом до мережі інтернет, або без доступу, але головна умова, щоб програма була встановлена на пристрій. Використовуючи зрозумілий користувачам інтерфейс, за стосунок стає простим у використанні та багаторазовим. А за допомогою простих технологій представлених в пристрої, можна забезпечити широкий функціонал.

Плюси мобільних застосунків:

- розробник може використати велику кількість обладнання вбудованого в пристрій (навігація, камера, компас, акселерометр, датчик освітлення);
- є можливість зберігати багато інформації на серверах, та на самому пристрої користувача;
- можливе виконання складних математичних обчислень без значних затрат часу;
- можуть виконуватись в фоновому режимі, не заважаючи іншій діяльності користувача;

Мінуси мобільних застосунків:

- забезпечення певного функціонала залежить від вимог певної платформи;

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		5

- дефіцит пам'яті на пристроях, через те, що більшість користувачів намагається встановити якомога більше додатків;
- через збільшення популярності мобільних застосунків, на їх ринку збільшується конкуренція;
- при розробці одразу під декілька платформ, можливі значні фінансові затрати;

1.2 Аналіз предметної області системи

1.2.1 Завдання та проблеми, які вирішує система

Android додаток повинен бути простий і зрозумілий у використанні для туристів, мати user – friendly інтерфейс а також, він повинен виконувати поставлені до нього задачі по створенню та оптимізації маршрутів мандрівок без затримок та без блокування у разі використання його іншим користувачем, що може статися в ситуації конкурентного доступу багатьох клієнтів одночасно. Додаток має бути досить простим у використанні та автономним цілтю використання лише одним непідготовленим користувачем – туристом.

Даний додаток представляє собою сукупність із систем клієнт – серверної взаємодії між пристроєм, на платформі Android, та віддаленим сервісом який буде отримувати і певним чином реагувати на запити з пристроїв мандрівників.

Безпосередньо Android додаток повинен забезпечувати наступні функції:

- Відображення всіх можливих місць для відвідування, для яких може:

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		6

– Можливість додавання користувачем нового місця до бази даних;

– Можливість додавання коментаря щодо місця для відвідування;

– Можливість збереження відвідуваних місць;

– Можливість поділитися враженнями з іншими користувачами;

Сервіс, який надає інформацію про оптимальні маршрути для Android додатку включає в себе функції:

– Збереження даних про відвідувані місця;

– Надання даних про місця для відвідування;

– Аутентифікація користувачів та адміністраторів;

Сервіс розроблено з метою подальшого масштабування проекту, відокремлення клієнтської частини від серверної та підвищення якості, в даному випадку швидкості та точності пошуку інформаційно-розважальних місць.

Сервіс повинен швидко реагувати та відповідати на запити, використовуючи маленькі за розміром структури даних у відповіді. Також, сервіс не повинен мати постійного стану, і тим самим, бути захищеним від загублення даних в разі відмови одного з вузлів системи.

1.3 Дослідження існуючих застосунків

1.3.1 Додаток «Walqlike»

Додаток розробив Stanislav Okolot. Він призначений для будування квест маршрутів для туристів.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		7

В додатку є такі функції:

- створення квест маршруту;
- ознайомлення з історією міста;
- розгадування загадок;

На рисунку 1.1 представлено знімок початкового екрана застосунку.

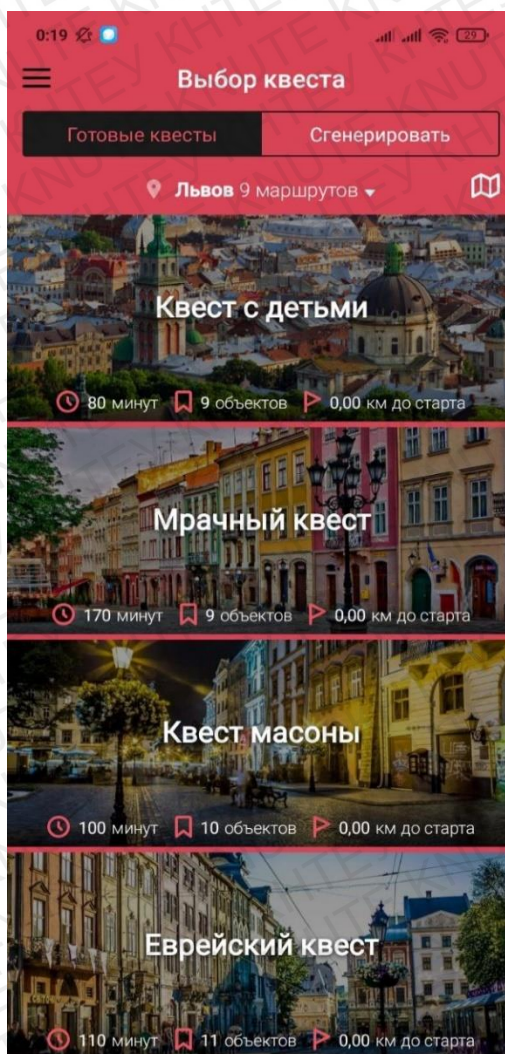


Рис. 1.1 – Знімок початкового екрана

На рисунку 1.2 та 1.3 представлені екрани ознайомлення з інформацією, та загадка відповідно.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		8

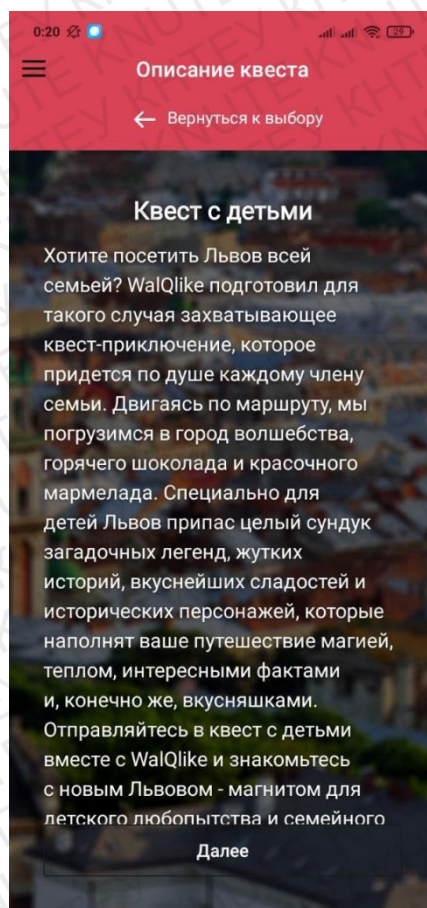


Рис. 1.2 – Экран ознайомлення з інформацією про місто

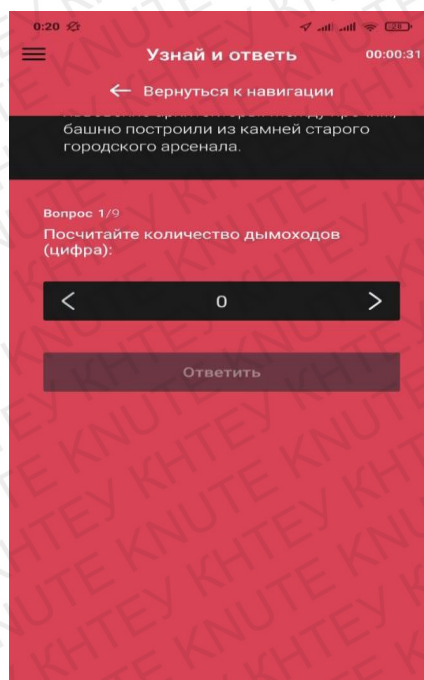


Рис. 1.3 – Экран питання від додатку

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		9

1.3.2 Додаток «Restaurant Guru»

Додаток розроблений SoftDeluxe. Його призначення – зберігання інформації про заклади харчування.

Функціонал додатку:

- перегляд представлених закладів;
- перегляд інформації про заклади;
- перегляд відгуків про заклад;

На рисунку 1.4 представлено екран перегляд категорій закладів

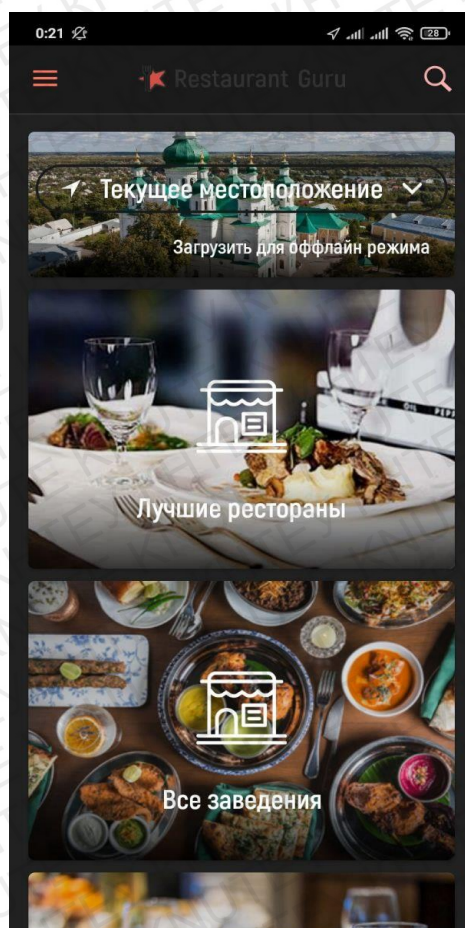


Рис. 1.4 – Экран просмотра категорий закладів

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		10

На рисунку 1.5 показано екран переліку закладів

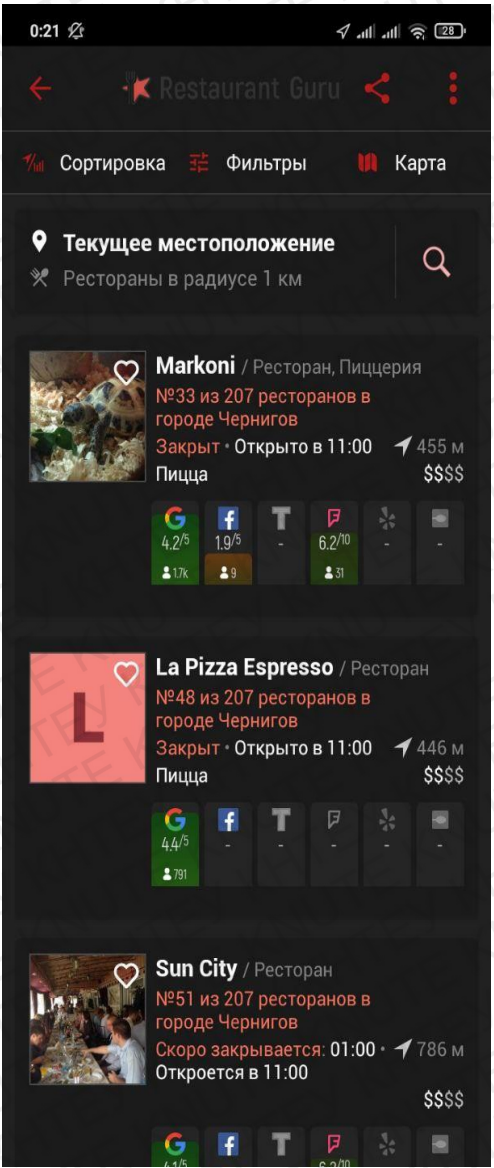


Рис. 1.6 – Экран переліку закладів

1.3.3 Порівняння наявних систем

Наявні додатки можуть виконати тільки частину функцій представлених в нашому проєкті. Для їх порівняння виділяються певні критерії:

- можливість дізнаватись інформацію про місто;
- можливість перегляду туристичного маршруту;
- можливість перегляду списку закладів міста;
- можливість перегляду інформації про заклади харчування;
- можливість побудови маршруту до певного закладу;

Використовуючи ці критерії можна зробити порівняння альтернативних додатків та нашого застосунку. В таблиці 1.1 представлена порівняльна характеристика програм.

Таблиця 1.1

Порівняльна характеристика застосунків

Критерій	Walqlike	Restaurant Guru	Інформаційно- розважальний додаток для туристів
Можливість дізнатись інформацію про місто	+	-	+
Можливість перегляду туристичного маршруту	+	-	+
Можливість перегляду списку закладів	-	+	+
Можливість перегляду інформації про заклади	-	+	+
Можливість побудови маршруту до закладу	-	+	+

В розробленому додатку будуть враховані всі критерії, які використовувались для порівняння з аналогами.

1.4 Завдання застосунку

Розроблений додаток має мати зрозумій та простий інтерфейс, для зручності користування.

Основні задачі додатку:

- реєстрація користувача;
- збір інформації про користувача;
- представлення списку закладів корисних для туристів;
- можливість побудувати маршрут до обраного закладу;
- створення квесту по місту в доповненні до інформації про пам'ятки культури;

Переглянувши всі задачі можна створити діаграму можливого використання додатку користувачем.

Діаграма можливих дій користувача представлена на рисунку 1.6.



Рис. 1.6 – Діаграма можливих дій користувача

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		14.12.20	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		14.12.20		В	13	52
Гарант		Цензура М.О.		14.12.20		Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		14.12.20				
					Вступ			

1.5 Постановка задачі

1.5.1 Функціональні вимоги

Дослідивши завдання застосунку, можна виділити такі вимоги:

- Простий та зрозумілий інтерфейс;
- Додаток має зберігати та відображати всі данні про заклади;
- Можливість побудови маршруту до обраного закладу;
- Можливість проходження квесту;

1.5.2 Не функціональні вимоги

Дослідивши поставлені завдання можна виділи деякі не функціональні вимоги:

- Функціональне та логічне розміщення елементів на сторінці застосунку;
- Задовільна швидкість для роботи застосунку;

1.5.3 Обмеження задачі

При створенні додатку потрібно керуватись певними обмеженнями:

- Використання середовища інтегрованої розробки AndroidStudio;
- Працездатність на пристроях з API 19 та вище;

1.6 Особливості реалізації додатку

1.6.1 Варіанти використання додатку

Додаток можна використовувати декількома способами. В системі буде представлено:

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		14

- Список закладів корисних для туристів;
- Прокладання маршруту до обраного закладу;
- Можливість проходження квест маршруту;

Після авторизації користувач бачить головне меню для обрання потрібної йому функції.

