

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

на тему:

**«Туристичний мобільний застосунок
IvanTurist»**

**Завідувач
випускаючої кафедри**

А. С. Довбиш

Керівник роботи

О. А. Шовкопляс

Студента групи ІН-71

В. А. Міщенко

СУМИ 2021

Сумський державний університет

(назва вузу)

Факультет ЕЛІТ Кафедра Комп'ютерних наук

Спеціальність «Комп'ютерні науки»

Затверджую _____

Зав. кафедри Довбиш А. С.

« _____ » _____ 20__ р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТУ

Міщенко Володимиру Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Туристичний мобільний застосунок IvanTurist

затверджую наказом по університету від « _____ » _____ 20__ р.
№ _____

2. Термін здачі студентом закінченого проєкту (роботи) _____

3. Вхідні дані до проєкту (роботи) _____

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити) 1) Інформаційний огляд сфери туризму. Туризм у світі та вплив пандемії на сферу. Огляд туристичних програмних продуктів, що забезпечують попит користувачів 2) Огляд інструментів для розробки. 3) Вибір методів розробки. 3) Проектування інформаційної системи. 4) Розроблення інформаційного та програмного забезпечення.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Термін виконання проєкту (роботи)	Примітка
1.	Аналіз проблеми. Постановка задачі дослідження	05.04–13.04.2021	
2.	Аналіз інструментів розробки мобільного туристичного додатку	14.04–11.04.2021	
3.	Розроблення структури додатку, планування переходу між інтерфейсами	12.04–19.04.2021	
4.	Розроблення програмної реалізації туристичного додатку	20.04–20.05.2021	
5.	Оформлення пояснювальної записки до дипломної роботи	20.05–04.06.2021	

Студент–дипломник

_____ (підпис)

Керівник проєкту

_____ (підпис)

РЕФЕРАТ

Записка: 77 стор., 24 рис., 4 табл., 1 додаток, 20 джерел.

Об'єкт дослідження – платформа розробки Android Studio.

Мета роботи – розробка мобільного туристичного додатку для системи Android з можливістю перегляду та поширення основної інформації точок іншим користувачам.

Методи дослідження – метод розробки мобільного додатку мовою об'єктно орієнтованого програмування для Android з використанням бібліотек утиліт Maps SDK та бази даних Firebase.

Результати – розроблено туристичний додаток орієнтований на вітчизняного споживача для дослідження теренів країни. У додатку користувач зі смартфона з доступом в інтернет має можливість переглядати рекомендації інших туристів та ділитися геопозицією та описом локації, що здається їм цікавою, в реальному часі.

При перегляді основної інформації про обрану точку користувач бачить географічне розташування на вбудованому фрагменті Google карти, який за потреби може перенаправити в мобільний додаток Google maps для побудови зручного маршруту.

Особливістю реалізації є можливість додавати точки до переліку вподобань для швидкого доступу та перегляд найпопулярніших точок в офлайн режимі.

Зміст

ВСТУП	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
1.1 Сфера туризму	8
1.2 Світова пандемія та туризм.....	8
1.3 Внутрішній туризм в Україні.....	9
1.4 Огляд інструментів для мандрівок.....	9
1.5 Maps SDK для Android.....	11
1.6 Найпопулярніші бібліотеки Maps SDK	11
1.6 Постановка задачі	12
2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ	14
2.1 Вибір засобів програмної реалізації.....	14
2.2 Вибір середовища розробки.....	14
2.3 Вибір бази даних.....	16
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	17
3.1 Проектування інформаційної системи.....	17
3.3 Програмна реалізація інформаційної системи.....	29
4 Аналіз роботи програми	35
4.1. Користувацький інтерфейф	35
4.2 База даних Firebase	42
ВИСНОВОК	44
СПИСОК ЛІТЕРАТУРИ.....	45
ДОДАТОК. ЛІСТИНГ КЛАСІВ МОБІЛЬНОГО ДОДАТКУ.....	48

ВСТУП

Україна є найбільшою за площею країною Європи, що містить у собі тисячі кілометрів річок, сотні природничих парків, мальовничі гори та вихід до двох морів, а багата історична спадщина створює додаткові сприятливі умови для розвитку туристичної галузі в Україні [1].

Туризм є найрозвинутішою сферою розваг, а також елементом культурної дипломатії. Він є одним з основних центрів для подальшого економічного зростання. Доказом цієї тези може слугувати дослідження Всесвітньої ради подорожей і туризму (WTTC) за яким частка від світового ВВП, що припадала на туризм у 2018 році складає 10,4%. Беззаперечним лишається і факт впливу пандемії на світовий туризм, згідно дослідження World Tourism Organization (UNWTO) у II кварталі 2020 року обіг туристів зменшився на 94,6%, а кожен наступний квартал більш ніж на 70% у порівнянні з відповідним періодами у 2019. Особливих втрат зазнала сфера міжнародних подорожей, всупереч цьому обіг туризму в межах країни демонструє позитивні зміни в кількісному показнику, що створює додатковий попит на пов'язаний з цією сферою товар. [2].

Амбасадорами галузі поширення інформації про маловідомі місця України як самих українців, так і для іноземців, що мандрують Україною є Ukraïner, як зазначили організатори проекту «Ukraïner розповідає пізнавальні історії про найвіддаленіші куточки, людей, мистецтво та бізнес в Україні». Вони демонструють різність та цікавість теренів через експедиції регіонами та зйомками блогів, які згодом презентуються як окремі відео на Youtube, повно екранне кіно, промо ролики у аеропортах та поїздах Intercity. Також одними з провідних діячів цієї галузі відкритів України є Youtube канали "Хаці" та "Таємна мапа", популярність яких доводить існування запиту серед громадян на розробки, пов'язані з даною темою.

Додатковим підтвердженням актуальності цієї роботи також може бути той факт, що згідно з даними Дослідницького холдингу Factum Group Ukraine у III кварталі 2019 року 66% регулярних входів до інтернету відбувається через смартфон, що більш ніж у 1,5 більше регулярних входів через стаціонарний домашній ноутбук чи комп'ютер[3].

Таким чином, метою даної роботи є розробка туристичного додатку для мандрівника, що слугуватиме кишеньковим порадиником у подорожах та відкриттях, з можливістю ділитися своїми точками і переглядати точки інших.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Сфера туризму

Туризм - це сфера людської діяльності пов'язана з тимчасовим виїздом за межі постійного перебування з метою відпочинку, дослідження чи професійно-діловою метою, що задовольняє потребу у зміні оточення, емоціях й допомагає знайти відповідь на духовні потреби людини.

Туризм розглядається в аспекті фізичного, емоційного та розумового виховання, подорожі сприяють:

- аналітичному мисленню та плануванню;
- розвитку колективної та особистої відповідальності;
- стресостійкості та гнучкості розуму;
- фізичній витривалості;
- задоволенню потреби в особистих відкриттях та нових знаннях;
- комунікабельності;
- заохоченню вивчення іноземних мов;
- патріотизму (через краєзнавство[4] та подорожі теренами країни).

1.2 Світова пандемія та туризм

Зважаючи, що за даними "Всесвітньої ради подорожей та туризму" (WTTC) туризм приносив 8,8 млрд доларів, а кожен десятий житель планети є працівником цієї галузі - пандемія привнесла великі корективи із закриттям кордонів, а згодом діючи за правилом "Впускай та контролюй", де транспорт, готелі, оздоровчі комплекси мають суворі кількісні та епідеміологічні обмеження.

Міжнародний туризм знаходиться в кризі, особливо в країнах де це є основою економіки [5], також оприлюднено невтішний прогноз ООН щодо відновлення туристичної галузі в світі, вустами генерального секретаря António Guterres[6], що роботу може втратити близько 120 млн осіб.

Натомість почав активніше розвиватися внутрішній туризм, а відмінністю нової епохи в туризмі є більший попит на кемпінги.

1.3 Внутрішній туризм в Україні

Як визначають експерти галузі туризму та туроператори сезону 2020 український споживач обирає вітчизняні курорти, туристичні маршрути в межах країни - значно більше, це підтверджується гео-аналітика мобільних пристроїв на оприлюднена "Vodafone Україна"[7] про відвідування Південних областей і моря влітку 2020 року у порівнянні з відповідним сезоном у минулому році.

Беззаперечним є той факт, що ці данні є наслідком закриття кордонів, але попри все, дана ситуація надає додаткову можливість покращення сервісу, популяризації маловідомих місць та створення тих умов та цін для розвитку бізнесу по всій країні, що здатні витримати конкуренцію, адже після спаду пандемії та відкриття широкого доступу до подорожей закордоном, вірогідно, рівень відвідуваності знову значно знизиться з причин нерівноцінності умов транспортного забезпечення, ціноутворення, доступності туристичних локацій та якості інформування в медіа, що впливає на обізнаність населення про ті чи інші туристичні локації.

1.4 Огляд інструментів для мандрівок

Ще до пандемії широку популярність отримали мандрівки сплановані самотійно чи групою, що включають в себе дослідження регіону і створення для списків обов'язкових місць для відвідувань. Маршрути можна будувати

самостійно, або скористатися вже розробленими порадиниками, як наприклад "ТОП-60 найцікавіших куточків України" [8].

Для планування мандрівок самостійно використовуючи смартфон можливо користуватися різними веб ресурсами, розгляньмо їх основні переваги й недоліки (табл.1).

Таблиця 1. Загальне порівняння інструментів планування маршрутів

Інструмент	Перевага	Недоліки
Веб-карти (приклад: Maps.me, Google maps та інші)	+зручність використання; +можливість побачити що поруч; +перегляд офлайн; +можливість додати до вподобань на карті з прив'язкою до авторизованої електронної пошти; +перегляд транспортного сполучення та інфраструктури;	-відсутність рекомендацій; -необхідність підтримки PlayMarket, AppStore; - залежність від заряду батареї смартфона;
Форуми мандрівників та сайти порадиники (приклад: "Veterdoit", "andytravelclub")	+можливість прочитати поради з логістики, сезону візиту; +можливість додати до закладок браузера.	-необхідність пошуку точки на карті самостійно; -потреба в підключенні до інтернету; -додаткові витрати часу на пошук.
Паперові путівники (приклад: карти міст, заказників)	+зібрання усього найцікавішого у певному регіоні.	-відсутність відгуків; -відсутність навігації; -необхідність пошуку примірника путівника.
Карти гірських маршрутів. (приклад: додаток: "Про походи")	+навігація; +довідник номерів рятувальної служби; +правдивість відображення туристичних маршрутів згідно маркування; +можливість роботи офлайн.	-обмеженість регіоном; -відсутність відгуків; -неможливість доповнювати своїми маршрутами; - залежність від заряду батареї смартфона

1.5 Maps SDK для Android

Утиліти — це сервісна програма для забезпечення роботи, діагностування й усунення проблем в системі, керування файлами.