Головне меню додатку представлено на рисунку 1.7



Рис. 1.7 – Головне меню додатку

Для зручності використання додатку всі категорії закладів будуть поділені на підкатегорії. Для зміни під категорії буде достатньо зсунути поточний екран або обрати потрібну категорії в верхній частині екрану.

На рисунку 1.8 представлено вигляд екрану при обраній категорії закладів.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		15



Рис. 1.8 – Вигляд екрану при обраній категорії «Їжа»

Висновки до розділу 1

Представлений додаток орієнтований на місто Чернігів. В майбутньому для покращення та розвитку можливе створення нових маршрутів та додавання більшого асортименту закладів.

При невеликих змінах додатку, його можна адаптувати під інші міста України та активно використовувати в туристичній сфері.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		16

РОЗДІЛ 2

ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ СПРОЩЕННЯ ОЗНАЙОМЛЕННЯ ТА РОЗМІЩЕННЯ В МІСТІ

2.1 Вибір технологій та засобів для написання додатку

Для реалізації поставленої задачі можна використати багато різних технічних засобів. Провівши аналіз, були обрані найбільш зручні для створення застосунку. Вибір детально описаний нижче.

2.1.1 Вибір мови програмування

Для порівняння було обрано декілька мов програмування:

- Java
- Kotlin

Java – об'єктно орієнтована мова програмування, була випущена компанією «Sun Microsystems» в 1995 році. Java-застосунки компілюються в байт-код, який при виконанні буде інтерпретований віртуальною машиною (JVM) під певну платформу. JVM – використовує байт-код Java та передає інструкції обладнання, як інтерпретатор. При створенні мови було запозичено деякі особливості з C++. Було взято за основу об'єктну модель C++, але проведено її модифікацію, що значно зменшило можливість конфліктних ситуацій, через помилки програміста.

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		03.03.21	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		03.03.21		P2	17	52
Гарант		Цензура М.О.		03.03.21				
Розробник		Нікітенко А.Є.		03.03.21	Проектування мобільного додатку для спрощення ознайомлення та розміщення в місті	Факультет інформаційних технологій, 4 курс, 7 група		

Перевагою даної мови програмування є незалежність від архітектури. Програма написана на мові Java працюватиме на будь-якій підтримуваний системній платформі без будь-яких змін у початковому коді. Для оптимізації перетворювання байт-коду в машинний код, було створено метод just-in-time. Суть методу в тому, що перетворення коду відбувається при першому запуску програми, а потім просто кешується.

Наступною перевагою можна назвати об'єктно-орієнтованість Java. Всі дані і дії, окрім примітивних типів, групуються в класи та об'єкти. Всі об'єкти є похідними від одного головного Object, з якого вони успадковують поведінку і властивості.

Не можна не відмітити безпеку Java. Одна з особливостей концепції віртуальної машини – помилки або виключення не призводять до повного краху системи. Існують інструменти автоматизованої обробки виключень, які надають основну інформацію відносно виключень в програмах Java.

Мова програмування Java має автоматичне збирання сміття для керування пам'яттю. Розробник створює об'єкти, а віртуальна машина відповідає за звільнення пам'яті коли вони стають не потрібні. Коли до об'єкта не залишається посилань, збирач сміття може автоматично прибрати його з пам'яті.

Недоліком цієї мови є не можливість використовувати її для створення систем збій в яких може призвести до небезпечних ситуацій. Згідно з пунктом ліцензії Microsoft, програми написані цією мовою є не надійними та іноді можуть ламатись.

Kotlin – це відносно молода мова програмування від компанії «JetBrains», створена у 2011 році. З 2017 року вона отримала офіційну підтримку для розробки Android застосунків. Kotlin підтримує як об'єктно-орієнтоване, так і процедурне програмування.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		18

Код написаний на Kotlin компілюється в байт-код JVM або в JavaScript.

Синтаксис мови схожий на JavaScript, Паскаль, TypeScript, Нахе, PL/SQL, F#, Go й Scala, C++, Java, C#, Rust й D.

Ця мова має Null-безпеку, а це значить, що при спробі присвоїти змінній значення null код не скомпілюється і значно зменшить кількість помилок застосунку.

Прості функції та структури можна об'явити всього одним рядком коду, а гетери та сетери задаються за кулісами для інтероперабельності з Java-кодом.

При використанні мови, в більшості випадків не треба явно вказувати оператори приведення, оскільки існує оператор is, який зробить всю роботу.

Мінусом мови Kotlin є те, що в порівнянні з іншими подібними мовами, у нього нема зручного способу для описання структури даних. Це мова в якій нема синтаксису для опису структури даних.

Провівши аналіз розглянутих вище мов програмування, було обрано Java, тому що це перевірена і зручна мова програмування для створення Android додатків.

2.1.2 Вибір середовища для розробки

В цьому розділі треба обрати середовище розробки серед нижче розглянутих.

Можливі варіанти:

- IntelliJ IDEA
- Eclipse
- Android Studio

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		19

IntelliJ IDEA – інтегроване середовище розробки програмного забезпечення для багатьох мов програмування. Перша версія з'явилась у 2001 році

Програма розуміє структуру мови програмування та виконує наступні дії:

- Підказує можливі помилки, при написанні коду без явної потреби витратити час на запуск створюваної програми.
- Коли використовується будь-яка функція, IntelliJ IDEA показує документацію для неї.
- Допомогає швидко перейменувати змінну. Назва буде змінена в усьому коді одночасно.

Починаючи з шостої версії IntelliJ IDEA надає можливість розробки графічного інтерфейсу користувача. З дев'ятої версії середовище розробки доступне у двох редакціях Community Edition і Ultimate Edition.

- Community Edition – це повністю вільна версія з доступом під ліцензією Apache 2.0, в ній реалізована повна підтримка Java SE, Kotlin, Groovy, Scala.
- Ultimate Edition – ця версія доступна під комерційною ліцензією, реалізована підтримка Java EE, UML-діаграм.

Дизайн середовища орієнтований на продуктивність роботи програмістів, дозволяє сконцентруватись на функціональних задачах, а IntelliJ IDEA буде виконувати всі рутинні операції.

Eclipse – вільне інтегроване середовище розробки модульних кросплатформених застосунків.

Основою Eclipse являється платформа розширеного клієнта. Вона складається з наступних компонентів:

- Ядро платформи

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		20

- OSGi (стандартне середовище надання комплектів)
- SWT (портований інструментарій)
- JFace (файлові буфери, робота з текстом, текстові редактори)
- Робоче середовище Eclipse

GUI в Eclipse написано з використанням інструментарію SWT. Інтерфейс користувача також залежить від проміжного шару GUI названого JFace, який спрощує побудову інтерфейсу.

Гнучкість Eclipse забезпечено за допомогою підключаємих модулів. Використовуючи їх у нас є можливість розробки багатьма мовами окрім Java.

Це середовище розробки в першу чергу платформа для розробки розширень, будь-який розробник може покращити Eclipse за допомогою своїх модулів.

Eclipse JDT (Java Development Tools) – найпопулярніший модуль, націлений на групову розробку – інтегроване середовище з системами управління версіями.

Eclipse SDK – це мінімальна версія, яка прекрасно підходить для знайомства з платформою. В подальшому розвитку ви зможете завантажити та використовувати більш слушну для вас версію.

В стандартний набір Eclipse SDK включено два плагіни, Java Development Tools і Plug-in Developer Environment, таким чином ми отримуємо повністю готову IDE.

Android Studio – інтегроване середовище розробки для платформи Android. У 2014 році з'явився перший стабільний реліз і це стала заміна плагіну ADT для платформи Eclipse.

Середовище розробки адаптоване для виконання типових завдань, тестування на сумісність з різними версіями платформи та є інструменти для проектування програм на платформі Android. Крім перелічених можливостей в

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		21

Android Studio присутні нова уніфікована підсистема складання і розгортання застосунків за допомогою інструментарію Gradle.

Для спрощення та прискорення розробки інтерфейсу користувача представлено колекцію типових елементів та візуальний редактор для їх розміщення, з можливістю попереднього перегляду різних станів інтерфейсу.

Особливостями Android Studio передбачені такі функції:

- Базування на Gradle
- Існують шаблони для поширених дизайнів застосунків
- Android-орієнтований рефакторинг та швидкі виправлення
- Можливість підписів до програм
- Зручний редактор макетів, який дозволяє використовувати drag-and-drop та велику кількість стандартних елементів, також є можливість створювати свої елементи
- Є можливість змінювати макет інтерфейсу не тільки через візуальну частину, але й за допомогою xml файлів

Провівши аналіз вище розглянутих середовищ розробки, ми оберемо Android Studio, тому що вона зможе виконати всі наші задачі, та є офіційним середовищем розробки від Google.

2.2 Структура застосунку

Структура застосунку представлена:

- API для користувача
- Зовнішні ресурси
- Фрагменти та модулі, які їми керують.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		22

2.3 Проєктування пакетів та класів

2.3.1 Проєктування пакетів

В ході розробки було спроектовано 3 основних пакети:

- adapter
- fragment
- activity

Розглянемо діаграму пакетів, яка зображена на рисунку 2.1

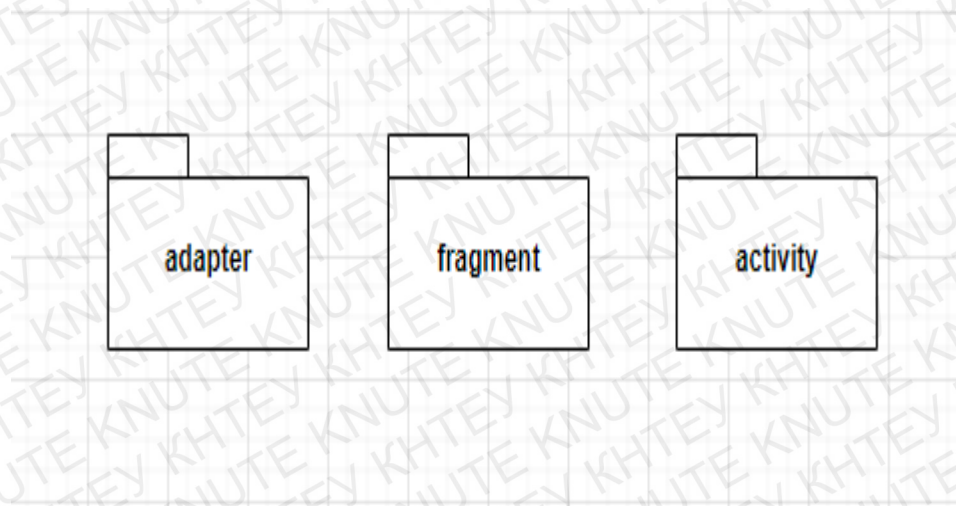


Рис. 2.1 – Діаграма пакетів

Кожен пакет виконує свою функцію та має певне призначення.

2.3.2 Проєктування класів

На рисунку 2.2 зображені класи з пакета adapter.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		23

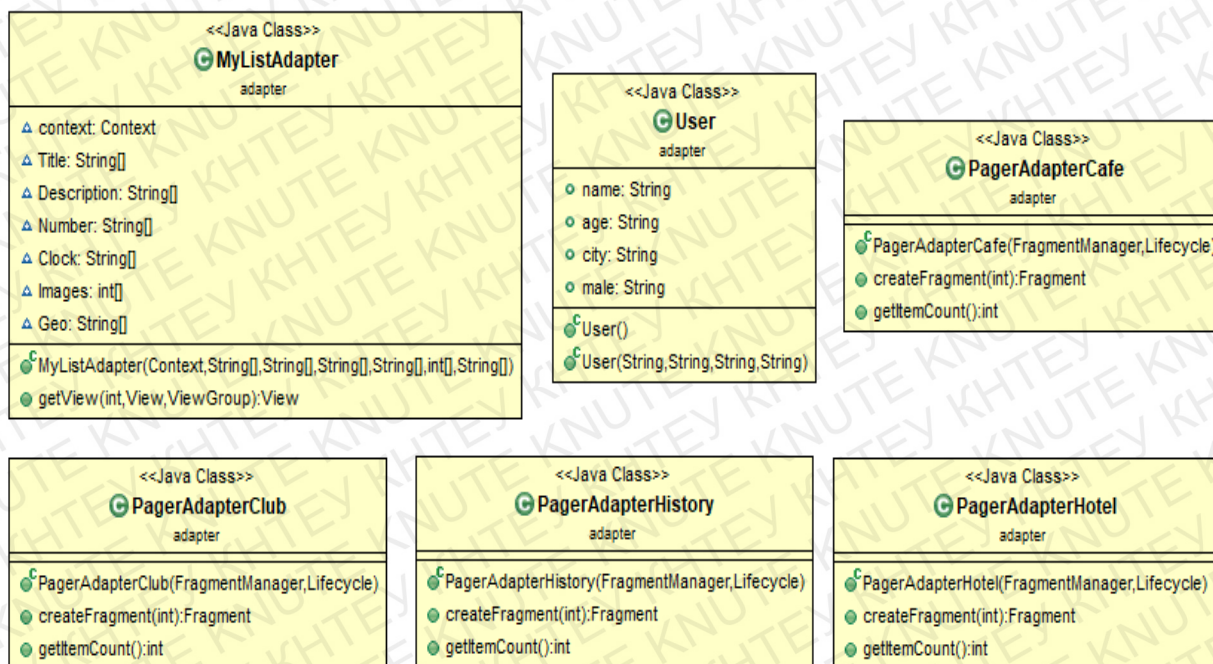


Рис. 2.2 – Класи пакету adapter

На рисунку 2.3 зображені класи, які входять в пакет fragment.

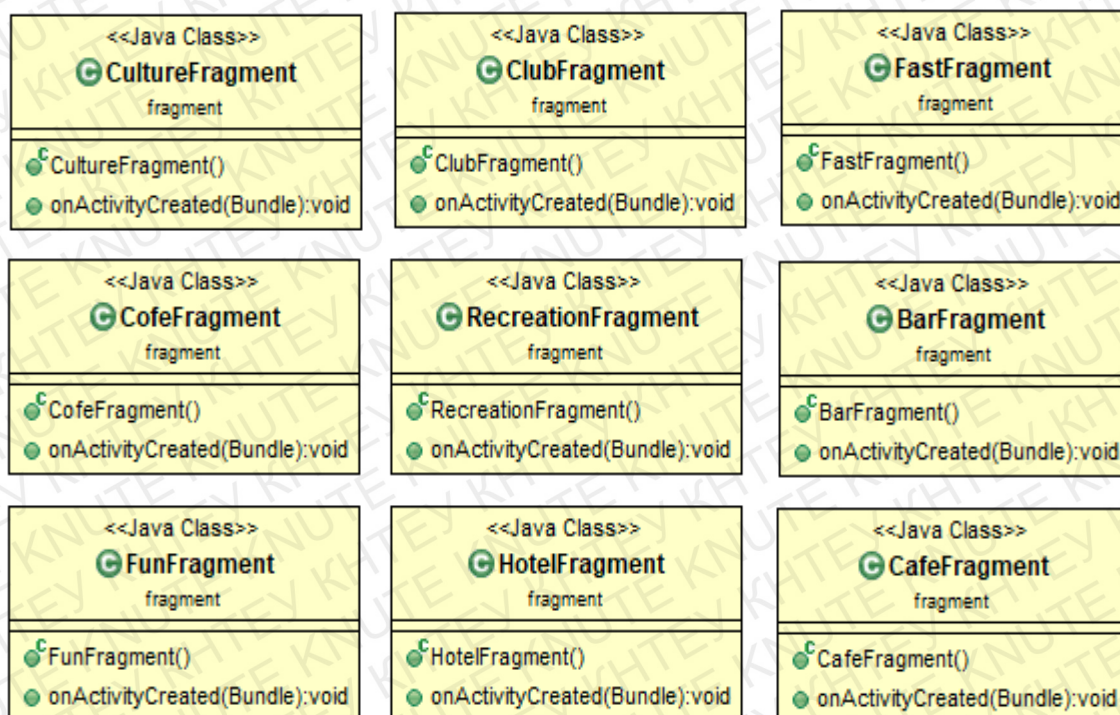


Рис. 2.3 – Класи пакету fragment

На рисунку 2.4 зображені класи, які входять в пакет activity.

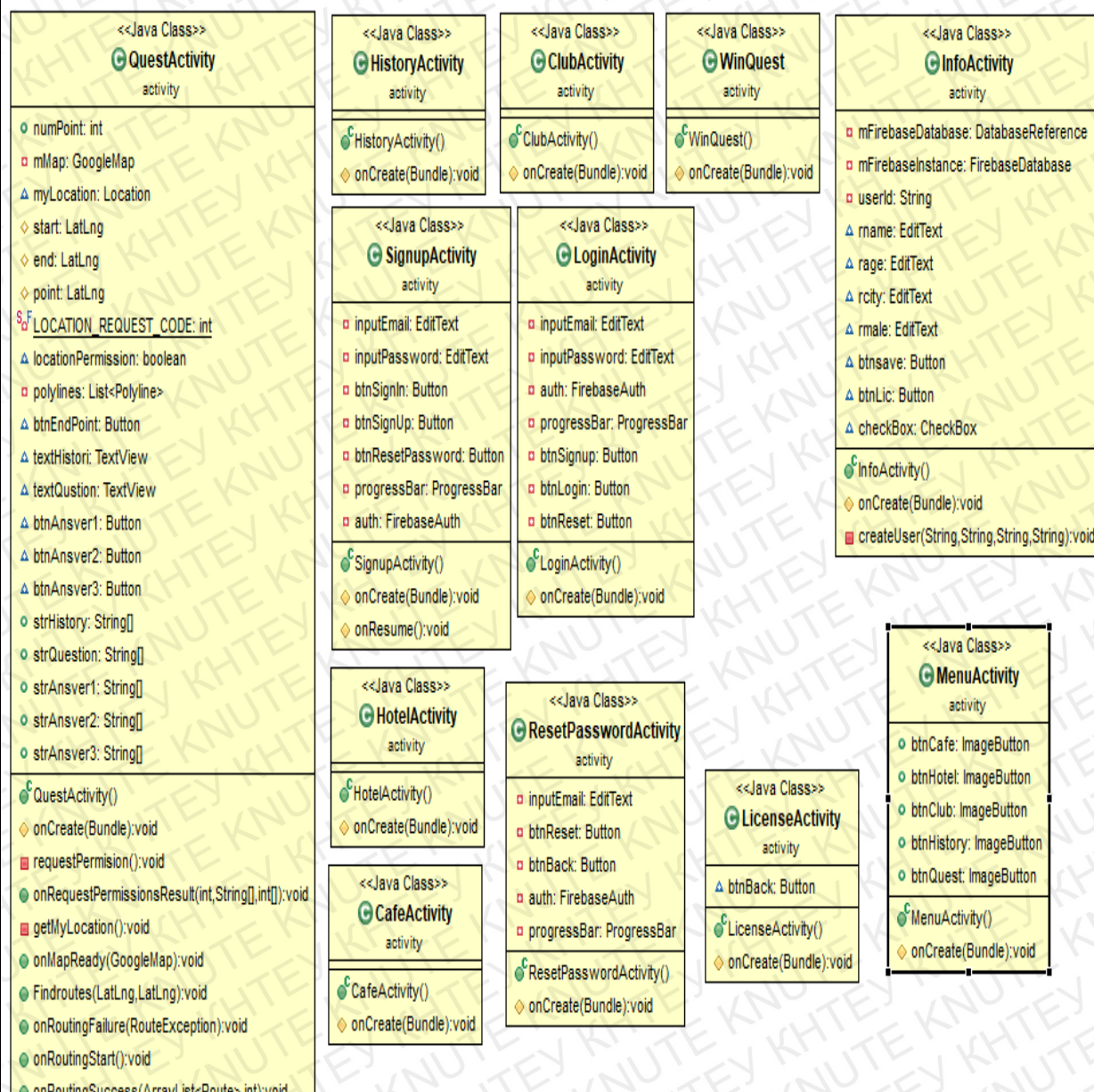


Рис. 2.4 – Класи пакету activity

2.4 Проектування додатку

2.4.1 Сценарій створення облікового запису в додатку

Для використання додатку спочатку треба створити обліковий запис або увійти в нього. На рисунку 2.5 проілюстровано вікно входу, а на рисунках 2.6-2.7 зображено вікно реєстрації.

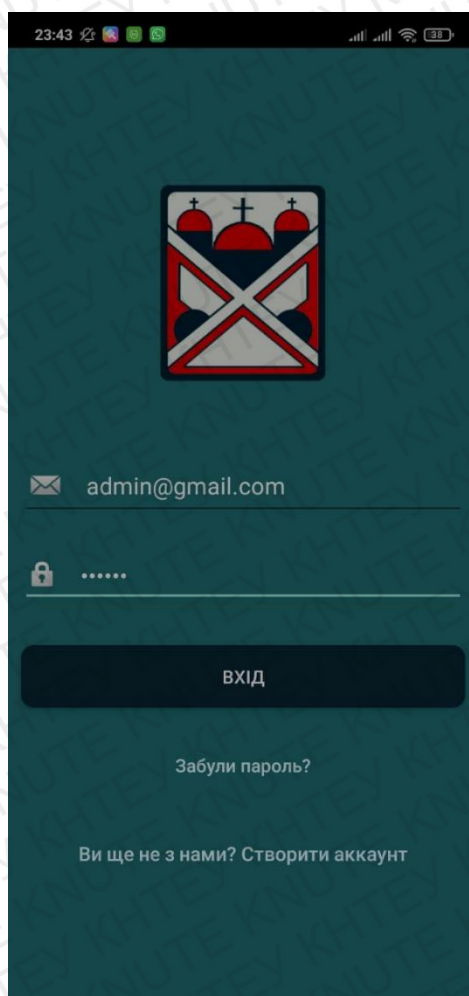


Рис. 2.5 – Вікно входу

Вікно реєстрації з натискання «Створити аккаунт».

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		03.03.21	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		03.03.21		Р2	26	52
Гарант		Цензура М.О.		03.03.21		Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		03.03.21				

Рис. 2.6 – Вікно реєстрації

Після того натискаємо кнопку «Створити аккаунт» і запусниться сценарій заповнення особової інформації.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		27

Рис. 2.7 – Вікно заповнення особової інформації

2.4.2 Сценарій обрання закладу та побудова маршруту

У застосунку можна переглянути списки наявних в Чернігові закладів, а також побудувати маршрут за допомогою Google maps. Функції перегляду закладів та побудова маршруту доступні після обрання відповідного пункту головного меню – проілюстровано на рисунку 2.8-2.9

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		28

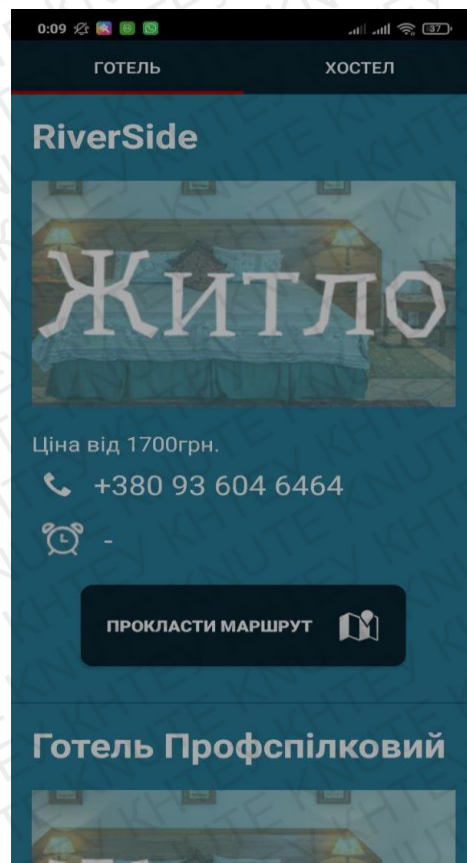


Рис. 2.8 – Список можливих закладів

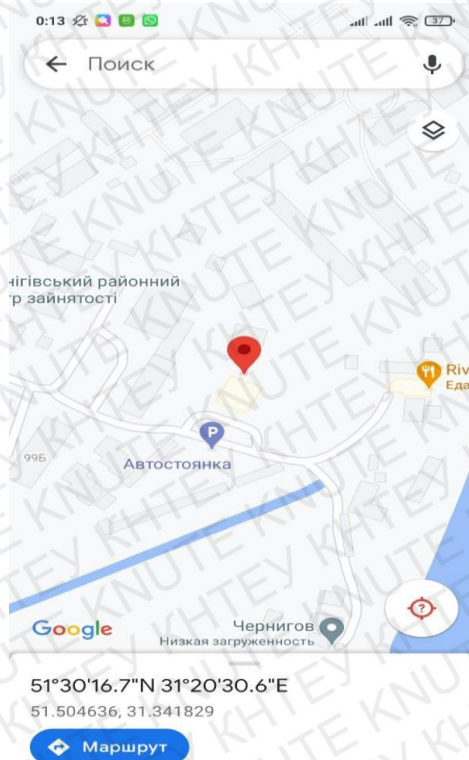


Рис. 2.9 – Обрано певний заклад

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		29

В застосунку можна змінювати підкатегорію за допомогою свайпів в ліво та право, а в верхній частині екрану ми можемо спостерігати зміни.