Використовуючи в Android застосунку бібліотеку утиліт MapsSDK, який підтримує мову програмування Java, стає доступним використання карт Google та можливостей пов'язаних з додаванням маркерів, налаштування зовнішнього вигляду та масштабу карт, додавати фігури, реакцію на маніпуляції [9].

Перевагою даної бібліотеки є відкритість коду та різність запропонованих бібліотек утиліт [10].

1.6 Найпопулярніші бібліотеки Maps SDK

Таблиця 2. Найпопулярніші бібліотеки MapsSDK

Опис	Назва
<p>Дозволяє відмальовувати на додатковому слої поверх карти усі об'єкти в GeoJSON вигляді.</p> <p>Використовуючи методи, на слой можна додавати об'єкти або їх видаляти, які будуть відображатися</p>	GeoJSON
<p>Інструмент дозволяє перетворити в географічні об'єкти KML і додати їх окремим шаром поверх карти.</p>	KML Reference
<p>Утиліта яка допомагає продемонструвати скупчення точок на карті в одній точці замість накладання великої кількості окремих маркерів, де розмір зони відповідає за покриття точками, а колір за інтенсивність точок.</p>	Теплова карта
<p>Втілює побажання додати на карту маркер з виноскою, адже звичайні маркери не</p>	IconGenerator

можуть містити контент (зображення, анімація, текст), який налаштовується (стиль, колір, розмір, анімація).	
<p>Дозволяє додавати кластер - як об'єднавчий елемент маркерів, аби не втрачати зручність карти при великій кількості маркерів.</p> <p>Кластер показує кількість маркерів в околі, а при зменшенні масштабу карти кластер розпадається на окремі кластери що об'єднують елементи в околі.</p>	ClusterMarker
<p>Дозволяє кодувати ламані лінії та багатокутники та додавати візуальних складових.</p> <p>Об'єкт описують точки що складаються з географічної довготи та широти, що зберігаються у вигляді кодованої строки із можливістю декодувати в послідовність об'єктів LatLng.</p>	PolyUtil
<p>Використовується для обчислення відстані між точками по прямій, промаркованого маршруту, периметр об'єктів із заданими координатами.</p>	SphericalLocalUtil

1.6 Постановка задачі

Розробити туристичний додаток використовуючи мову програмування Java з використанням можливостей MapsSDK і має мінімальний функціонал, а саме:

- мати доступ до найпопулярніших рекомендованих точок в режимі офлайн;
- дивитися опис та основну інформацію про точки запропоновані іншими користувачами з реальному часі;
- ділитися своїми рекомендаціями з іншими користувачами в реальному часі;

- залишати користувацькі коментарі то точок і переглядати коментарі інших користувачів;
- додавати точки до вподобань і мати їх перелік.

2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1 Вибір засобів програмної реалізації

Для реалізації мобільного додатку на Android була вибрана мова програмування Java.

Java - розвинена об'єктно-орієнтована мова програмування загального призначення. Вона використовується для розробки додатків для мобільних пристроїв, ігрових консолей, стаціонарних та суперкомп'ютерів, обробки великих об'ємів даних, вбудованих систем тощо[11]. Запозичення вигляду і напрацювань відбулося від мови C++, але більш зручна і розширена великою кількістю вбудованих інструментів, фреймворків та сторонніх бібліотек, що робить розробку більш комфортною і продуктивною. Перевагою Java є також JavaVirtualMachine, який дозволяє запускати програму на всіх сучасних пристроях і відповідає правилу «Write once, run anywhere» [12]. Універсальність мови завоювала прихильність великої кількості розробників, станом на 2021 рік Java посідає перше місце у Back-endсфері (23,4%), перше місце у використанні Quality Assurance (39,3%) [13], в інших сферах більш посередні місця через велику конкуренцію та більшу зручність написання коду для спеціальних сфер іншими мовами [14]. Популярність даної мови надає можливість входу в галузь новачкам, шляхом вивчення мови через велику кількість курсів, форумів, а також ком'юніті що розбирає разом, знаходить шляхи вирішення питань які виникають на етапах вивчення та етапах розробки.

2.2 Вибір середовища розробки

Для розробки мобільного застосунку обране середовище AndroidStudio, яка є офіційним для створення додатків для Android. Вона була заснована на основі IntelliJ IDEA, і крім потужного редактора коду та інструментів для

розробки Android Studio пропонує додаткові функції, що підвищують продуктивність при створенні програм для Android[15], такі як :

- гнучка система побудови на основі Gradle;
- швидкий та багатофункціональний емулятор;
- інструменти для тестування;
- підтримка C++ та NDK;
- вбудована підтримка Google Cloud Platform , що спрощує інтеграцію Google Cloud Messaging та App Engine;
- єдине середовище, для розробки всіх пристроїв Android;
- шаблони коду та інтеграція з GitHub;
- інструменти для вирішення проблем продуктивності, зручності використання, сумісності версій, тощо.

2.3 Вибір бази даних

Для зберігання даних, їх отримання на пристрій користувача та можливості запису нових показників обрано базу даних Firebase, що є продуктом Google, в якій данні знаходяться у хмарному сховищі у форматі JSON і синхронізуються в режимі реального часу з кожним підключеним клієнтом. ReactNativeFirebase забезпечує вбудовану інтеграцію з Android та iOSFirebaseSDK [16].

Найчастіше використовується для роботи в реальному часі чатів та мобільних ігор. Робота в реальному часі забезпечується через посилання на конкретні вузли, в свою чергу вони можуть бути підвузлами. З допомогою цього інструменту працюють інтернет видання The New York Times та The Economist, ігри від Gameloft, Halfbrick.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Проектування інформаційної системи

Розробка інформаційної системи починається з проектування, опису ще неіснуючого продукту. Задача проектування описати об'єкти бази даних, ланцюг переходів між користувацькими екранами; класи, що передають та отримують данні, продумати дизайн додатку, написання програми.

Діаграми UML

Діаграма UML (Unified Modeling Language) - уніфікована мова для проектного моделювання, її використання демонструє взаємодію з системою, відображає різні випадки користування [17], в яких бере участь користувач. Діаграми UML використовуються для [18]:

- визначення та організації функціональних вимог у системі;
- розподіл завдань та контроль виконання;
- проектування інформаційних систем;
- передачі проекту до розробки;
- моделювання базового потоку подій у випадку використання;
- представлення цілей взаємодії між системою і користувачем.

Найпоширеніші види діаграм UML наступні:

- діаграма прецедентів;
- діаграма класів;
- діаграма активностей;
- діаграма послідовності;
- діаграма розгортання;
- діаграма співробітництва;
- діаграма об'єктів;
- діаграма станів.

Розглянемо діаграму прецедентів (рис. 3.1), що представляється трьома учасниками:

- розробник, що грає роль лише на етапі розробки та запуску проекту: проектування інформаційних систем;
- користувач, що безпосередньо виконує основну роль і споживача, і виробника нового контенту для додатку;
- База даних, яка підключається автоматично при відкритті додатку та надає користувачу точки і отримує від нього точки на запис.

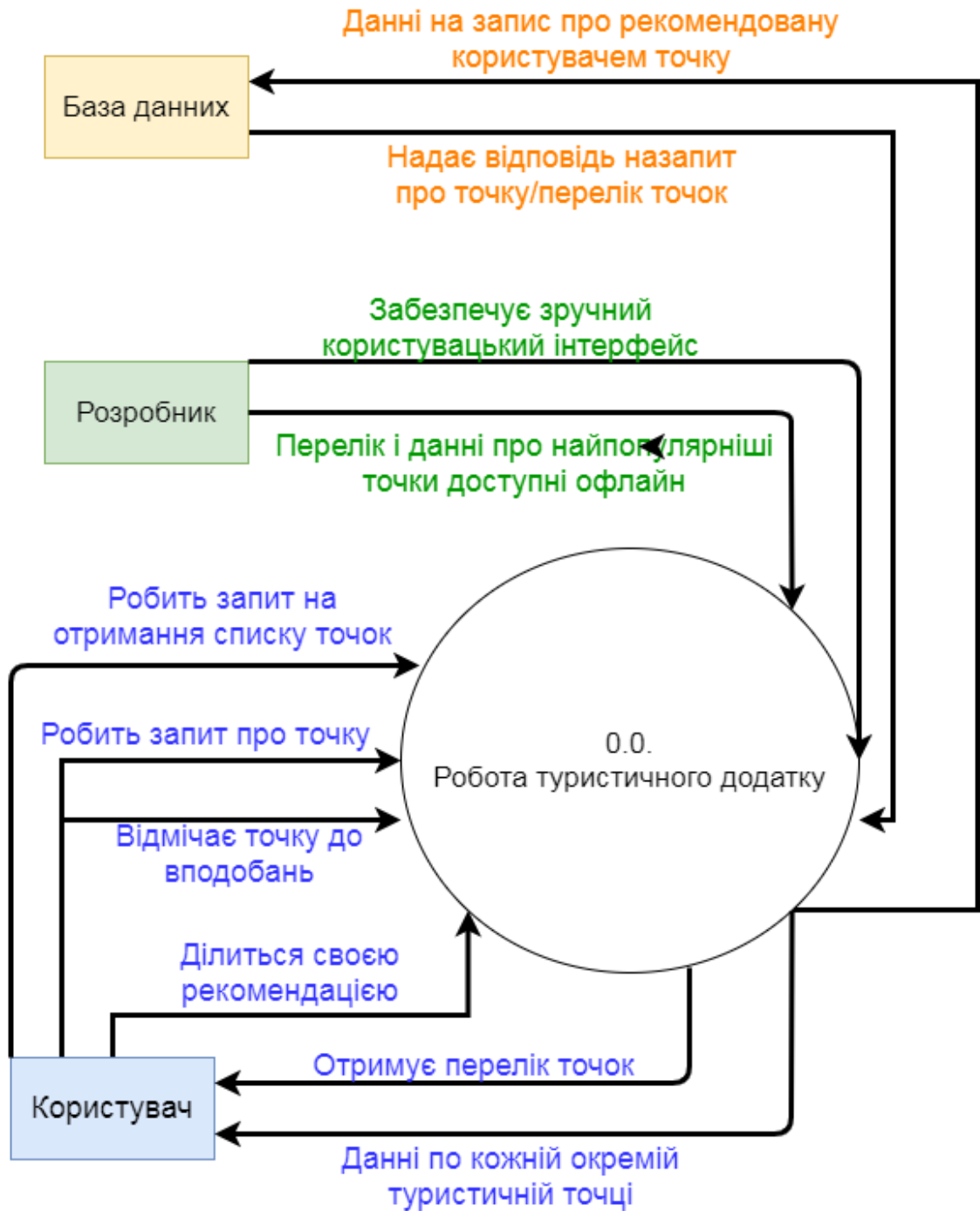


Рисунок 3.1. Діаграма прецедентів

Діаграма сутність – зв'язок

Entity-relationship diagram (далі ER-діаграма) – візуальне представлення, що являє собою різновид блок-схем з відображенням зв'язку між сутностями (люди, місця, події, об'єкти, тощо) в системі або базі даних. Сутності об'єднуються в групи й мають мати однакові характеристики й можуть брати участь в подібних зв'язках.

ER-діаграми використовуються для побудови дизайну реляційних баз даних.

В нашій системі є чотири суб'єкти, які використовуються в системі, побудуймо діаграму «сутність – зв'язок» для чотирьох сутностей, що зберігатимуться в базі даних (рисунок. 3.2).

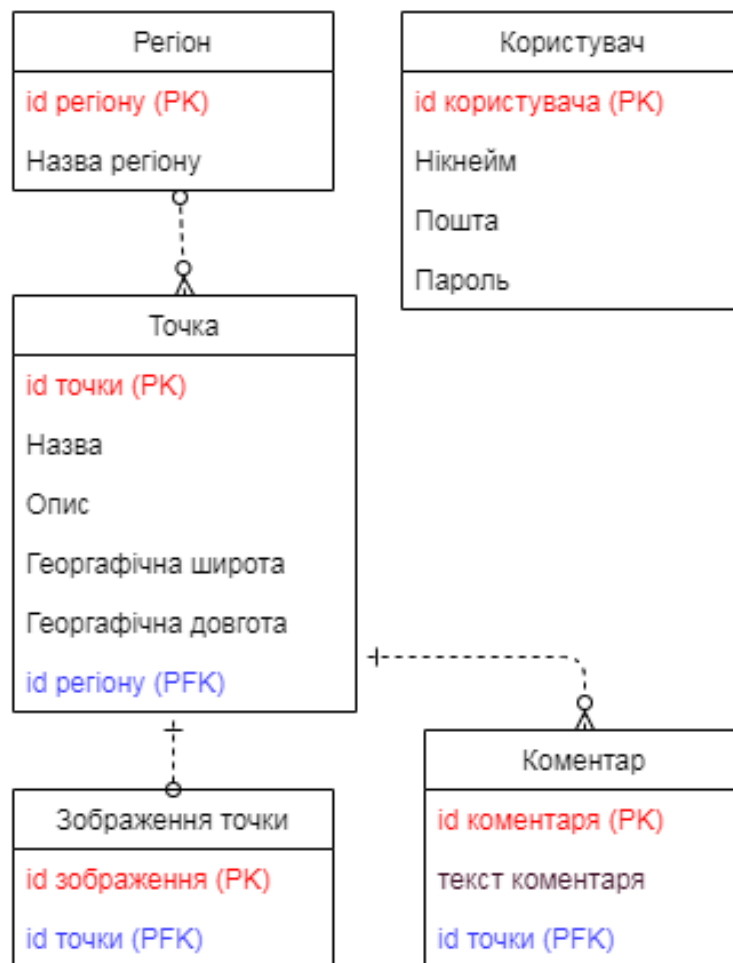


Рисунок 3.2. ER-діаграма

Діаграма класів

У проектуванні програмної частини буде доречно використати діаграму класів, що містять в собі перелік загальних атрибутів та методів, що являють в собою статичні показові об'єкти моделі [17].

Класи на діаграмі позначаються прямокутником з назвою класу вгорі, з атрибутами у верхній половині та методами у нижній половині.

Між собою класи поєднують різні типи зв'язків, а саме:

- асоціація (однонаправлений деякий зв'язок між класами);
- агрегація (один із об'єктів класу включає в себе деякі об'єкти іншого класу);
- композиція (більш строгий варіант агрегації, має жорстку залежність часу існування екземплярів класу контейнера та екземплярів класів що містяться в ньому).

Зобразимо діаграму класів мобільного-додатку для Android.

Для запису користувацьких точок через форму у базу даних використовують клас `Shere_Point_Activity`, для зчитування усіх точок з бази даних та виводу їх переліку викорисовуємо клас `Watch_users_point_Activity` (рисунок 3.3). Інформація про усіх зареєстрованих користувачів записуються в базу даних завдяки формі реєстрації та класу `Registration_Activity` (рисунок 3.4).

Додавання до списку "Уподобання", видалення з нього та оновлення списку точок відбувається завдяки класу `DataAdapter` та інтерфейсу `RecOnClickListener`, який реагує на натискання відповідного елемента за позицією його в `RecyclerView` (рисунок 3.5)

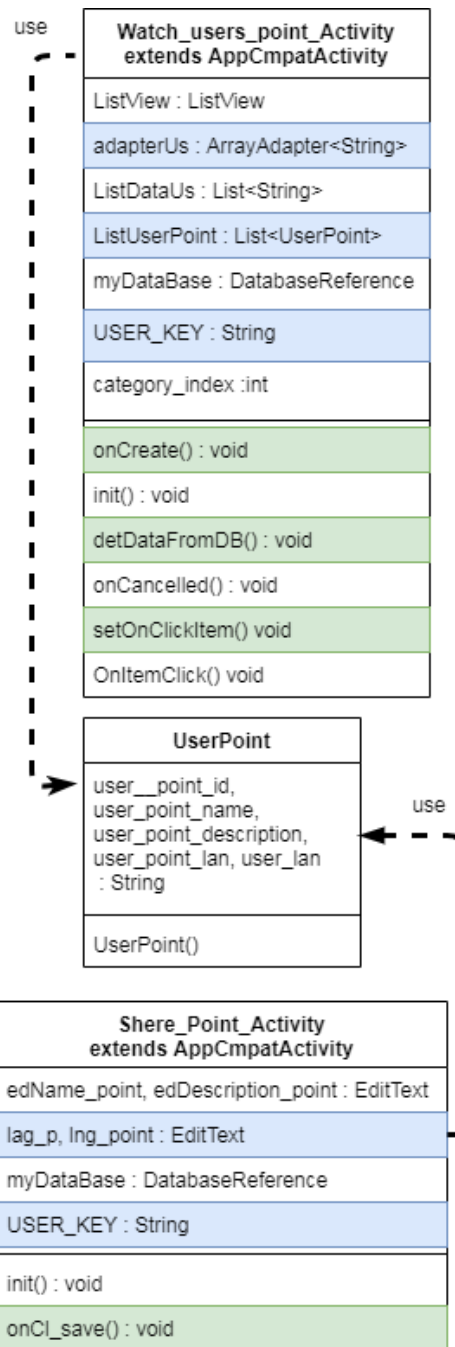


Рисунок 3.3. Діаграма класів роботи з користувацькими точками

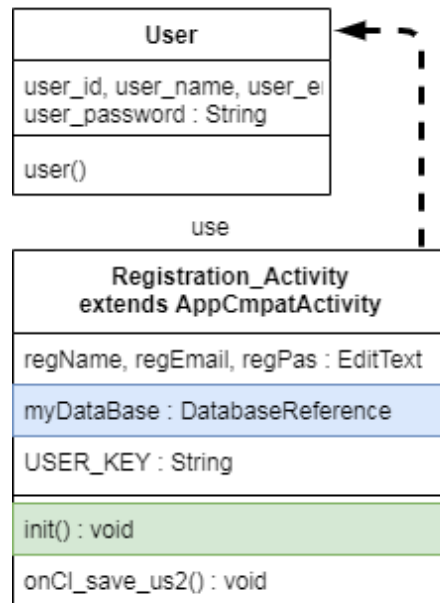


Рисунок 3.4. Діаграма класів реєстрації користувача

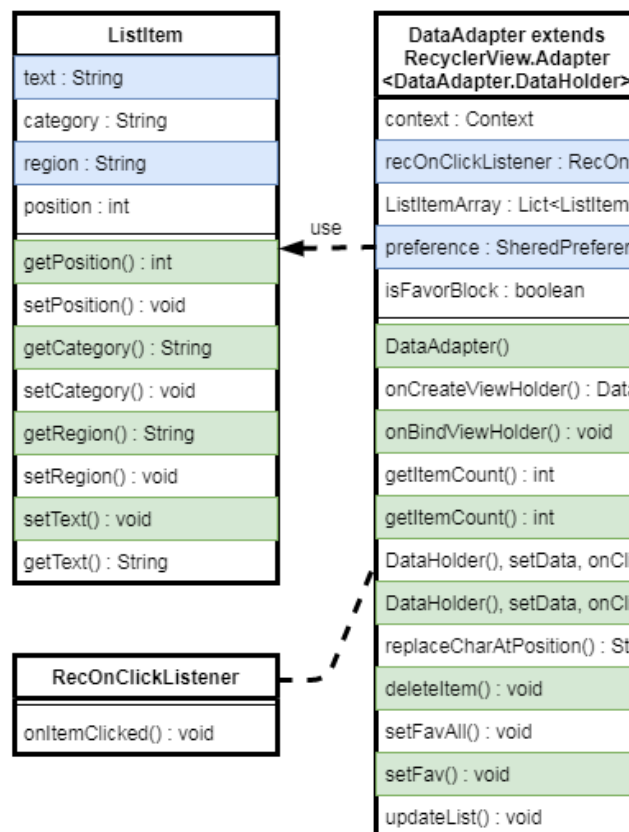


Рисунок 3.5. Діаграма класів заповнення списків переліком точок

Constant
USER_KEY : string
LAT_POINT :double
LNG_POINT :double
CATEGORY_NAME :String