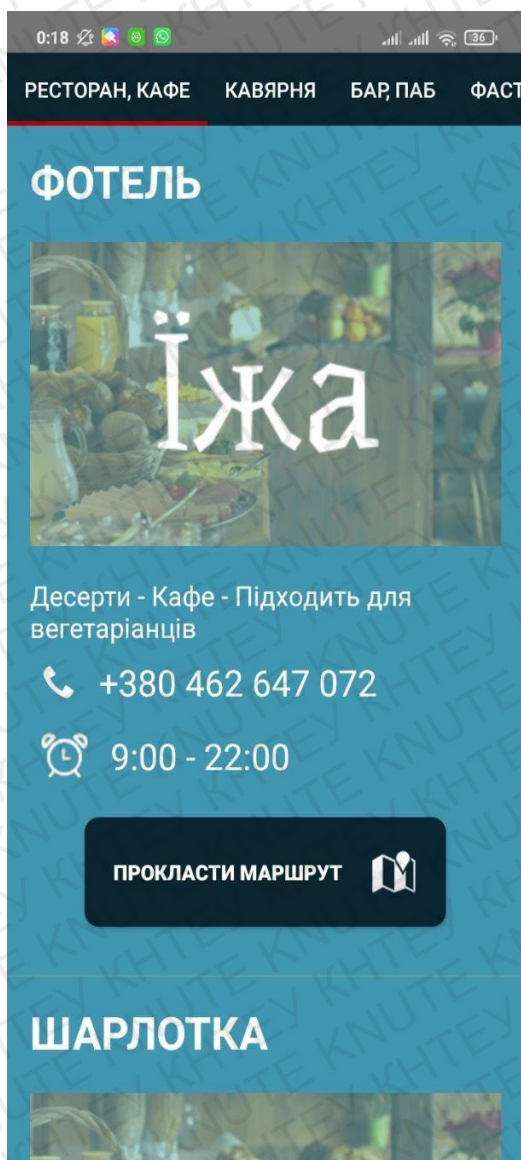


Рис. 2.10 – Обрано підкатегорію «Ресторани, кафе»

КНТЕУ 121 07-15.БР					Аркуш
Зм	Арк уш	№ докум	Підпис	Дата	30



Рис. 2.11 – обрано підкатегорію «Кав'ярня»

2.4.3 Сценарій проходження квесту

Сценарій роботи проходження квесту по місту. Для початку треба обрати з головного меню відповідний пункт. Меню зображено на рисунку 2.12

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		31



Рис. 2.12 – Головне меню програми

Після відкриття відповідного вікна, вмикаємо свою геопозицію та натискаємо на кнопку «Отримати току», для нас буде побудований маршрут від нашої позиції до потрібного пункту призначення. Побудова маршруту зображена на рисунку 2.13.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		32

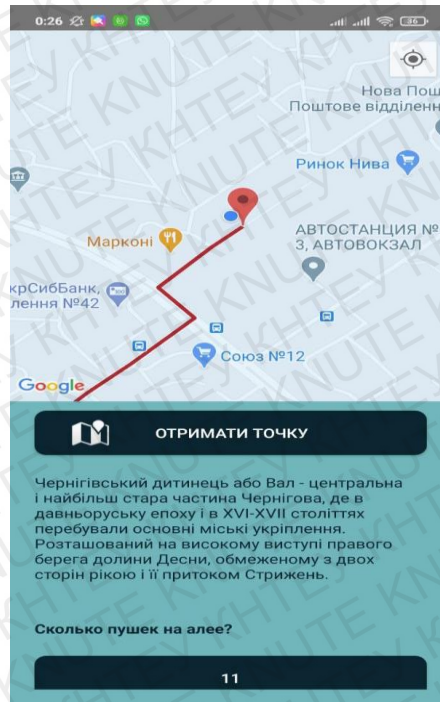


Рис. 2.13 – Побудова маршруту до пункту призначення

Для отримання нової точки вам потрібно правильно відповісти поставлене питання. Після проходження всіх точок ви побачите привітальне повідомлення, яке зображено на рисунку 2.14.

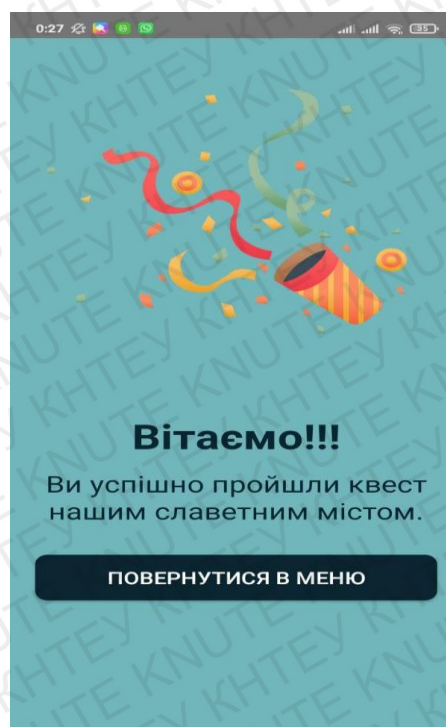


Рис. 2.14 – Привітальне повідомлення

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		33

Висновки до розділу 2

1. На основі аналізу переваг і недоліків існуючих мов програмування мобільних додатків було обрано мову Java, тому що вона задовольняє всі потреби та є офіційною мовою програмування для платформи Android, та є можливість використання на інших платформах.
2. Розглянуто декілька допустимих варіантів середовищ розробки, та обрано Android Studio, як зручний інструмент, що має велику підтримку платформи Android та признаний офіційним середовищем розробки мобільних додатків.
3. Спроектовано архітектуру проекту, та розроблені діаграми пакетів та класів.
4. Розроблені сценарії реєстрації користувача додатку, вибору закладу відпочинку та побудови маршруту до вибраних закладів.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		34

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ

3.1 Результати розробки пакетів

Для зручності знаходження класів, та зрозумілого виокремлення їх призначення, були створені пакети, які розділяють вихідний код на логічні частини.

Нижче у таблиці 3.1 представлений список пакетів та їх опис.

Таблиця 3.1

Опис пакетів вихідних файлів

Назва пакету	Список файлів	Призначення
activity	CafeActivity.java ClubActivity.java HistoryActivity.java HotelActivity.java InfoActivity.java LicenseActivity.java LoginActivity.java MenuActivity.java QuestActivity.java ResetPasswordActivity.java SignupActivity.java WinQuest.java	Класи, які використовуються для відображення активіті та взаємодії з ним.
adapter	MyListAdapter.java PagerAdapterCafe.java PagerAdapterClub.java PagerAdapterHistory.java PagerAdapterHotel.java User.java	Класи-адаптери, які використані при роботі з компонентами для відображення інформації.

					КНТЕУ 121 07-15.БР				
Зм	Аркуш	№ докум	Підпис	Дата					
Зав.каф.		Криворучко О.В.		10.04.21	«Розробка інформаційно-розважального мобільного додатку для туристів»		Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		10.04.21			РЗ	35	52
Гарант		Цензура М.О.		10.04.21			Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		10.04.21	Розробка додатку				

3.2 Результати розробки класів

3.2.1 Класи пакету activity

Пакет activity складається з класів, які ми використовуємо для відображення елементів активіті. Класи LoginActivity.java та SignupActivity.java використовуються для виконання входу та реєстрації відповідно. Існує клас для скидання паролю ResetPasswordActivity.java. Для заповнення інформації про користувача InfoActivity.java. Для перегляду ліцензійної згоди ми використовуємо LicenseActivity.java. CafeActivity.java, ClubActivity.java, HistoryActivity.java, HotelActivity.java – ці класи призначені для відображення основних сторінок і верхнього меню певного підрозділу. Для реалізації квесту та привітання користувача ми створили QuestActivity.java та WinQuest.java.

3.2.1.1 Клас LoginActivity

Клас унаслідований від AppCompatActivity. Використовується для входу користувачів в акаунт. Всі дані про користувачів зберігаються в Firebase. Код класу показано на лістинг 3.1.

Лістинг 3.1 – Клас LoginActivity

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		36

```
public class LoginActivity extends AppCompatActivity {
```

```
    private EditText inputEmail, inputPassword;  
    private FirebaseAuth auth;  
    private ProgressBar progressBar;  
    private Button btnSignup, btnLogin, btnReset;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        auth = FirebaseAuth.getInstance();
```

```
        setContentView(R.layout.activity_login);
```

```
        inputEmail = (EditText) findViewById(R.id.email);
```

```
        inputPassword = (EditText) findViewById(R.id.password);
```

```
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
```

```
        btnSignup = (Button) findViewById(R.id.btn_signup);
```

```
        btnLogin = (Button) findViewById(R.id.btn_login);
```

```
        btnReset = (Button) findViewById(R.id.btn_reset_password);
```

```
        auth = FirebaseAuth.getInstance();
```

```
        btnSignup.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                startActivity(new Intent(LoginActivity.this,  
SignupActivity.class));
```

```
            }
```

```
        });
```

```
        btnReset.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                startActivity(new Intent(LoginActivity.this,  
ResetPasswordActivity.class));
```

```
            }
```

```
        });
```

```
        btnLogin.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                String email = inputEmail.getText().toString();
```

```
                final String password = inputPassword.getText().toString();
```

```
                if (TextUtils.isEmpty(email)) {
```

```
                    Toast.makeText(getApplicationContext(), "Введіть Email!",  
Toast.LENGTH_SHORT).show();
```

```
                    return;
```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		37

```

    }

    if (TextUtils.isEmpty(password)) {
        Toast.makeText(getApplicationContext(), "Введіть пароль!",
Toast.LENGTH_SHORT).show();
        return;
    }

    progressBar.setVisibility(View.VISIBLE);

    //authenticate user
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult>
task) {
            // If sign in fails, display a message to the
user. If sign in succeeds
            // the auth state listener will be notified and
logic to handle the
            // signed in user can be handled in the listener.
            progressBar.setVisibility(View.GONE);
            if (!task.isSuccessful()) {
                // there was an error
                if (password.length() < 6) {
inputPassword.setError(getString(R.string.minimum_password));
                } else {
                    Toast.makeText(LoginActivity.this,
getString(R.string.auth_failed), Toast.LENGTH_LONG).show();
                }
            } else {
                startActivity(new Intent(LoginActivity.this,
MenuActivity.class));
                Toast.makeText(LoginActivity.this,
getString(R.string.auth_success), Toast.LENGTH_LONG).show();
            }
        }
    });
}
}
}
}

```

3.2.1.2 Клас SignupActivity

Цей клас призначений для реєстрації користувачів. Він приймає введену інформацію, перевіряє, щоб поля були не пусті, та заносить користувача до бази даних.

3.2.1.3 Клас ResetPasswordActivity

Цей клас призначений для можливості зміни паролю користувачем. Функція отримує пошту користувача та за допомогою сервісів Firebase пароль буде відновлено.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		38

3.2.1.4 Клас CafeActivity

Класи CafeActivity.java, ClubActivity.java, HistoryActivity.java, HotelActivity.java побудовані по подібній схемі і призначені для відображення верхнього меню категорії та надання можливості переходу між підкатегоріями. Код класу CafeActivity наведено в лістингу 3.2

Лістинг 3.2 – Клас CafeActivity

```
public class CafeActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cafe);
        final TabLayout tabLayout = findViewById(R.id.tabLayout);

        FragmentManager fm = getSupportFragmentManager();
        PagerAdapterCafe sa = new PagerAdapterCafe(fm, getLifecycle());
        final ViewPager2 pa = findViewById(R.id.viewPager);
        pa.setAdapter(sa);

        tabLayout.addTab(tabLayout.newTab().setText("Ресторан, кафе"));
        tabLayout.addTab(tabLayout.newTab().setText("Кавярня"));
        tabLayout.addTab(tabLayout.newTab().setText("Бар, паб"));
        tabLayout.addTab(tabLayout.newTab().setText("Фастфуд"));

        tabLayout.addTabSelectedListener(new TabLayout.OnTabSelectedListener()
        {
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                pa.setCurrentItem(tab.getPosition());
            }

            @Override
            public void onTabUnselected(TabLayout.Tab tab) {
            }

            @Override
            public void onTabReselected(TabLayout.Tab tab) {
            }
        });

        pa.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {
            @Override
            public void onPageSelected(int position) {
                tabLayout.selectTab(tabLayout.getTabAt(position));
            }
        });
    }
}
```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		39

3.2.2 Класи пакету adapter

Пакет adapter складається з класів, які призначені для створення адаптерів в застосунку.

3.2.2.1 Клас MyListAdapter

Цей клас призначений для створення та компонування елементів списку, в нашому випадку для створення кожного закладу. Тут ми зазначаємо звідки береться інформація для кожного елементу.

Код класу MyListAdapter наведено в лістингу 3.3

Лістинг 3.3 – Клас MyListAdapter

```
public class MyListAdapter extends ArrayAdapter<String> {
    Context context;
    String Title[];
    String Description[];
    String Number[];
    String Clock[];
    int Images[];
    String Geo[];

    public MyListAdapter(@NonNull Context context, String title[], String
description[], String number[], String clock[], int images[], String geo[]) {
        super(context, R.layout.style_row, R.id.textView1, title);
        this.context = context;
        this.Title = title;
        this.Description = description;
        this.Number = number;
        this.Clock = clock;
        this.Images = images;
        this.Geo = geo;
    }
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        LayoutInflater inflater =
(LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row = inflater.inflate(R.layout.style_row, parent, false);
        ImageView images = row.findViewById(R.id.image);
        TextView myTitle = row.findViewById(R.id.textView1);
        TextView myDescription = row.findViewById(R.id.textView2);
        TextView myNumber = row.findViewById(R.id.textView3);
        TextView myClock = row.findViewById(R.id.textView4);

        Button btn = row.findViewById(R.id.btnGoPoint);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		40

```

        Toast.makeText(context.getApplicationContext(), "Шукаємо
точку...", Toast.LENGTH_LONG).show();
        Uri gmmIntentUri = Uri.parse("geo:0,0?q="+Geo[position]+" (Місце
призначення)");
        Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
        mapIntent.setPackage("com.google.android.apps.maps");
        context.startActivity(mapIntent);
    }
});

images.setImageResource(Images[position]);
myTitle.setText(Title[position]);
myDescription.setText(Description[position]);
myNumber.setText(Number[position]);
myClock.setText(Clock[position]);
return row;
}
}

```

3.2.3 Класи пакету fragment

Пакет fragment складається з класів, які призначені для заповнення інформацією всіх сторінок підкатегорій. В нашому випадку в цих класах зберігається інформація про назву, опис, номер телефона, час роботи та геопозицію закладу. Потім інформація передається в адаптер.

3.2.3.1 Клас BarFragment

Цей клас призначений для зберігання інформації про бари нашого міста. Дані зберігаються в масивах, а в кінці передаються в адаптер, для кожного закладу своя інформація.

Код класу BarFragment наведено в лістингу 3.4

Лістинг 3.4 – Клас BarFragment

```

public class BarFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"Zivot a Pivo", "Robata", "Pub CASH", "ФАЛЬВАРЕК"};
        String mDescription[]={"Бар - Європейська - Паб"
            , "Японська - Паб з пивоварнею - Піцца"
            , "Паб"
            , "Паб - Східноєвропейська - Українська"};
        String mNumber[]={"+380 67 321 6214", "+380 63 100 4008", "+380 462 605
942", "93-58-53"};
        String mClock[]={"17:00 - 22:00", "11:30 - 01:00", "12:00 - 00:00", "11:00 -
00:00"};
        int
images[]={R.drawable.cafe, R.drawable.cafe, R.drawable.cafe, R.drawable.cafe};
        String geo[]={ "51.49498303165736, 31.30393744851036", "51.49077798329214,
31.300424563446665", "51.500615071360095, 31.28421703584785", "51.491726001911495,
31.297441649150468"};
    }
}

```

						КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата			41

```

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.3 Розробка інтерфейсу користувача

3.3.1 Екран входу

Розглянемо екран входу, він відкривається після запуску застосунку. Інтерфейс екрану входу зображено на рисунку 3.1 за відображення інтерфейсу додатку відповідають файли розмітки.