Рисунок 3.6. Діаграма класу констант

MainActivity extends AppCompatActivity implements Navigation.OnNavigationItemSelectedListener
draver : DrawerLayout
ListFDData : List<ListItem>
recOnClickListener : RecOnClickListener
adapterF : DataAdapter
rcView : RecyclerView
checkCategory_forFavor : String
pref : SharedPreferences
arrey_lat_point : double[]
arrey_lng_point : double[]
category_index : int
toolbar : Toolbar
onCreate() : void
onItemClick() : void
onCreateOptionsMenu() : boolean
onNavigationItemSelectedListener() : boolean
onCl_shere_point() : void
onCl_registretion() : void
updateFavorite() : void
init() : void
updateMineList() : void
srtRecOnClickListener() : void
onItemClicked() : void
replaceCharAtPosition() : String
saveString() : void

Рисунок 3.7. Діаграма класу MainActivity

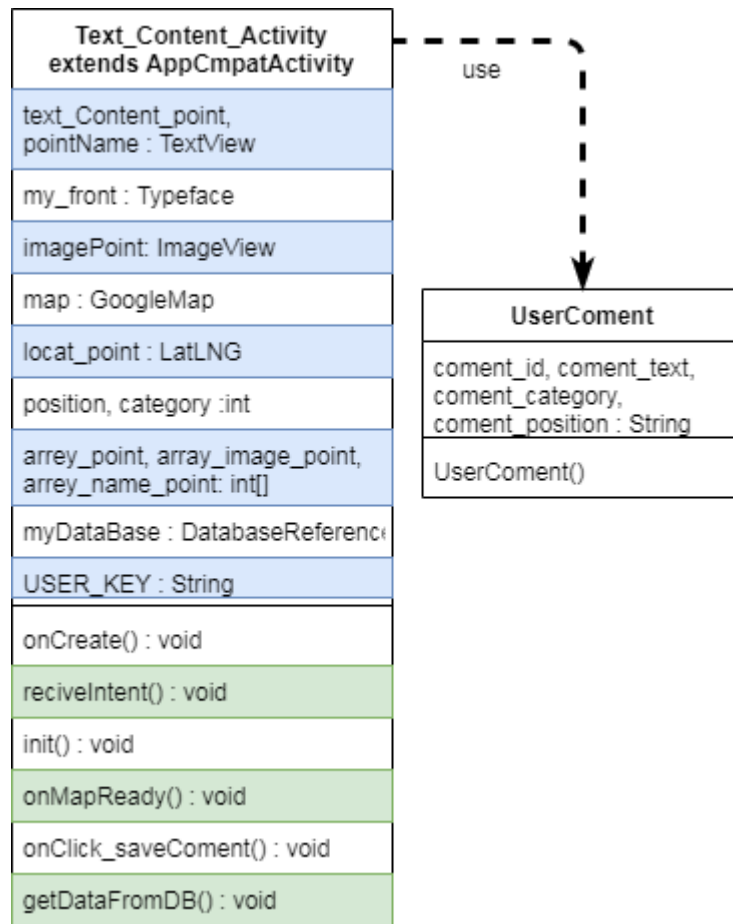


Рисунок 3.8. Діаграма класів відображення всієї інформації про точку та коментарів користувачів про неї

На рисунку 3.8 зображено діграму клас `Text_Content_Activity`, що заповнює інтерфейс даними про точку, а саме зображення, опис, назва, мата з координатами і коментарі до точки. Клас `UserComent` є шаблоном екземпляру типу `UserComent`, що використовується для збереження та отримання з Бази даних коментарів.

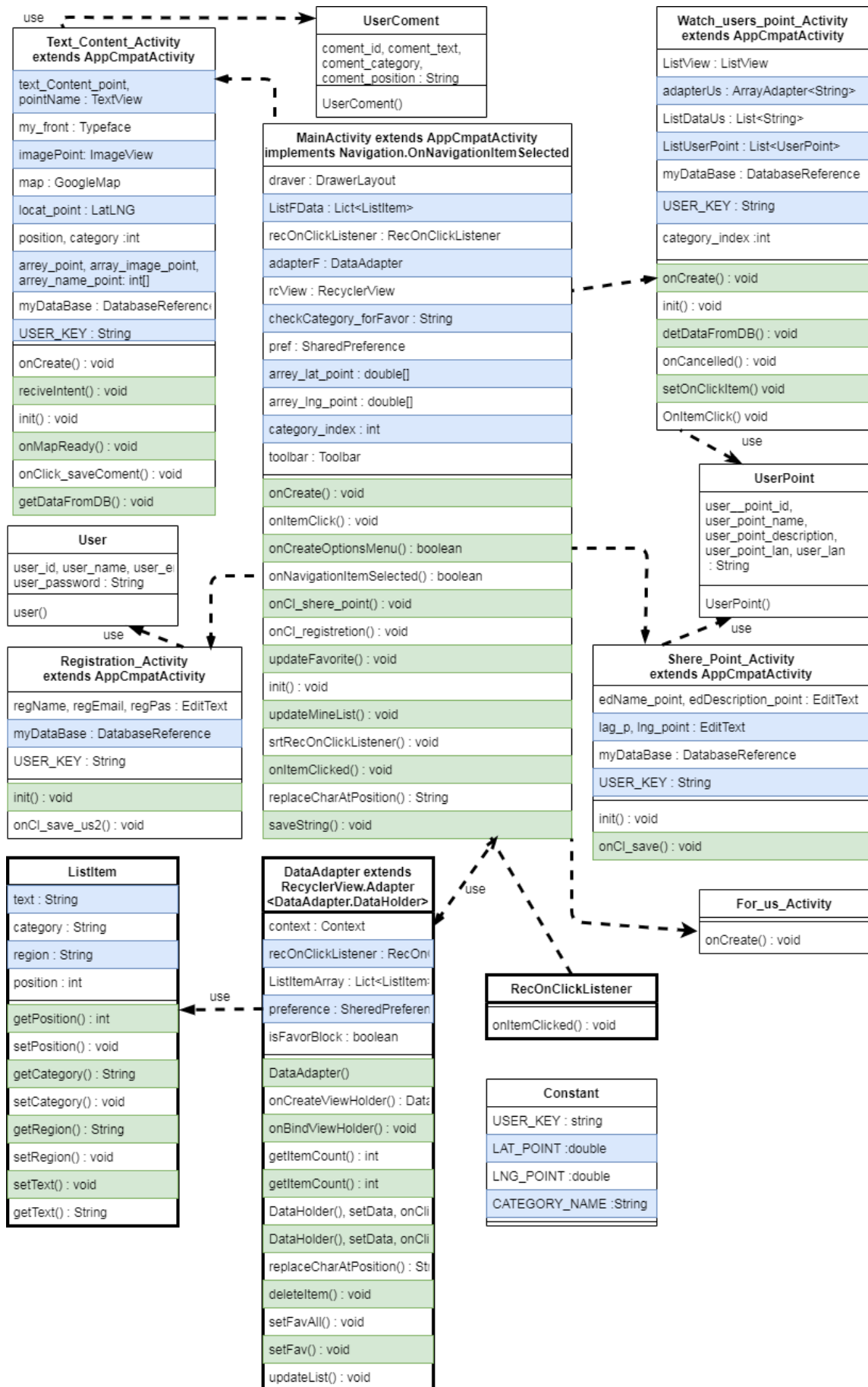


Рисунок 3.9. Діаграма класів додатку із зв'язками

Таблиця 3. Таблиця Activity додатку

Назва	Коротка довідка про клас
MainActivity	Головний клас додатку
Home_content_Activity	Клас, що викликає макет з користувацькою інформацією
For_us_Activity	Клас для переходу на екран інформації про розробника.
MainActivity	Головний клас, який завантажується при відкритті додатку.
Registration_Activity	Клас що виводить на екран форму реєстрацію користувача.
Share_Point_Activity	Клас що виводить на екран форму для реєстрації користувацької точки.
Text_Content_Activity	Клас, що виводить на екран усі данні про підтверджену точку
User	Клас для ініціалізації екземплярів типу User
UserPoint	Клас для ініціалізації екземплярів типу UserPoint
Watch_users_point_Activity	Клас, що виводить на екран усі точки запропоновані користувачами
DataAdapter	Клас, який: 1. заповнює та оновлює список RecyclerView 2. оброблює натиснення на елементи item_layout 3. додає і видаляє з списку "Уподобань".
ListItem	Клас ініціалізації екземплярів типу ListItem
RecOnClickListener	Інтерфейс, слухає натисень на RecyclerView
Constain	Клас, що зберігає константи
UserComent	Клас для ініціалізації екземплярів типу UserComent

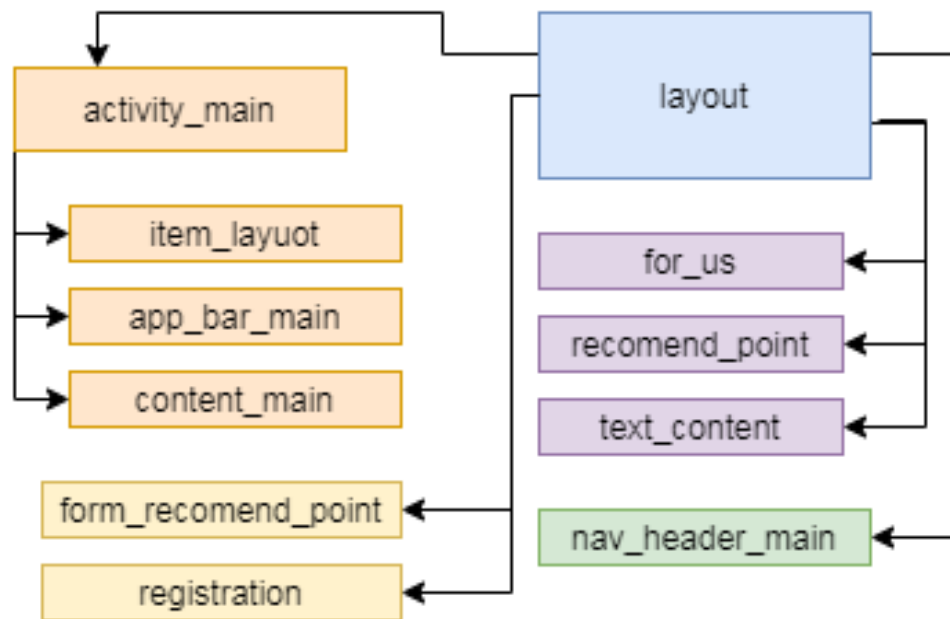


Рисунок 3.10. Діаграма екранів додатку

Таблиця 4. Таблиця макетів додатку

activity_main	Макет головного екрану
item_lyaout	Макет одиниці списку RecyclerView, що використовується дублюючись для відображення списку точок
app_bar_main	Панель інструментів
content_main	Макет початкового екрану
form_recomend_poind	Форма реєстрації користувачьких точок
registration	Форма реєстрації
nav_header_main	Макет заголовку
for_us	Макет, що заповнений інформацією про проект та розробників Макет шапки макетів з назвою додатку
recomend_point	Макет зі списком усіх точок користувачів (записаних через форму)

text_content	Макет, що містить основну інформацію про підтверджену точку
--------------	---

3.3 Програмна реалізація інформаційної системи

Програмна реалізація туристичного мобільного додатку, головним класом є MainActivity, давайте розглянемо функції та методи[19], без яких додаток не виконуватиме основні свої функції.

Для додавання точки до вподобань написано функцію setRecOnClickListener, що запускається в методі onCreate. Функція реагує на натискання на ImageView елементу ItemLayout в RecyclerView, передаючи свою позицію.

Функція onItemClick () – передає позицію елементу RecyclerView до класів.

```
private void setRecOnClickListener()
{
    recOnClickListener = new RecOnClickListener() {
        @Override
        public void onItemClick(int position)
        {
            String tempCat = pref.getString(checkCateg_forFavor, "none");
            Log.d("MyLOG", "Saved data favorite: " +
pref.getString(checkCateg_forFavor, "none"));
            if(tempCat != null)
            {
                if(tempCat.charAt(position) == '0')
                {
                    saveString(replaceCharAtPosition(position, '1',
tempCat)); }
                if(tempCat.charAt(position) == '1')
                {saveString(replaceCharAtPosition(position, '0',
tempCat));}
            }
        }
    };
}
```

Функція `replaceCharAtPosition ()` – заміняє символ в `String` рядку, який має кількість символів відповідну до кількості точок в певній групі, де 1 позначає, що елемент відмічено вподобаним. Після цього функція `saveString()` зберігає рядок в пам'яті додатку, аби при наступному сеансі після закриття відновити список точок "Уподобання".

```
private String replaceCharAtPosition(int position, char ch, String st)
{
    char [] charArray = st.toCharArray(); //розкладання слова на
СИМВОЛИ
    charArray[position] = ch;
    return new String(charArray);
}

private void saveString(String stToSave)
{
    SharedPreferences.Editor editor = pref.edit();
    editor.putString(checkCateg_forFavor, stToSave);
    editor.apply();
    Log.d("MyLOG", "Saved data favorite: " +
pref.getString(checkCateg_forFavor, "none"));
}
```

Функція оновлює і перезаповнює список "Уподобане" відповідно до точок які відмічені символом "1":

```
private void updateFavororite()
{
    List<ListItem> listFavor = new ArrayList<>();
    List <String[]> listData = new ArrayList<>();
```

```

listData.add(getResources().getStringArray(R.array.all_point_array));

listData.add(getResources().getStringArray(R.array.regions_array));

String [] cat_array = {"all_point_array", "regions_array"};

for(int i= 0; i<listData.size(); i++)
{
    for(int j = 0; j < listData.get(i).length; j++)
    {
        String d = pref.getString(cat_array[i], "none");

        if(d!= null) if(d.charAt(j)=='1')
        {
            ListItem item = new ListItem();

            item.setText(listData.get(i)[j]);

            item.setPosition(j);

            item.setCategory(cat_array[i]);

            listFavor.add(item);

        }
    }
}

adapterF.updateList(listFavor, true);

}

```

Для запису користувацької точки в базу даних спочатку ініціалізуємо текстові поля і підключаємося до бази даних Firebase, далі перевіряємо чи всі поля заповнені й перетворюємо їх в тип String і записуємо до бази даних

```
private void init()
```

```

    {

        edName_point=findViewById(R.id.Text_1_recPointName);

        edDescription_point=findViewById(R.id.editText_2_about);

        lag_p=findViewById(R.id.editText_3_lon);

        lng_p=findViewById(R.id.editText_3_lng);

        myDataBase =
        FirebaseDatabase.getInstance().getReference(USER_KEY);

    }

    public void onClick_save(View view)

    {

        String id= myDataBase.getKey();

        String name = edName_point.getText().toString();

        String description =
        edDescription_point.getText().toString();

        String lan = lag_p.getText().toString();

        String lng = lng_p.getText().toString();

        UserPoint newUser = new UserPoint(id, name, description, lan,
        lng);

        if (!TextUtils.isEmpty(name) &&
        !TextUtils.isEmpty(description) && !TextUtils.isEmpty(lan) &&
        !TextUtils.isEmpty(lng))

        {

            myDataBase.push().setValue(newUser);

            Toast.makeText(this, "Точка зареєстрована, дякуємо",
            Toast.LENGTH_SHORT).show();

        }

    }

```



```

        else{Toast.makeText(this, "Не всі поля заповненні",
Toast.LENGTH_SHORT).show();}

    }

```

Метод `setOnClickListener` класу `Watch_users_point_Activity` що він відслідковує натиск на елемент списку користувачьких рекомендованих точок - отримує позицію елемента і відкриває шар шаблону `Text_Content_Activity` і заповнює його даними, отриманими перед цим з бази даних і переданих іншому Activity[20], використовуючи об'єкт `Intent` його метод `putExtra` по назві типу `String` ("position", "point_name", "description, lat та інші)

```

private void setOnCkickItem() {

    listViewUs.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

        @Override

        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

            UserPoint userPoint = listUserPoint.get(position);

            Intent intent = new
Intent(Watch_users_point_Activity.this, Text_Content_Activity.class);

            intent.putExtra("category", category_index);

            intent.putExtra("position", position);

            intent.putExtra("point_name",
userPoint.user_point_name);

            intent.putExtra("point_descriotion",
userPoint.user_point_discription);

            intent.putExtra("lat",
Double.valueOf(userPoint.user_point_lan));

            intent.putExtra("lgn",
Double.valueOf(userPoint.user_point_lng));

```

```
        startActivity(intent);  
    }  
});  
}  
  
point_name.setText(array_name_point[position]);  
  
LAT=i.getDoubleExtra("lat", 0);  
  
LNG=i.getDoubleExtra("lng", 0);  
  
title_point = getString(array_name_point[position]);  
  
break;
```

В той же час `ActivityText_Content_Activity` отримує дані, передані йому, і заповнює їм макет у відповідні поля назву якого він отримує з методу `getExtra`.

4 Аналіз роботи програми

4.1. Користувацький інтерфейф

Коли вперше запускається додаток на телефоні ми отримуємо екран (Рисунок 4.1) де ми маємо можливість зареєструватися, перейти до екрану рекомендованих точок, перегляду точок інших користувачів, переглянути інформацію про розробників.

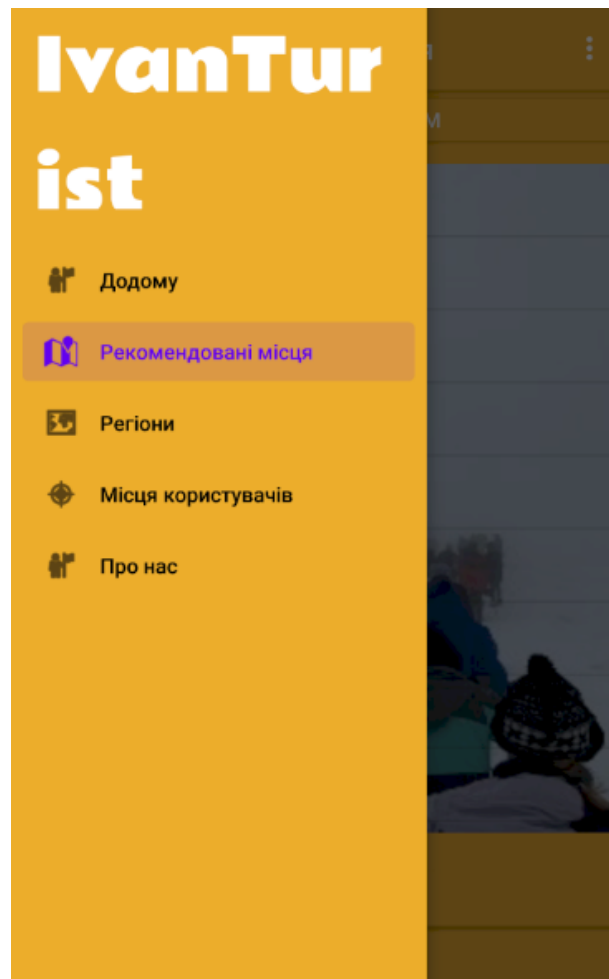


Рисунок 4.1. Екран додатку з висувним меню користувача

При переході через меню за посиланням «Рекомендовані місця» ми потрапляємо до екрану з переліком Рекомендованих підтверджених місць (Рисунок 4.2), а при переході на одне з запропонованих місць можна потрапити до опису локації, її історії, як дістатися, геолокації та фотографії (Рисунок 4.3, 4.4). Оновлюваний список реалізовано через listView, а оновлення контенту в макеті через передачу позиції на яку було натиснуто.