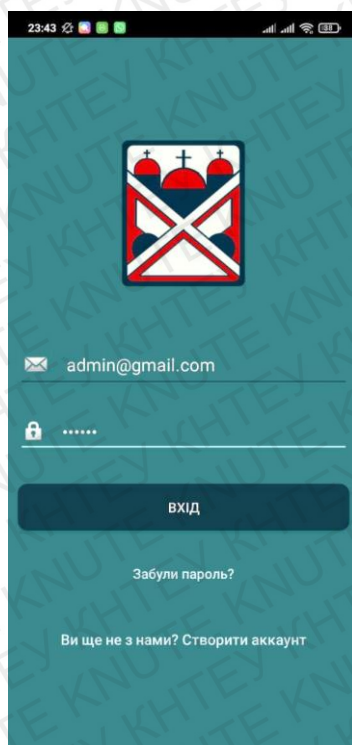


Рис. 3.1 – Інтерфейс екрану входу

Лістинг 3.5 – Файл розмітки для відображення екрану входу

```

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

```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		42

```

android:gravity="center"
android:orientation="vertical"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">
<ImageView
    android:id="@+id/imageView"
    android:layout_width="200dp"
    android:layout_height="180dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginBottom="30dp"
    app:srcCompat="@drawable/logo" />
<EditText
    android:id="@+id/email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:layout_marginBottom="10dp"
    android:drawableLeft="@android:drawable/sym_action_email"
    android:drawablePadding="16dp"
    android:ems="10"
    android:hint="@string/hint_email"
    android:inputType="textEmailAddress"
    android:textColor="@android:color/white"
    android:textColorHint="@android:color/white" />
<EditText
    android:id="@+id/password"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:layout_marginBottom="10dp"
    android:drawableLeft="@android:drawable/ic_lock_idle_lock"
    android:drawablePadding="16dp"
    android:ems="10"
    android:hint="@string/hint_password"
    android:inputType="textPassword"
    android:textColor="@android:color/white"
    android:textColorHint="@android:color/white" />
<Button
    android:id="@+id/btn_login"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="20dp"
    android:layout_marginRight="10dp"
    android:background="@drawable/btn_style"
    android:text="@string/btn_login"
    android:textColor="@android:color/white" />
<Button
    android:id="@+id/btn_reset_password"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		43

```

        android:layout_marginTop="20dip"
        android:background="@null"
        android:text="@string/change_password"
        android:textAllCaps="false"
        android:textColor="@color/colorPrimaryDark" />
    <Button
        android:id="@+id/btn_signup"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@null"
        android:text="@string/btn_link_to_register"
        android:textAllCaps="false"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="15dp" />
</LinearLayout>
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_gravity="center|bottom"
    android:layout_marginStart="411dp"
    android:layout_marginLeft="411dp"
    android:layout_marginEnd="411dp"
    android:layout_marginRight="411dp"
    android:layout_marginBottom="731dp"
    android:visibility="gone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

3.3.2 Екран головного меню

Розглянемо екран головного меню, в ньому представлено 5 кнопок, кожна відповідає за свою категорію. Файл розмітки для відображення екрану головного меню наведено в лістингу 3.6.

Лістинг 3.6 – Файл розмітки для відображення екрану головного меню

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimaryDark"
    tools:context=".activity.MenuActivity"
    tools:layout_editor_absoluteX="20dp"
    tools:layout_editor_absoluteY="20dp">
    <LinearLayout

```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		44

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="20dp"
android:gravity="center"
android:orientation="vertical"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">
<ImageButton
    android:id="@+id/btncafe"
    style="@android:style/Widget.ImageButton"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_margin="3dp"
    android:layout_weight="1"
    android:background="@android:color/transparent"
    android:scaleType="centerCrop"
    android:scrollbarThumbHorizontal="@color/colorPrimary"
    app:srcCompat="@drawable/cafe" />
<ImageButton
    android:id="@+id/btnhistory"
    style="@android:style/Widget.ImageButton"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_margin="3dp"
    android:layout_weight="1"
    android:background="@android:color/transparent"
    android:scaleType="centerCrop"
    app:srcCompat="@drawable/history" />
<ImageButton
    android:id="@+id/btnquest"
    style="@android:style/Widget.ImageButton"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_margin="3dp"
    android:layout_weight="1"
    android:background="@android:color/transparent"
    android:scaleType="centerCrop"
    app:srcCompat="@drawable/quest" />
<ImageButton
    android:id="@+id/btnclub"
    style="@android:style/Widget.ImageButton"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_margin="3dp"
    android:layout_weight="1"
    android:background="@android:color/transparent"
    android:scaleType="centerCrop"
    android:scrollbarThumbHorizontal="@color/colorPrimary"
    app:srcCompat="@drawable/club" />
<ImageButton
    android:id="@+id/btnhotel"

```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		45

```

style="@android:style/Widget.ImageButton"
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_margin="3dp"
android:layout_weight="1"
android:background="@android:color/transparent"
android:scaleType="centerCrop"
android:scrollbarThumbHorizontal="@color/colorPrimary"
app:srcCompat="@drawable/hotel" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

3.3.3 Екран обраної певної категорії

Розглянемо екран обраної певної категорії. В головному меню ми обираємо певну категорію і відбувається перехід на екран з пейджером та верхнім меню з під категоріями. Файл розмітки для відображення екрану певної категорії наведено в лістингу 3.7.

Лістинг 3.7 – Файл розмітки для відображення екрану певної категорії

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/colorPrimary"
tools:context=".activity.CafeActivity">
<com.google.android.material.tabs.TabLayout
android:id="@+id/tabLayout"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:isScrollContainer="true"
android:overScrollMode="ifContentScrolls"
android:textColor="@android:color/white"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:tabBackground="@color/colorPrimaryDark"
app:tabGravity="fill"
app:tabIndicatorColor="@color/colorAccent"
app:tabIndicatorFullWidth="true"
app:tabIndicatorGravity="bottom"
app:tabIndicatorHeight="3dp"
app:tabMode="scrollable"
app:tabTextColor="@android:color/white" />
<androidx.viewpager2.widget.ViewPager2
android:id="@+id/viewPager"
android:layout_width="0dp"
android:layout_height="0dp"
android:background="@color/colorPrimary"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/tabLayout" />

```

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		46

Висновки до розділу 3

В результаті розробки застосунку було створено пакети, які дозволили розділити програмний код на зручні та логічні частини.

Були розроблені інтерфейси користувача – екран авторизації, головного меню, певних категорій, проходження квесту та інші.

Додаток був розроблений згідно вимог технічного завдання.

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		47

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У ході виконання кваліфікаційної роботи, згідно з технічним завданням було спроектовано і створено Android-застосунок для полегшення ознайомлення з містом.

У першому розділі був проведений аналіз розробки застосунку, в наступному розділі було спроектовано кваліфікаційну роботу (проектування архітектури та розроблені діаграми пакетів та класів), а в кінці реалізувався додаток. Під час створення інформаційної системи враховувалися основні тенденції розвитку мобільних застосунків під Android, були враховані всі недоліки та переваги.

Додаток був реалізований за допомогою середовища розробки AndroidStudio та використання мови програмування Java та AndroidSDK. Удосконалення додатку можливо в сторону розширення списку закладів, розробки серверної частини, а також покращення інтерфейсу та надійності системи.

Проект має практичну цінність та актуальність, через те, що його використання значно спростить та пришвидшить ознайомлення з містом та основними пам'ятками. Це посприє покращенню іміджу міста.

Проаналізувавши все сказане, ми можемо зробити висновки, що застосунок має певні недоліки, які можуть бути ліквідовані у майбутньому. Додаток реалізовано згідно з технічним завданням і тому результат отриманий в даній кваліфікаційній роботі вважаю відповідним поставленим задачам.

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		24.04.21	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		24.04.21		ВП	48	52
Гарант		Цензура М.О.		24.04.21		Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		24.04.21				
					Висновки та пропозиції			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дипломне проектування. Організаційно-методичні вказівки до дипломного проектування для студентів спеціальностей 7.05010201–“Комп’ютерні системи та мережі”, 7.05010202–“Системне програмування”, 7.05010203– “Спеціалізовані комп’ютерні системи”. /Укл.: Акименко А.М., Зінченко А. Л., Нікітенко Є. В., Пріла О. А., Роговенко А. І., Стасюк С. С., Хижняк А. В., Харченко М. В.– Чернігів: ЧДТУ, 2011. – 39 с.

2. Технології проектування програмних систем. Методичні вказівки до виконання лабораторних та самостійних робіт для студентів спеціальностей 7.05010201, 8.05010201 – «Комп’ютерні системи та мережі» та 7.05010202, 8.05010202 – «Системне програмування». / Укл.: Казимир В. В., Пріла О. А. – Чернігів: ЧДТУ, 2013. - 71 с., рос. мовою.

3. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Проектування інтерфейсів користувача» для студентів напрямку підготовки 8.05010201 – “Комп’ютерні системи та мережі” та 8.05010202 – “Системе програмування”. /Укл.: А. В. Хижняк, О. А Скоп - Чернігів: ЧДТУ, 2010. - 63с. рос. мовою.

4. Шилдт, Герберт. Java 8. Повне керівництво, 9-е вид.: Пер. 3 англ.. – М.: ООО «И.Д. Вильямс», 2015. – 1376 с.: іл.. – Парал. тит. англ. ISBN 978-5-8459-1918-2(рус).

5. Android Studio - Вікіпедія [Електроннийресурс]. - Режимдоступу: URL: https://uk.wikipedia.org/wiki/Android_Studio - Назвазекрану.

					КНТЕУ 121 07-15.БР			
Зм	Аркуш	№ докум	Підпис	Дата				
Зав.каф.		Криворучко О.В.		24.04.21	«Розробка інформаційно-розважального мобільного додатку для туристів»	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.		24.04.21		СВД	49	52
Гарант		Цензура М.О.		24.04.21		Факультет інформаційних технологій, 4 курс, 7 група		
Розробник		Нікітенко А.Є.		24.04.21				
					Список використаних джерел			

6. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Java та C# технології прикладного програмування» для студентів напрямку підготовки 6.050201 – „Комп'ютерна інженерія”. /Укл.: Зінченко А. Л., Павловський В. І., Лукін А. В. – Чернігів: ЧДТУ, 2010. – 157 с. рос. мовою.

7. Firebase – Офіційна документація[Електронний ресурс] – Режим доступу: URL: <https://firebase.google.com/docs/reference>

8. Google Maps SDK – Офіційна документація Google Maps SDK [Електронний ресурс] – Режим доступу: URL: <https://developers.google.com/android/reference/com/google/android/gms/maps/package-summary?hl=ru>

					КНТЕУ 121 07-15.БР	Аркуш
Зм	Арк уш	№ докум	Підпис	Дата		50

ДОДАТКИ

Додаток А

1. Пакет activity

1.1 CafeActivity.java

```
package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.viewpager2.widget.ViewPager2;

import android.os.Bundle;

import com.google.android.material.tabs.TabLayout;

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.PagerAdapterCafe;
import doublel.studio.tourquest.adapter.PagerAdapterHotel;

public class CafeActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cafe);
        final TabLayout tabLayout = findViewById(R.id.tabLayout);

        FragmentManager fm = getSupportFragmentManager();
        PagerAdapterCafe sa = new PagerAdapterCafe(fm, getLifecycle());
        final ViewPager2 pa = findViewById(R.id.viewPager);
        pa.setAdapter(sa);

        tabLayout.addTab(tabLayout.newTab().setText("Ресторан, кафе"));
        tabLayout.addTab(tabLayout.newTab().setText("Кавярня"));
        tabLayout.addTab(tabLayout.newTab().setText("Бар, паб"));
        tabLayout.addTab(tabLayout.newTab().setText("Фастфуд"));

        tabLayout.addTabSelectedListener(new TabLayout.OnTabSelectedListener()
        {
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                pa.setCurrentItem(tab.getPosition());
            }

            @Override
            public void onTabUnselected(TabLayout.Tab tab) {
            }

            @Override
            public void onTabReselected(TabLayout.Tab tab) {
            }
        })
    }
}
```

```

    });

    pa.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {
        @Override
        public void onPageSelected(int position) {
            tabLayout.selectTab(tabLayout.getTabAt(position));
        }
    });
}
}