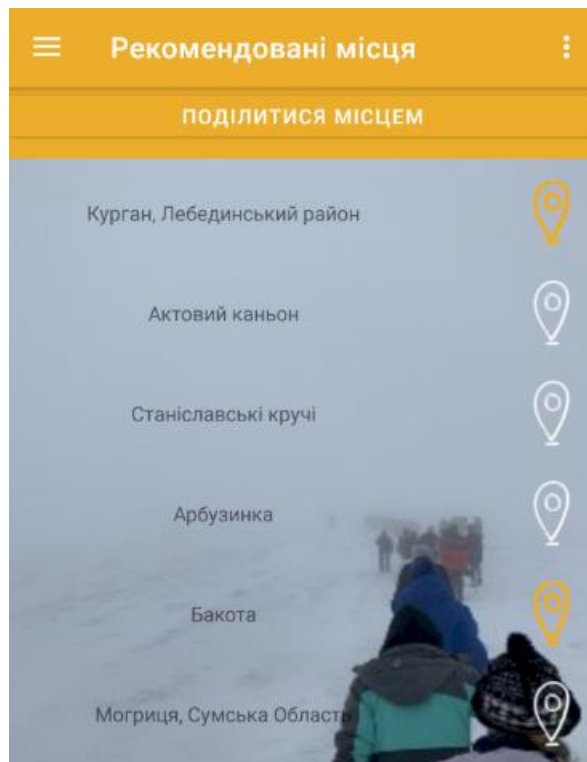


Рисунок 4.2. Список рекомендованих місць

Зауважте, що на рисунку 4.2 коло елементів RecyclerView списку «Курган, Лебединський район» та «Бакота» стоять вирізненні жовні значки – це ImageButton, які в свою чергу символізують відмічений уподобанням елемент, що переноситься до списку «Вподобане» (Рисунок 4.3)

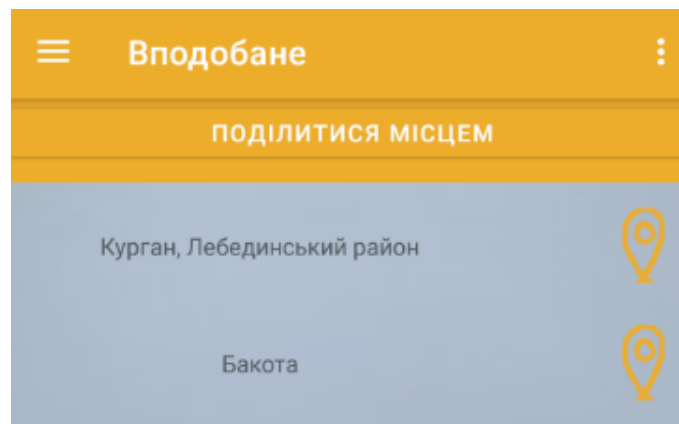


Рисунок 4.3. Список "Вподобане"

Варто зазначити, що при знятті прапорця вподобання елемент видаляється зі списку «Уподобане» і вимикає прапорець в інших списках.

село Курган



Село Курган знаходиться на правому березі річки Псел, вище за течією на відстані 3 км розташоване село Михайлівка Річка в цьому місці звивиста, утворює лимани, стариці і заболочені озера. Назва Село Курган до 1949 року називалося Курган – Озак. Така назва походить, як свідчать перекази, від куполоподібної гори Курган

Рисунок 4.4. Заповнення макету даними про обрану точку

На рис. 4.4 та 4.5 продемонстровано додатковий макет, що заповнюється контентом (зображення, заголовок, опис і фрагмент карти з маркером) відповідної точки.

На рисунку 8 продемонстрована робота маркера, який має свої атрибути, це колір, координати на карті та підпис; також видно роботу камери карти яка збільшує карту в 20 разів, що допомагає бачити на карті не усі 6 континентів, а максимально наближений фрагмент в центрі якого наш маркер.

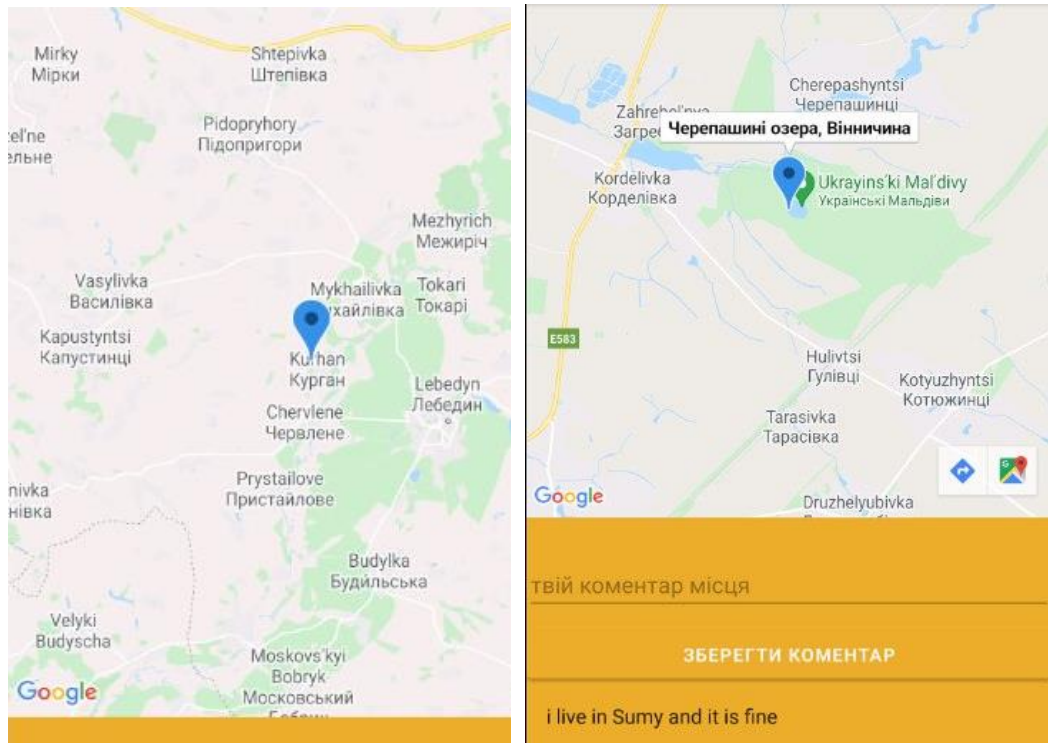


Рисунок 4.5. Заповнення макету даними про обрану точку

Екран реєстрації (рисунок 4.7) та екран реєстрації точки (рисунок 4.8) викликається з головного екрану після натиснення на відповідні кнопки (Рисунок 4.6) на якому в даному випадку продемонстрований макет заповнений списком регіонів.

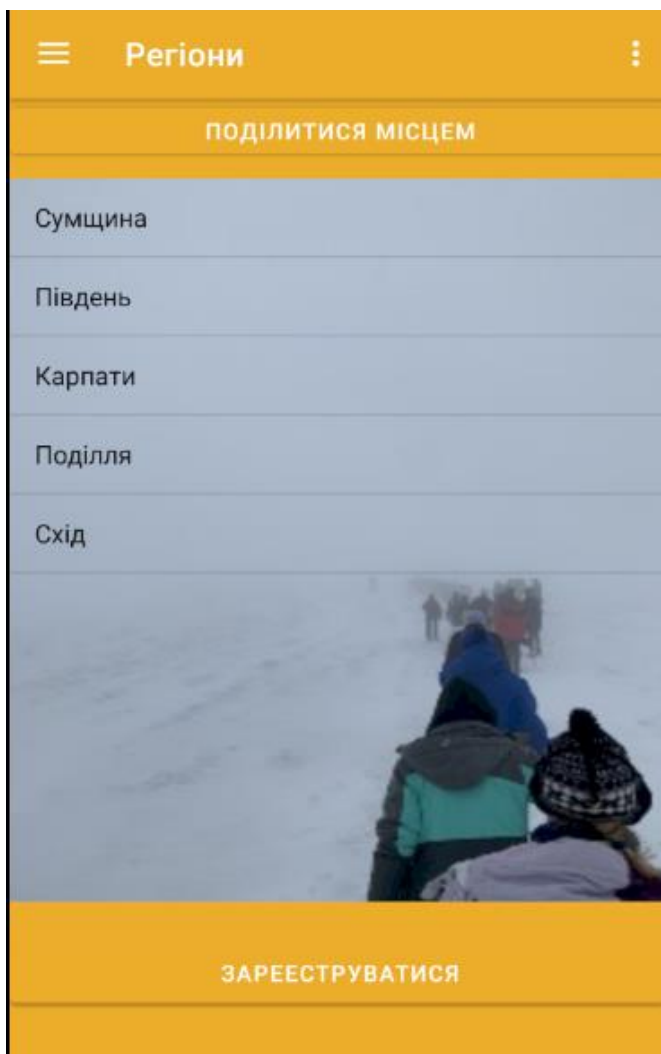
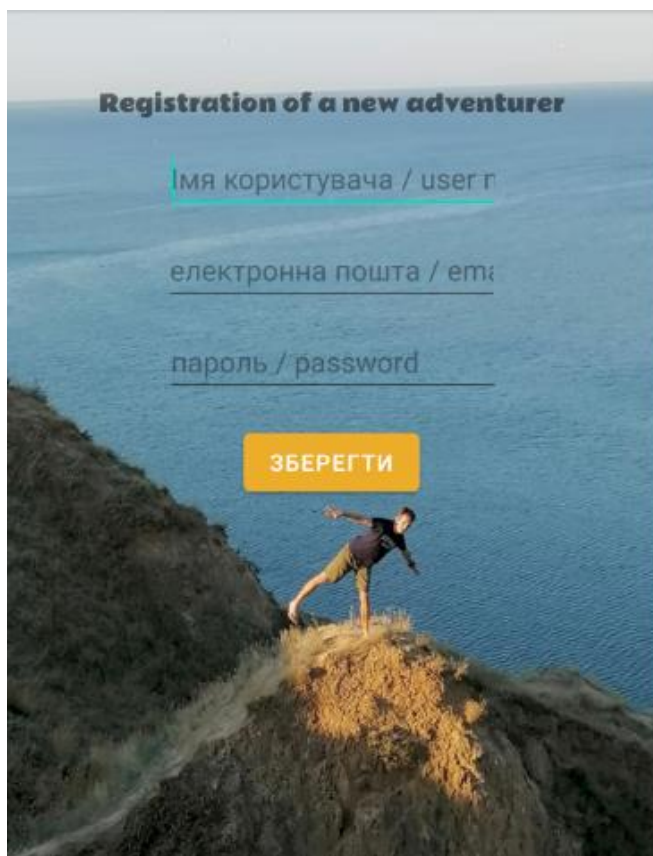