```

1.2 ClubActivity.java

```

package doublel.studio.tourquest.activity;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.viewpager2.widget.ViewPager2;

import android.os.Bundle;

import com.google.android.material.tabs.TabLayout;

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.PagerAdapterClub;
import doublel.studio.tourquest.adapter.PagerAdapterHotel;

public class ClubActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_club);
        final TabLayout tabLayout = findViewById(R.id.tabLayout);

        FragmentManager fm = getSupportFragmentManager();
        PagerAdapterClub sa = new PagerAdapterClub(fm, getLifecycle());
        final ViewPager2 pa = findViewById(R.id.viewPager);
        pa.setAdapter(sa);

        tabLayout.addTab(tabLayout.newTab().setText("Активний відпочинок"));
        tabLayout.addTab(tabLayout.newTab().setText("Торгові центри"));
        tabLayout.addTab(tabLayout.newTab().setText("Клуби"));

        tabLayout.addTabSelectedListener(new TabLayout.OnTabSelectedListener()
        {
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                pa.setCurrentItem(tab.getPosition());
            }

            @Override
            public void onTabUnselected(TabLayout.Tab tab) {
            }
        }
    }
}

```

```

        @Override
        public void onTabReselected(TabLayout.Tab tab) {

        }
    });

    pa.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {
        @Override
        public void onPageSelected(int position) {
            tabLayout.selectTab(tabLayout.getTabAt(position));
        }
    });
}
}

```

1.3 HistoryActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.viewpager2.widget.ViewPager2;

import android.os.Bundle;

import com.google.android.material.tabs.TabLayout;

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.PagerAdapterCafe;
import doublel.studio.tourquest.adapter.PagerAdapterHistory;

public class HistoryActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);
        final TabLayout tabLayout = findViewById(R.id.tabLayout);

        FragmentManager fm = getSupportFragmentManager();
        PagerAdapterHistory sa = new PagerAdapterHistory(fm,
getLifecycle());
        final ViewPager2 pa = findViewById(R.id.viewPager);
        pa.setAdapter(sa);

        tabLayout.addTab(tabLayout.newTab().setText("Заклади культури"));

        tabLayout.addTabOnTabSelectedListener(new
TabLayout.OnTabSelectedListener() {
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                pa.setCurrentItem(tab.getPosition());
            }
        });
    }
}

```

```

        @Override
        public void onTabUnselected(TabLayout.Tab tab) {

        }

        @Override
        public void onTabReselected(TabLayout.Tab tab) {

        }
    });

    pa.registerOnPageChangeCallback(new
ViewPager2.OnPageChangeCallback() {
        @Override
        public void onPageSelected(int position) {
            tabLayout.selectTab(tabLayout.getTabAt(position));
        }
    });
}
}

```

1.4 HotelActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.viewpager2.widget.ViewPager2;

import android.os.Bundle;

import com.google.android.material.tabs.TabLayout;

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.PagerAdapterHistory;
import doublel.studio.tourquest.adapter.PagerAdapterHotel;

public class HotelActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hotel);
        final TabLayout tabLayout = findViewById(R.id.tabLayout);

        FragmentManager fm = getSupportFragmentManager();
        PagerAdapterHotel sa = new PagerAdapterHotel(fm, getLifecycle());
        final ViewPager2 pa = findViewById(R.id.viewPager);
        pa.setAdapter(sa);

        tabLayout.addTab(tabLayout.newTab().setText("Готель"));
        tabLayout.addTab(tabLayout.newTab().setText("Хостел"));
    }
}

```

```

        tabLayout.addTabSelectedListener(new
        TabLayout.OnTabSelectedListener() {
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                pa.setCurrentItem(tab.getPosition());
            }

            @Override
            public void onTabUnselected(TabLayout.Tab tab) {

            }

            @Override
            public void onTabReselected(TabLayout.Tab tab) {

            }
        });

        pa.registerOnPageChangeCallback(new
        ViewPager2.OnPageChangeCallback() {
            @Override
            public void onPageSelected(int position) {
                tabLayout.selectTab(tabLayout.getTabAt(position));
            }
        });
    }
}

```

1.5 InfoActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.User;

public class InfoActivity extends AppCompatActivity {

    private DatabaseReference mFirebaseDatabase;

```

```

private FirebaseDatabase mFirebaseInstance;

private String userId;

EditText rname, rage, rcity, rmale;
Button btnsave, btnLic;
CheckBox checkBox;

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

    mFirebaseInstance = FirebaseDatabase.getInstance();
    mFirebaseDatabase = mFirebaseInstance.getReference("users");

    rname = (EditText) findViewById(R.id.rname);
    rage = (EditText) findViewById(R.id.rage);
    rcity = (EditText) findViewById(R.id.rcity);
    rmale = (EditText) findViewById(R.id.rmale);
    btnsave = (Button) findViewById(R.id.btn_save);
    btnLic = (Button) findViewById(R.id.btn_lic);
    checkBox = (CheckBox) findViewById(R.id.checkBox);

    btnLic.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(InfoActivity.this,
LicenseActivity.class));
        }
    });

    btnsave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String name = rname.getText().toString();
            String age = rage.getText().toString();
            String city = rcity.getText().toString();
            String male = rmale.getText().toString();
            if(checkBox.isChecked()) {
                createUser(name, age, city, male);
            }
        }
    });
}

private void createUser(String name, String age, String city, String
male){
    if(TextUtils.isEmpty(userId)){
        userId = mFirebaseDatabase.push().getKey();
    }
    User user = new User(name, age, city, male);
    mFirebaseDatabase.child(userId).setValue(user);
    Toast.makeText(InfoActivity.this, "ok", Toast.LENGTH_SHORT).show();
}

```

```

        startActivity(new Intent(InfoActivity.this, MenuActivity.class));
    }
}

```

1.6 LicenseActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;

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

import doublel.studio.tourquest.R;

public class LicenseActivity extends AppCompatActivity {
    Button btnBack;

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

        btnBack = (Button) findViewById(R.id.btn_back);

        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LicenseActivity.this,
InfoActivity.class));
            }
        });
    }
}

```

1.7 LoginActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

```

```
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import doublel.studio.tourquest.R;

public class LoginActivity extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private FirebaseAuth auth;
    private ProgressBar progressBar;
    private Button btnSignup, btnLogin, btnReset;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        auth = FirebaseAuth.getInstance();
        setContentView(R.layout.activity_login);

        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        btnSignup = (Button) findViewById(R.id.btn_signup);
        btnLogin = (Button) findViewById(R.id.btn_login);
        btnReset = (Button) findViewById(R.id.btn_reset_password);

        auth = FirebaseAuth.getInstance();

        btnSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this,
SignupActivity.class));
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this,
ResetPasswordActivity.class));
            }
        });

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = inputEmail.getText().toString();
                final String password = inputPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getApplicationContext(), "Введите
Email!", Toast.LENGTH_SHORT).show();
                    return;
                }
            }
        });
    }
}
```

```

        if (TextUtils.isEmpty(password)) {
            Toast.makeText(getApplicationContext(), "Введіть
пароль!", Toast.LENGTH_SHORT).show();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);

        //authenticate user
        auth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull
Task<AuthResult> task) {
                // If sign in fails, display a message to
the user. If sign in succeeds
                // the auth state listener will be notified
and logic to handle the
                // signed in user can be handled in the
listener.

                progressBar.setVisibility(View.GONE);
                if (!task.isSuccessful()) {
                    // there was an error
                    if (password.length() < 6) {
                        inputPassword.setError(getString(R.string.minimum_password));
                    } else {
                        Toast.makeText(LoginActivity.this,
getString(R.string.auth_failed), Toast.LENGTH_LONG).show();
                    }
                } else {
                    startActivity(new
Intent(LoginActivity.this, MenuActivity.class));
                    Toast.makeText(LoginActivity.this,
getString(R.string.auth_success), Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
}
}

```

1.8 MenuActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;

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

```

```
import double1.studio.tourquest.R;

public class MenuActivity extends AppCompatActivity {

    public ImageButton btnCafe, btnHotel, btnClub, btnHistory, btnQuest;

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

        btnCafe = (ImageButton) findViewById(R.id.btncafe);
        btnHotel = (ImageButton) findViewById(R.id.btnhotel);
        btnClub = (ImageButton) findViewById(R.id.btnclub);
        btnHistory = (ImageButton) findViewById(R.id.btnhistory);
        btnQuest = (ImageButton) findViewById(R.id.btnquest);

        btnCafe.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MenuActivity.this, CafeActivity.class));
            }
        });
        btnHotel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MenuActivity.this,
HotelActivity.class));
            }
        });
        btnClub.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MenuActivity.this, ClubActivity.class));
            }
        });
        btnHistory.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MenuActivity.this,
HistoryActivity.class));
            }
        });
        btnQuest.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MenuActivity.this,
QuestActivity.class));
            }
        });
    }
}
```

1.9 QuestActivity.java

```
package doublel.studio.tourquest.activity;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.directions.route.AbstractRouting;
import com.directions.route.Route;
import com.directions.route.RouteException;
import com.directions.route.Routing;
import com.directions.route.RoutingListener;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.android.material.snackbar.Snackbar;

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

import doublel.studio.tourquest.R;

public class QuestActivity extends FragmentActivity implements
    OnMapReadyCallback,
        GoogleApiClient.OnConnectionFailedListener, RoutingListener {

    public int numPoint = 0;
    //google map object
    private GoogleMap mMap;

    //current and destination location objects
    Location myLocation = null;
    protected LatLng start = null;
    protected LatLng end = null;
    protected LatLng point = null;
```

```
//to get location permissions.  
private final static int LOCATION_REQUEST_CODE = 23;  
boolean locationPermission = false;
```

```
//polyline object  
private List<Polyline> polyLines = null;  
Button btnEndPoint;
```

```
TextView textHistory, textQuestion;  
Button btnAnswer1, btnAnswer2, btnAnswer3;
```

```
public String[] strHistory = {"Спасо-Преображенський собор - п'ятиглавий  
восьмистовпного храм в Чернігові, діючий храм-музей. Найстаріший збережений  
пам'ятник давньоруського зодчества, протягом століть неодноразово  
перебудовувався. Служив головним храмом Чернігово-Сіверського князівства і  
усипальницею місцевих князів."},
```

```
"Православна церква в українському місті Чернігові, пам'ятка  
архітектури національного значення. Яскравий зразок українського бароко  
Лівобережжя. Зведена на кошти братів Семена Лизогуба і Якова Лизогуба, за  
заповітом їхнього батька Юхима Лизогуба. Церква побудована в пам'ять про героїзм  
їх діда Якова Лизогуба і козаків чернігівського полку."},
```

```
"Чернігівський обласний академічний український музично-драматичний  
театр імені Т. Г. Шевченка - обласний музично-драматичний театр в Чернігові.  
Театр є одним з найстаріших обласних театрів України: він був створений в 1920-і  
роки",
```

```
"Храм святої Параскеви П'ятниці - православний храм в Чернігові. Храм  
належить Українській православній церкві Київського патріархату. Храм побудований  
в кінці XII - початку XIII століття на чернігівському посаді біля ринку на кошти  
купців."};
```

```
public String[] strQuestion = {"Скільки церков навколо вас?",  
"Яке жіноче ім'я носить церква?",  
"Скільки колон має драматичний театр?",  
"З яким днем тижня співзвучна назва церкви?"};
```

```
public String[] strAnswer1 = {"2", "Валентина", "7", "Середа"};  
public String[] strAnswer2 = {"3", "Лідія", "8", "Четверг"};  
public String[] strAnswer3 = {"4", "Катерина", "9", "П'ятниця"};
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_quest);  
    requestPermission();
```

```
textHistory = findViewById(R.id.textHistory);  
textQuestion = findViewById(R.id.textQuestion);
```

```
btnAnswer1 = findViewById(R.id.btnAnswer1);  
btnAnswer2 = findViewById(R.id.btnAnswer2);  
btnAnswer3 = findViewById(R.id.btnAnswer3);
```

```
btnAnswer1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(numPoint == 0){
```

```

        Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
    }else if(numPoint == 1){
        Toast.makeText(QuestActivity.this, "Правильно, відправляйтесь
на нову точку", Toast.LENGTH_LONG).show();
        point = new LatLng(51.48707604738959, 31.305276896497556);
        end = point;
        mMap.clear();
        start=new
LatLng(myLocation.getLatitude(),myLocation.getLongitude());
        Findroutes(start,end);
        numPoint++;
        newText();
    }else if(numPoint == 2){
        Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
    }else if(numPoint == 3){
        Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
    }else if(numPoint == 4){
        Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
    }
}
});
btnAnsver2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(numPoint == 0){
            Toast.makeText(QuestActivity.this, "Правильно, відправляйтесь
на нову точку", Toast.LENGTH_LONG).show();
            point = new LatLng(51.4875503782806, 31.30976154996699);
            end = point;
            mMap.clear();
            start=new
LatLng(myLocation.getLatitude(),myLocation.getLongitude());
            Findroutes(start,end);
            numPoint++;
            newText();
        }else if(numPoint == 1){
            Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
        }else if(numPoint == 2){
            Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
        }else if(numPoint == 3){
            Toast.makeText(QuestActivity.this, "Правильно, відправляйтесь
на нову точку", Toast.LENGTH_LONG).show();
            point = new LatLng(51.49244705502791, 31.301317955991575);
            end = point;
            mMap.clear();
            start=new
LatLng(myLocation.getLatitude(),myLocation.getLongitude());
            Findroutes(start,end);
            numPoint++;
            newText();
        }
    }
});

```

```

        }else if(numPoint == 4){
            Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
        }
    }
});
btnAnsver3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(numPoint == 0){
            Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
        }else if(numPoint == 1){
            Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
        }else if(numPoint == 2){
            Toast.makeText(QuestActivity.this, "Правильно, відправляйтесь
на нову точку", Toast.LENGTH_LONG).show();
            point = new LatLng(51.491077641768975, 31.29866793350057);
            end = point;
            mMap.clear();
            start=new
LatLng(myLocation.getLatitude(),myLocation.getLongitude());
            Findroutes(start,end);
            numPoint++;
            newText();
        }else if(numPoint == 3){
            Toast.makeText(QuestActivity.this, "Неправильно, спробуйте ще
раз!", Toast.LENGTH_LONG).show();
        }else if(numPoint == 4){
            Toast.makeText(QuestActivity.this, "Правильно, відправляйтесь
на нову точку", Toast.LENGTH_LONG).show();
            point = new LatLng(51.49241029830765, 31.301264333010135);
            end = point;
            mMap.clear();
            start=new
LatLng(myLocation.getLatitude(),myLocation.getLongitude());
            Findroutes(start,end);
            newText();
            startActivity(new Intent(QuestActivity.this,
WinQuest.class));
        }
    }
});

btnEndPoint = findViewById(R.id.btnPoint);
btnEndPoint.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(numPoint == 0){
            textHistori.setVisibility(View.VISIBLE);
            textQustion.setVisibility(View.VISIBLE);
            btnAnsver1.setVisibility(View.VISIBLE);
            btnAnsver2.setVisibility(View.VISIBLE);
            btnAnsver3.setVisibility(View.VISIBLE);
        }
    }
});

```

```

        point = new LatLng(51.4890248823466, 31.30707061919747);
        end = point;
        mMap.clear();
        start=new
        LatLng(myLocation.getLatitude(),myLocation.getLongitude());
        Findroutes(start,end);
    }
});
//init google map fragment to show map.
SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
        .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
}

private void requestPermission() {
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
            LOCATION_REQUEST_CODE);
    } else {
        locationPermission = true;
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
    switch (requestCode) {
        case LOCATION_REQUEST_CODE: {
            if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED)
            {
                //if permission granted.
                locationPermission = true;
                getMyLocation();
            } else {
                // permission denied, boo! Disable the
                // functionality that depends on this permission.
            }
            return;
        }
    }
}

//to get user location
private void getMyLocation() {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    {

```

```

        return;
    }

    mMap.setMyLocationEnabled(true);
    mMap.setOnMyLocationChangeListener(new
    GoogleMap.OnMyLocationChangeListener() {
        @Override
        public void onMyLocationChange(Location location) {

            myLocation = location;
            LatLng ltlng = new
            LatLng(location.getLatitude(),location.getLongitude());
            CameraUpdate cameraUpdate = CameraUpdateFactory.newLatLngZoom(
                ltlng, 16f);
            mMap.animateCamera(cameraUpdate);

        }
    });

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        getLocation();

    }

    // function to find Routes.
    public void Findroutes(LatLng Start, LatLng End)
    {

        if(Start==null || End==null) {
            Toast.makeText(QuestActivity.this,"Unable to get
            location",Toast.LENGTH_LONG).show();
        }
        else
        {

            Routing routing = new Routing.Builder()
                .travelMode(AbstractRouting.TravelMode.WALKING)
                .withListener(this)
                .alternativeRoutes(true)
                .waypoints(Start, End)
                .key("AIzaSyBbqBcbkPgWuFN_KWU1NVtakkOrF78YqEY") //also
            define your api key here.
                .build();
            routing.execute();

        }
    }

    //Routing call back functions.
    @Override

```

```

    public void onRoutingFailure(RouteException e) {
        View parentLayout = findViewById(android.R.id.content);
        Snackbar snackbar= Snackbar.make(parentLayout, e.toString(),
        Snackbar.LENGTH_LONG);
        snackbar.show();
    }
    // Findroutes(start,end);

    @Override
    public void onRoutingStart() {
        Toast.makeText(QuestActivity.this, "Finding
        Route...", Toast.LENGTH_LONG).show();
    }

    //If Route finding success..
    @Override
    public void onRoutingSuccess(ArrayList<Route> route, int shortestRouteIndex)
    {

        CameraUpdate center = CameraUpdateFactory.newLatLng(start);
        CameraUpdate zoom = CameraUpdateFactory.zoomTo(16);
        if(polyLines!=null) {
            polyLines.clear();
        }
        PolylineOptions polyOptions = new PolylineOptions();
        LatLng polylineStartLatLng=null;
        LatLng polylineEndLatLng=null;

        polyLines = new ArrayList<>();
        //add route(s) to the map using polyline
        for (int i = 0; i <route.size(); i++) {

            if(i==shortestRouteIndex)
            {
                polyOptions.color(getResources().getColor(R.color.colorAccent));
                polyOptions.width(7);
                polyOptions.addAll(route.get(shortestRouteIndex).getPoints());
                Polyline polyline = mMap.addPolyline(polyOptions);
                polylineStartLatLng=polyline.getPoints().get(0);
                int k=polyline.getPoints().size();
                polylineEndLatLng=polyline.getPoints().get(k-1);
                polyLines.add(polyline);
            }
            else {

            }

        }

        //Add Marker on route starting position
        MarkerOptions startMarker = new MarkerOptions();
        startMarker.position(polylineStartLatLng);
        startMarker.title("My Location");
        mMap.addMarker(startMarker);
    }

```

```

//Add Marker on route ending position
MarkerOptions endMarker = new MarkerOptions();
endMarker.position(polylineEndLatLng);
endMarker.title("Destination");
mMap.addMarker(endMarker);
}

@Override
public void onRoutingCancelled() {
    Findroutes(start,end);
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    Findroutes(start,end);
}

public void newText(){
    if(numPoint == 1){
        textHistori.setText(strHistory[0]);
        textQustion.setText(strQuestion[0]);
        btnAnsver1.setText(strAnsver1[0]);
        btnAnsver2.setText(strAnsver2[0]);
        btnAnsver3.setText(strAnsver3[0]);
    }else if(numPoint == 2){
        textHistori.setText(strHistory[1]);
        textQustion.setText(strQuestion[1]);
        btnAnsver1.setText(strAnsver1[1]);
        btnAnsver2.setText(strAnsver2[1]);
        btnAnsver3.setText(strAnsver3[1]);
    }else if(numPoint == 3){
        textHistori.setText(strHistory[2]);
        textQustion.setText(strQuestion[2]);
        btnAnsver1.setText(strAnsver1[2]);
        btnAnsver2.setText(strAnsver2[2]);
        btnAnsver3.setText(strAnsver3[2]);
    }else if(numPoint == 4){
        textHistori.setText(strHistory[3]);
        textQustion.setText(strQuestion[3]);
        btnAnsver1.setText(strAnsver1[3]);
        btnAnsver2.setText(strAnsver2[3]);
        btnAnsver3.setText(strAnsver3[3]);
    }else if(numPoint == 5){
        textHistori.setText(strHistory[4]);
        textQustion.setText(strQuestion[4]);
        btnAnsver1.setText(strAnsver1[4]);
        btnAnsver2.setText(strAnsver2[4]);
        btnAnsver3.setText(strAnsver3[4]);
    }
}
}
}

```

1.10 ResetPasswordActivity.java

```
package doublel.studio.tourquest.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

import doublel.studio.tourquest.R;

public class ResetPasswordActivity extends AppCompatActivity {

    private EditText inputEmail;
    private Button btnReset, btnBack;
    private FirebaseAuth auth;
    private ProgressBar progressBar;

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

        setContentView(R.layout.activity_reset_password);

        inputEmail = (EditText) findViewById(R.id.email);
        btnReset = (Button) findViewById(R.id.btn_reset_password);
        btnBack = (Button) findViewById(R.id.btn_back);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        auth = FirebaseAuth.getInstance();

        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String email = inputEmail.getText().toString().trim();

                if (TextUtils.isEmpty(email)) {
```

```

        Toast.makeText(getApplicationContext(), "Введите ваш Email",
        Toast.LENGTH_SHORT).show();
        return;
    }

    progressBar.setVisibility(View.VISIBLE);
    auth.sendPasswordResetEmail(email)
        .addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText(ResetPasswordActivity.this,
                    getString(R.string.forgot_password_msg), Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(ResetPasswordActivity.this,
                    getString(R.string.fail_email), Toast.LENGTH_SHORT).show();
                }

                progressBar.setVisibility(View.GONE);
            }
        });
    }
});
}
}

```

1.11 SignupActivity.java

```

package doublel.studio.tourquest.activity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import doublel.studio.tourquest.R;

public class SignupActivity extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private ProgressBar progressBar;
    private FirebaseAuth auth;

```

```

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

    //Get Firebase auth instance
    auth = FirebaseAuth.getInstance();

    btnSignIn = (Button) findViewById(R.id.sign_in_button);
    btnSignUp = (Button) findViewById(R.id.sign_up_button);
    inputEmail = (EditText) findViewById(R.id.email);
    inputPassword = (EditText) findViewById(R.id.password);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);

    btnSignIn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });

    btnSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            String email = inputEmail.getText().toString().trim();
            String password = inputPassword.getText().toString().trim();

            if (TextUtils.isEmpty(email)) {
                Toast.makeText(getApplicationContext(), "Введіть Email!",
                    Toast.LENGTH_SHORT).show();
                return;
            }

            if (TextUtils.isEmpty(password)) {
                Toast.makeText(getApplicationContext(), "Введіть пароль!",
                    Toast.LENGTH_SHORT).show();
                return;
            }

            if (password.length() < 6) {
                Toast.makeText(getApplicationContext(),
                    getString(R.string.minimum_password), Toast.LENGTH_SHORT).show();
                return;
            }

            progressBar.setVisibility(View.VISIBLE);
            //create user
            auth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(SignupActivity.this, new
                    OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult>
                            task) {

```

```

        Toast.makeText(SignupActivity.this, "Користувачу
створений успішно", Toast.LENGTH_SHORT).show();
        progressBar.setVisibility(View.GONE);
        // If sign in fails, display a message to the
        user. If sign in succeeds
        // the auth state listener will be notified and
        logic to handle the
        // signed in user can be handled in the listener.
        if (!task.isSuccessful()) {
            Toast.makeText(SignupActivity.this,
            "Регістрація не виконана", Toast.LENGTH_SHORT).show();
        } else {
            startActivity(new Intent(SignupActivity.this,
            InfoActivity.class));
            finish();
        }
    }
    });
}
});
}

@Override
protected void onResume() {
    super.onResume();
    progressBar.setVisibility(View.GONE);
}
}

```

1.12 WinQuest.java

```

package doublel.studio.tourquest.activity;

import androidx.appcompat.app.AppCompatActivity;

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

import doublel.studio.tourquest.R;

public class WinQuest extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_win_quest);
        Button btn = findViewById(R.id.btnWin);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(WinQuest.this, MenuActivity.class));
            }
        });
    }
}

```

```

    }
    });
}
}

```

2. Пакет adapter

2.1 MyListAdapter.java

```

package double1.studio.tourquest.adapter;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

import double1.studio.tourquest.R;

public class MyListAdapter extends ArrayAdapter<String> {
    Context context;
    String Title[];
    String Description[];
    String Number[];
    String Clock[];
    int Images[];
    String Geo[];

    public MyListAdapter(@NonNull Context context, String title[], String
description[], String number[], String clock[], int images[], String geo[]) {
        super(context, R.layout.style_row, R.id.textViev1, title);
        this.context = context;
        this.Title = title;
        this.Description = description;
        this.Number = number;
        this.Clock = clock;
        this.Images = images;
        this.Geo = geo;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        LayoutInflater layoutInflater =
(LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row = layoutInflater.inflate(R.layout.style_row, parent,false);
        ImageView images = row.findViewById(R.id.image);
        TextView myTitle = row.findViewById(R.id.textViev1);
        TextView myDescription = row.findViewById(R.id.textViev2);
        TextView myNumber = row.findViewById(R.id.textViev3);
    }
}

```

```

        TextView myClock = row.findViewById(R.id.textView4);

        Button btn = row.findViewById(R.id.btnGoPoint);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(context.getApplicationContext(), "Шукаємо
точку...", Toast.LENGTH_LONG).show();
                Uri gmmIntentUri = Uri.parse("geo:0,0?q="+Geo[position]+"(Місце
призначення)");
                Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
                mapIntent.setPackage("com.google.android.apps.maps");
                context.startActivity(mapIntent);
            }
        });

        images.setImageResource(Images[position]);
        myTitle.setText(Title[position]);
        myDescription.setText(Description[position]);
        myNumber.setText(Number[position]);
        myClock.setText(Clock[position]);

        return row;
    }
}

```

2.2 PagerAdapterCafe.java

```

package doublel.studio.tourquest.adapter;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.lifecycle.Lifecycle;
import androidx.viewpager2.adapter.FragmentStateAdapter;

import doublel.studio.tourquest.fragment.BarFragment;
import doublel.studio.tourquest.fragment.CafeFragment;
import doublel.studio.tourquest.fragment.CofeFragment;
import doublel.studio.tourquest.fragment.FastFragment;

public class PagerAdapterCafe extends FragmentStateAdapter {
    public PagerAdapterCafe(@NonNull FragmentManager fragmentManager,
        @NonNull Lifecycle lifecycle) {
        super(fragmentManager, lifecycle);
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        if(position == 0){
            return new CafeFragment();
        } else if(position == 1){
            return new CofeFragment();
        } else if(position == 2){
            return new BarFragment();
        }
        return new FastFragment();
    }

    @Override
    public int getItemCount() {
        return 4;
    }
}