Рисунок 4.6. Макет заповнений списком регіонів



Registration of a new adventurer

Імя користувача / user name

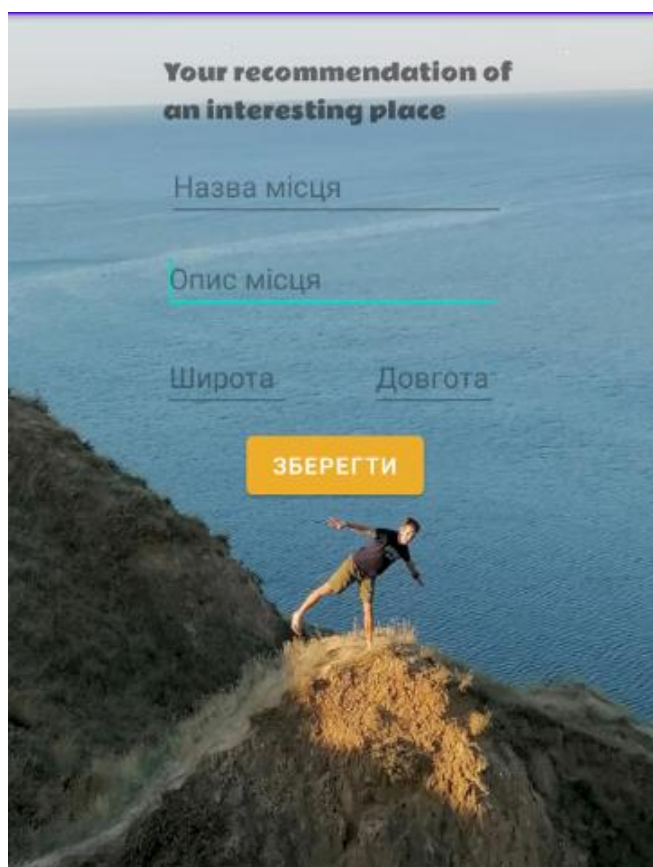
електронна пошта / email

пароль / password

ЗБЕРЕГТИ

The form is overlaid on a background image of a person standing on a cliff edge overlooking the ocean. The text is in a light blue color, and the 'ЗБЕРЕГТИ' button is orange.

Рисунок 4.7. Форма реєстрації користувача



Your recommendation of an interesting place

Назва місця

Опис місця

Широта Довгота

ЗБЕРЕГТИ

The form is overlaid on the same background image as Figure 4.7. The text is in a light blue color, and the 'ЗБЕРЕГТИ' button is orange.

Рисунок 4.8. Форма реєстрації користувацького місця

На екрані «Місця користувачів» демонструються усі користувацькі точки (рисунок 4.9)

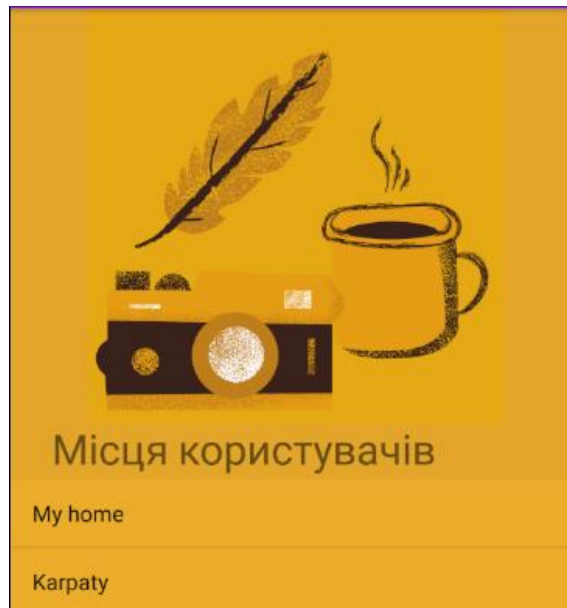


Рисунок 4.9. Екран "Місця Користувачів"

Натиснувши в меню навігації розділ «Про нас» можна перейти на екран з інформацією (рисунок 4.10)



Рисунок 4.10. Екран розділу "Про нас"

4.2 База даних Firebase

Відкриття профілю розробника в сервісі Firebase з попередньою авторизацією, ми можемо побачити ієрархічну систему записів, що приходять від користувача і записується за ключем. Серед ключів у нас є US_POINT, USER, COMENT_KEY, що представляють сутність (рисунок 4.11).

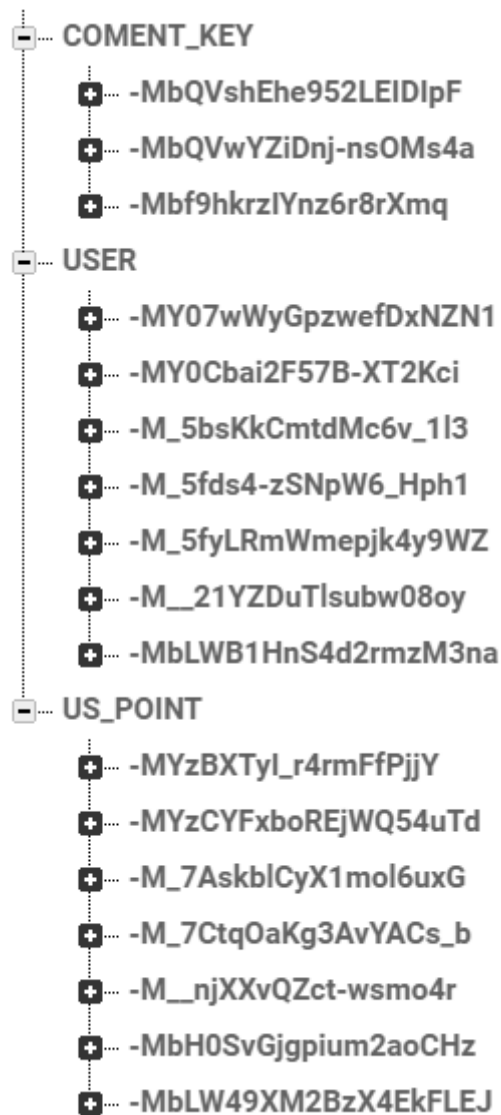


Рисунок 4.11. Представлення бази даних Firebase

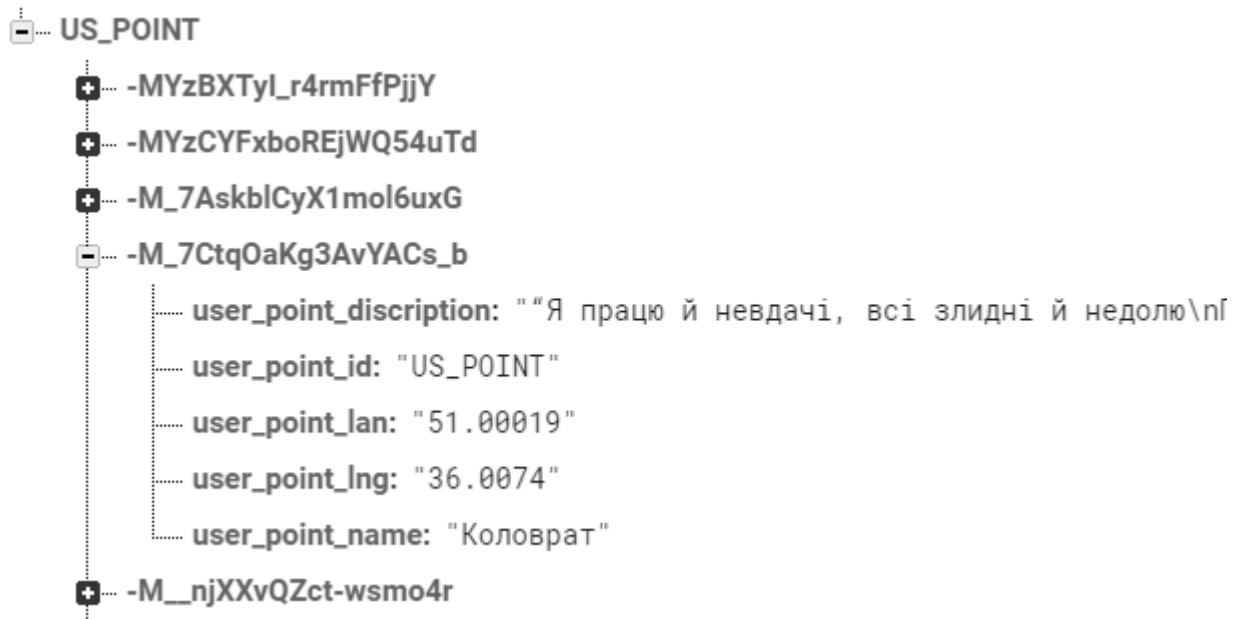


Рисунок 4.12. Запис опису точки

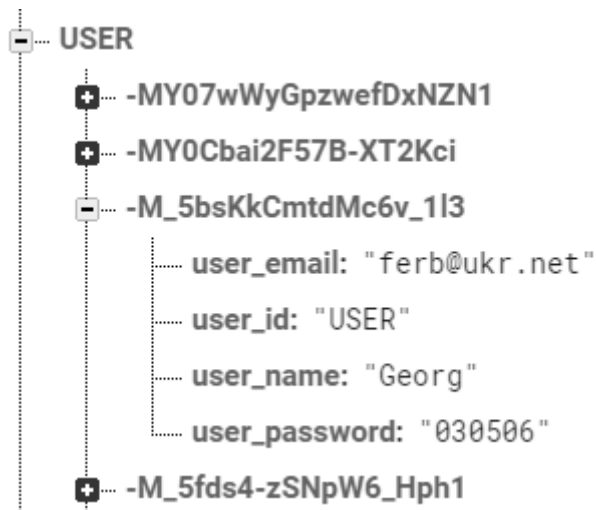


Рисунок 4.13. Запис користувача



Рисунок 4.14. Запис коментаря до точки

ВИСНОВОК

Під час виконання кваліфікаційної бакалаврської роботи розроблено туристичний додаток, що представляє собою систему для збереження, перегляду та поширення туристичних точок між користувачами.