```

```
    }  
}
```

2.3 PagerAdapterClub.java

```
package doublel.studio.tourquest.adapter;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.lifecycle.Lifecycle;  
import androidx.viewpager2.adapter.FragmentStateAdapter;  
  
import doublel.studio.tourquest.fragment.ClubFragment;  
import doublel.studio.tourquest.fragment.FunFragment;  
  
import doublel.studio.tourquest.fragment.MarketplaceFragment;  
  
public class PagerAdapterClub extends FragmentStateAdapter {  
    public PagerAdapterClub(@NonNull FragmentManager fragmentManager, @NonNull  
Lifecycle lifecycle) {  
        super(fragmentManager, lifecycle);  
    }  
  
    @NonNull  
    @Override  
    public Fragment createFragment(int position) {  
        if(position == 0){  
            return new FunFragment();  
        } if(position ==1 ){  
            return new MarketplaceFragment();  
        }  
        return new ClubFragment();  
    }  
  
    @Override  
    public int getItemCount() {  
        return 3;  
    }  
}
```

2.4 PagerAdapterHistory.java

```
package doublel.studio.tourquest.adapter;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.lifecycle.Lifecycle;  
import androidx.viewpager2.adapter.FragmentStateAdapter;  
  
import doublel.studio.tourquest.fragment.ClubFragment;  
import doublel.studio.tourquest.fragment.CultureFragment;  
import doublel.studio.tourquest.fragment.FunFragment;  
import doublel.studio.tourquest.fragment.MarketplaceFragment;  
  
public class PagerAdapterHistory extends FragmentStateAdapter {  
  
    public PagerAdapterHistory(@NonNull FragmentManager fragmentManager, @NonNull  
Lifecycle lifecycle) {  
        super(fragmentManager, lifecycle);  
    }  
  
    @NonNull
```

```

@Override
public Fragment createFragment(int position) {
    return new CultureFragment();
}

@Override
public int getItemCount() {
    return 1;
}
}

```

2.5 PagerAdapterHotel.java

```

package double1.studio.tourquest.adapter;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.lifecycle.Lifecycle;
import androidx.viewpager2.adapter.FragmentStateAdapter;

import double1.studio.tourquest.fragment.BarFragment;
import double1.studio.tourquest.fragment.FastFragment;
import double1.studio.tourquest.fragment.HostelFragment;
import double1.studio.tourquest.fragment.HotelFragment;

public class PagerAdapterHotel extends FragmentStateAdapter {
    public PagerAdapterHotel(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle) {
        super(fragmentManager, lifecycle);
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        if(position == 0){
            return new HotelFragment();
        }
        return new HostelFragment();
    }

    @Override
    public int getItemCount() { return 2; }
}

```

2.6 User.java

```

package double1.studio.tourquest.adapter;

public class User {
    public String name;
    public String age;
    public String city;
    public String male;

    public User() {
        dataSnapshot.getValue(User.class)
    }

    public User(String name, String age, String city, String male) {
        this.name = name;
        this.age = age;
        this.city = city;
        this.male = male;
    }
}

```

```
}  
}
```

3. Пакет fragment

3.1 BarFragment.java

```
package doublel.studio.tourquest.fragment;  
  
import android.os.Bundle;  
  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.ListFragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import doublel.studio.tourquest.R;  
import doublel.studio.tourquest.adapter.MyListAdapter;  
  
public class BarFragment extends ListFragment {  
    @Override  
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {  
        super.onActivityCreated(savedInstanceState);  
        String mTitle[]={"Zivot a Pivo","Robata","Pub CASH","ФАЛЬВАРЕК"};  
        String mDescription[]={"Бар - Европейска - Паб"  
                                ,"Японска - Паб з пивоварнею - Піцца"  
                                ,"Паб"  
                                ,"Паб - Східноєвропейська - Українська"};  
        String mNumber[]={"+380 67 321 6214","+380 63 100 4008","+380 462 605  
942","93-58-53"};  
        String mClock[]={"17:00 - 22:00","11:30 - 01:00","12:00 - 00:00","11:00 -  
00:00"};  
        int  
        images[]={R.drawable.cafe,R.drawable.cafe,R.drawable.cafe,R.drawable.cafe};  
        String geo[]={ "51.49498303165736, 31.30393744851036","51.49077798329214,  
31.300424563446665","51.500615071360095, 31.28421703584785","51.491726001911495,  
31.297441649150468"};  
  
        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,  
mDescription, mNumber, mClock, images, geo);  
        setListAdapter(adapter);  
    }  
}
```

3.2 CafeFragment.java

```
package doublel.studio.tourquest.fragment;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;
```

```

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;

import java.util.zip.Inflater;

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class CafeFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"ФОТЕЛЬ", "ШАРЛОТКА", "АБАЖУР", "ЗОЛОТИЙ КЛЮЧИК", "ВОН
AMI"};
        String mDescription[]={"Десерти - Кафе - Підходить для вегетаріанців"
            , "Десерти - Кафе - Європейська"
            , "Десерти - Кафе - Європейська"
            , "Десерти - Кафе - Європейська"
            , "Кафе"};
        String mNumber[]={"+380 462 647 072", "+380 50 465 0101", "+380 462 603
738", "+380 462 773 047", "+380 93 238 2810"};
        String mClock[]={"9:00 - 22:00", "8:00 - 23:00", "9:00 - 19:00", "8:00 -
21:00", "8:00 - 23:00"};
        int
images[]={R.drawable.cafe, R.drawable.cafe, R.drawable.cafe, R.drawable.cafe, R.drawa
ble.cafe};
        String geo[]={ "51.49059025210853,
31.304884105649474", "51.490578637151366, 31.300714296662605", "51.489342877353664,
31.30335977028785", "51.49150928799578, 31.301193023764867", "51.49021949587248,
31.301227233153327"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.3 ClubFragment.java

```

package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

```

```

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

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class ClubFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"777-Градецкий","Бурштинова кішка","Оранж Бар"};
        String mDescription[]={"Нічний клуб"
            ,"Капоке-бар"
            ,"Бар"};
        String mNumber[]={"+380 63 026 0436","+380 73 444 0919","+380 93 917
9177"};
        String mClock[]={"18:00 - 00:00","19:00 - 6:00","16:00 - 23:00"};
        int images[]={R.drawable.club,R.drawable.club,R.drawable.club};
        String geo[]{"51.502998610449865,
31.281984906226107","51.49045384435712, 31.301496539063606","51.499413056531424,
31.290379593889057"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images,geo);
        setListAdapter(adapter);
    }
}

```

3.4 CofeFragment.java

```

package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

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

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class CofeFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"Coffeemania","Чашка Espresso Bar","Bake House"};
        String mDescription[]={"Кофе - Чай"
            ,"Кофе - Чай"
            ,"Булочна - Кофе - Чай"};
    }
}

```

```

        String mNumber[]={"+380 462 677 771","+380 462 653 280","+380 66 000
1126"};
        String mClock[]={ "7:30 - 21:00", "10:00 - 22:00", "8:00 - 22:00"};
        int images[]={R.drawable.cafe,R.drawable.cafe,R.drawable.cafe};
        String geo[]={ "51.49224185965426, 31.296763105980713", "51.49907809236771,
31.289857268376217", "51.493244676017774, 31.29518682123477"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.5 CultureFragment.java

```

package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

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

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class CultureFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={ "Антонієві печери", "Дитинець", "Троїцький
собор", "П'ятницька церква", "Спасо-Преображенський собор"};
        String mDescription[]={ "Гроти та печери"
            , "Історичне місце"
            , "Церки та собори"
            , "Церки та собори"
            , "Церки та собори"};
        String mNumber[]={ "-", "-", "+380 46 264 5241", "+380 46 297 2721", "+380 46
264 7532"};
        String mClock[]={ "9:00 - 16:30", "00:00 - 00:00", "6:30 - 19:00", "-", "-"};
        int
images[]={R.drawable.history,R.drawable.history,R.drawable.history,R.drawable.his
tory,R.drawable.history};
        String geo[]={ "51.47837060506192, 31.28364681262386", "51.48951360657358,
31.30630343422861", "51.477522784783545, 31.28044293519439", "51.4925864980277,
31.30085979364891", "51.489142030897746, 31.3076482332313"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.6 FastFragment.java

```
package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

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

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class FastFragment extends ListFragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        String mTitle[]={"McDonald's","ЧеБыр Street Food","Чикен Хат","Shaurma Khorovats"};
        String mDescription[]={"Фастфуд",
                                "Фастфуд",
                                "Фастфуд",
                                "Фастфуд"};
        String mNumber[]={"+380 50 463 9146","+380 73 133 3383","+380 46 261 1888","+380 93 788 7000"};
        String mClock[]={"7:00 - 23:45","10:00 - 21:00","8:00 - 00:00","10:00 - 22:00"};
        int
        images[]={R.drawable.cafe,R.drawable.cafe,R.drawable.cafe,R.drawable.cafe};
        String geo[]={"51.49276019803862, 31.291583706951418","51.49219533893309, 31.293467221798196","51.4919027596423, 31.2970982620585","51.49924104241296, 31.289469390338112"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle, mDescription, mNumber, mClock, images,geo);
        setListAdapter(adapter);
    }
}
```

3.7 FunFragment.java

```
package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```

```

import double1.studio.tourquest.R;
import double1.studio.tourquest.adapter.MyListAdapter;

public class FunFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"XRoom - квест в реальности", "Квест-проект
        Ізоляція", "Картинг-центр Картодром"};
        String mDescription[]={"Квест кімнати"
        , "Квест кімнати"
        , "Картинг-центр"};
        String mNumber[]={"+380 63 439 7126", "+380 97 517 3818", "+380 93 956
        9163"};
        String mClock[]={"10:00 - 21:00", "9:00 - 23:30", "13:00 - 21:00"};
        int images[]={R.drawable.club, R.drawable.club, R.drawable.club};
        String geo[]={"51.50147985289576, 31.295075650128393", "51.49321191719858,
        31.304391819929315", "51.54098686256299, 31.23850050662417"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
        mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.8 HostelFragment.java

```

package double1.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.ListFragment;

import double1.studio.tourquest.adapter.MyListAdapter;
import double1.studio.tourquest.R;

public class HostelFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"Hostel Hola", "Хостел Папас", "Лео (Хостел и Баня)"};
        String mDescription[]={"Від 350грн."
        , "Від 300грн."
        , "Ціна не відома"};
        String mNumber[]={"+380 46 297 4710", "+380 93 000 8483", "+380 63 120
        2026"};
        String mClock[]={"-", "-", "-"};
        int images[]={R.drawable.hotel, R.drawable.hotel, R.drawable.hotel};
        String geo[]={"51.47763867247721, 31.275100579261206", "51.49239157885308,
        31.29926296499301", "51.47645633178038, 31.27392719910673"};
    }
}

```

```

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.9 HotelFragment.java

```

package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.ListFragment;

import doublel.studio.tourquest.adapter.MyListAdapter;
import doublel.studio.tourquest.R;

public class HotelFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"RiverSide", "Готель Профспілковий", "Готель
Придеснянський", "Готель Україна"};
        String mDescription[]={"Ціна від 1700грн."
            , "Ціна від 650грн."
            , "Ціна невідома"
            , "Ціна від 750грн."};
        String mNumber[]={"+380 93 604 6464", "+380 46 295 2369", "+380 46 295
4802", "+380 46 265 1400"};
        String mClock[]={"-", "-", "-", "-"};
        int
images[]={R.drawable.hotel, R.drawable.hotel, R.drawable.hotel, R.drawable.hotel};
        String geo[]={ "51.504636296329785, 31.341829423451102", "51.5059261444157,
31.33869329450326", "51.503731150553165, 31.338522378934012", "51.49456459325661,
31.295288534225193"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.10 MarketplaceFragment.java

```

package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

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

```

```

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class MarketplaceFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"ЦУМ Чернігів", "ТРЦ HOLLYWOOD", "ТЦ Рояль"};
        String mDescription[]={"Торговий центр"
            , "Торговий центр"
            , "Торговий центр"};
        String mNumber[]={"-", "-", "-"};
        String mClock[]={"10:00 - 21:00", "10:00 - 21:00", "8:00 - 22:00"};
        int images[]={R.drawable.club, R.drawable.club, R.drawable.club};
        String geo[]={"51.49926441827906, 31.28950157740539", "51.5150986892178,
31.306176610233987", "51.4791320599867, 31.272080369851242"};

        MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,
mDescription, mNumber, mClock, images, geo);
        setListAdapter(adapter);
    }
}

```

3.11 RecreationFragment.java

```

package doublel.studio.tourquest.fragment;

import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;

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

import doublel.studio.tourquest.R;
import doublel.studio.tourquest.adapter.MyListAdapter;

public class RecreationFragment extends ListFragment {
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        String mTitle[]={"Сім книг", "Гном і гном", "За за за"};
        String mDescription[]={"Вкусная еда для тебя и меня для всей семьи и так
далее просто текст для виду но сейчас пойдет я думаю нужно еще немного написать
чего-то но покажет потом", "Вкусная еда для тебя и меня для всей семьи и так далее
просто текст для виду но сейчас пойдет я думаю нужно еще немного написать чего-то
но покажет потом", "Вкусная еда для тебя и меня для всей семьи и так далее просто
текст для виду но сейчас пойдет я думаю нужно еще немного написать чего-то но
покажет потом"};
        String mNumber[]={"+38098999999", "38098659399", "380738894522"};
        String mClock[]={"10:00 - 22:00", "8:00 - 22:00", "9:00 - 23:00"};
    }
}

```

```
int images[]={R.drawable.cafe,R.drawable.cafe,R.drawable.cafe};  
String geo[]={ "51.50548575426945, 31.33526934146993", "51.50548575426945,  
31.33526934146993", "51.50548575426945, 31.33526934146993"};
```

```
MyListAdapter adapter = new MyListAdapter(getActivity(), mTitle,  
mDescription, mNumber, mClock, images, geo);  
setListAdapter(adapter);  
}  
}
```