Застосунок створено за допомогою об'єктно-орієнтованої мови програмування Java з використанням можливостей бібліотек утиліт MapsSDK і має мінімальний функціонал, а саме:

- мати доступ до найпопулярніших рекомендованих точок в режимі офлайн;
- дивитися опис та основну інформацію про точки запропоновані іншими користувачами з реальному часі;
- ділитися своїми рекомендаціями з іншими користувачами в реальному часі;
- залишати користувацькі коментарі то точок і переглядати коментарі інших користувачів;
- додавати точки до вподобань і мати їх перелік.

Розроблено користувацький інтерфейс, що позитивно впливає на зручність користування та візуальну складову. Додаток працює з даними, їх передача, збереження та отримання на девайс користувача відбувається через підключення до бази даних Firebase Database, що дозволяє мати одночасний доступ у реальному часі усім користувачам.

У подальшому додаток може бути розширено додаванням додаткових можливостей, як то завантаження великих об'ємів медіа, чатів, аудіогідів, що збільшить функціональність додатку та використання його у комерційних цілях та амбасадорства України закордоном.

СПИСОК ЛІТЕРАТУРИ

- [1] С. О Полковниченко, “ПРИВАБЛИВІСТЬ УКРАЇНИ НА ЄВРОПЕЙСЬКОМУ РИНКУ ТУРИСТИЧНИХ ПОСЛУГ,” *Ефективна економіка*, р. 8, 2020, doi: 10.32702/2307-2105-2020.1.78.
- [2] W. Tourism Organization, “UNWTO World Tourism Barometer and Statistical Annex, December 2020,” *UNWTO World Tour. Barom.*, vol. 18, no. 7, pp. 1–36, Dec. 2020, doi: 10.18111/wtobarometereng.2020.18.1.7.
- [3] “Майже 23 млн українців регулярно користуються Інтернетом – дослідження” [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://mind.ua/news/20204323-majzhe-23-mln-ukrayinciv-regulyarno-koristuyutsya-internetom-doslidzhennya>.
- [4] “Краєзнавство як дійовий засіб патріотичного виховання молоді. Oles Honchar Kherson regional universal scientific library.” [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://lib.kherson.ua/en-kr_rob-2.htm.
- [5] F. C. Morales, “Data reflecting the importance of tourism in the world | | Global Tourism Forum,” 2020. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.globaltourismforum.org/blog/2020/07/13/data-reflecting-the-importance-of-tourism-in-the-world/>.
- [6] “UN Live United Nations Web TV - António Guterres (secretario general) sobre la presentación del Documento de Políticas sobre el Turismo y la Covid-19,” 2020. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <http://webtv.un.org/watch/ant%C3%B3nio-guterres-secretario-general-sobre-la-presentaci%C3%B3n-del-documento-de-pol%C3%ADticas-sobre-el-turismo-y-la-covid-19/6184151592001/?term=csw&lan=russian>.
- [7] “Число туристов на курортах Одесской, Николаевской и Херсонской

- областей выросло на 30% - 'Vodafone Украина.'" [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://interfax.com.ua/news/telecom/673327.html>.
- [8] "ТОП-60 найцікавіших куточків України | Andy Travel Club. Подорожі на краю світу." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу: <https://andy-travel.com.ua/top-60-naucikavishyh-kutochkiv-ukrayiny>.
- [9] "Google Maps API + SDK - Overview for developers." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://ubilabs.com/en/google-maps/api-sdk-for-developers>.
- [10] "Maps SDK for Android Utility Library | Google Developers." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://developers.google.com/maps/documentation/android-sdk/utility>.
- [11] "What is Java and why is it important? - Code Institute." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://codeinstitute.net/blog/what-is-java/>.
- [12] "What Is WORA in Java? - Powered by Woz U." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://woz-u.com/blog/what-is-wora-in-java/>.
- [13] "Рейтинг мов програмування 2021: частка Python зменшується, а TypeScript обійшов C++ | DOU." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://dou.ua/lenta/articles/language-rating-jan-2021/>.
- [14] "Java vs Kotlin для Android: мнения разработчиков / Хабр." [Электронный ресурс] : [Веб-сайт]. – Электронні дані. – Режим доступу:<https://habr.com/ru/post/461877/>.
- [15] "Meet Android Studio | Android Developers." [Электронный ресурс] :

- [Веб-сайт]. – Електронні дані. – Режим доступу:<https://developer.android.com/studio/intro>.
- [16] “Realtime Database | React Native Firebase.” [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:<https://rnfirebase.io/database/usage>.
- [17] “Діаграми UML для моделювання процесів і архітектури проекту.” [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:<https://evergreens.com.ua/ua/articles/uml-diagrams.html>.
- [18] “UML Use Case Diagram Tutorial | Lucidchart.” [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:<https://www.lucidchart.com/pages/uml-use-case-diagram>.
- [19] “#11 - Функции, а также методы.” <https://itproger.com/course/java/11>.
- [20] “Java и Android | Передача данных между Activity. Сериализация.” [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:<https://metanit.com/java/android/2.12.php>.

ДОДАТОК. ЛІСТИНГ КЛАСІВ МОБІЛЬНОГО ДОДАТКУ

com.example.vacab_ivanturist.MainActivity

```

com.example.vacab_ivanturist

public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener{

    private DrawerLayout drawer;

    private ListView List;

    private String [] My_Array;

    private ArrayAdapter<String> Adapter;

    private Toolbar toolbar;

    private int category_index=0;

    private double [] array_lat_point = {Constant.LAT_point_1,
Constant.LAT_point_2, Constant.LAT_point_3};

    private double [] array_lng_point = {Constant.LNG_point_1,
Constant.LNG_point_2, Constant.LNG_point_3};

    private RecOnClickListener recOnClickListener;

    private DataAdapter adapterF;

    private List<ListItem> ListFData; //цілий тип-пасив, що складається
з стрінга і булеан

    private RecyclerView rcView;

    private String checkCateg_forFavor = "";

    private SharedPreferences pref;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

```



```

List=findViewById(R.id.listView_user2);

My_Array = getResources().getStringArray(R.array.about_app);

Adapter=new
ArrayAdapter<>(this,android.R.layout.simple_list_item_1,new
ArrayList<String>(Arrays.asList(My_Array)));

List.setAdapter(Adapter);

toolbar = findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

toolbar.setTitle(R.string.app_name);

drawer = findViewById(R.id.drawer_layout);

NavigationView navigationView = findViewById(R.id.nav_view);

navigationView.setNavigationItemSelectedListener(this);

ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(

        this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);

drawer.addDrawerListener(toggle);

toggle.syncState();

List.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

        @Override

        public void onItemClick(AdapterView<?> parent, View view,
int position, long id)

        {

                Intent intent = new
Intent(MainActivity.this,Text_Content_Activity.class);

                intent.putExtra("category", category_index);

                intent.putExtra("position", position);

                intent.putExtra("lat", array_lat_point[position]);

                intent.putExtra("lng", array_lng_point[position]);

```

```

        startActivity(intent);
    }

});

setOnClickListener();

init();
}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action bar if it is
present.

    getMenuInflater().inflate(R.menu.main, menu);

    return true;
}

//реакції на маніпуляції з меню

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    int id = item.getItemId();

    if (id ==R.id.like_point)
    {

        updateFavororite();

        toolbar.setTitle("Вподобане");

        category_index=1;
    }

    if (id ==R.id.nav_all_points)
    {

```

```

updateMineList (getResources () .getStringArray (R.array.all_point_array),
"all_point_array");

        My_Array =
getResources () .getStringArray (R.array.all_point_array);

        Adapter.clear ();

        Adapter.addAll (My_Array);

        Adapter.notifyDataSetChanged ();

        toolbar.setTitle (R.string.menu_points);

        category_index=1;

    }

    else if (id==R.id.nav_regions)

    {
updateMineList (getResources () .getStringArray (R.array.regions_array),
"regions_array");

        //My_Array =
getResources () .getStringArray (R.array.regions_array);

        //Adapter.clear ();

        //Adapter.addAll (My_Array);

        //Adapter.notifyDataSetChanged ();

        toolbar.setTitle (R.string.menu_regions);

        Toast.makeText (this, "Про регіони нічого поки не має",
Toast.LENGTH_SHORT).show ();

        category_index=2;

    }

    else if (id==R.id.nav_other_points)

    {

//updateMineList (getResources () .getStringArray (R.array.regions_array),
"regions_array");

```

```

        Intent intent = new
Intent (MainActivity.this,Watch_users_point_Activity.class);

        startActivity(intent);

        toolbar.setTitle (R.string.menu_other_points);

        category_index=3;

    }

    else if (id==R.id.nav_for_us)

    {

        Intent intent = new
Intent (MainActivity.this,For_us_Activity.class);

        startActivity(intent);

    }

    drawer.closeDrawer (GravityCompat.START);

    returntrue;

}

public void onCl_share_point (View view)

{

    Intent intent = new
Intent (MainActivity.this,Shere_Point_Activity.class);

    startActivity(intent);

}

public void onCl_registretion (View view)

{

    Intent intent = new
Intent (MainActivity.this,Registration_Activity.class);

    startActivity(intent);

}

private void updeteFavororite()

```

```

    {

        List<ListItem> listFavor = new ArrayList<>();

        List <String[]> listData = new ArrayList<>();

listData.add(getResources().getStringArray(R.array.all_point_array));
listData.add(getResources().getStringArray(R.array.regions_array));

        String [] cat_array = {"all_point_array", "regions_array"};

        for(int i= 0; i<listData.size(); i++)

        {

            for(int j = 0; j < listData.get(i).length; j++)

            {

                String d = pref.getString(cat_array[i], "none");

                if(d!= null) if(d.charAt(j)=='1')

                {

                    ListItem item = new ListItem();

                    item.setText(listData.get(i)[j]);

                    item.setPosition(j);

                    item.setCategory(cat_array[i]);

                    listFavor.add(item);

                }

            }

        }

        adapterF.updateList(listFavor, true);

    }

private void init()

{

    pref = getSharedPreferences("CAT", MODE_PRIVATE);

```

```

rcView = findViewById(R.id.recycleView_on_layout);

rcView.setLayoutManager(new LinearLayoutManager(this));

ListFDData = new ArrayList<>();

String[] rec_points_array =
getResources().getStringArray(R.array.all_point_array);

adapterF = new DataAdapter(this, recOnClickListener, ListFDData);

updateMineList(rec_points_array, "all_point");

rcView.setAdapter(adapterF);
}

private void updateMineList(String[] array, String category)
{
    checkCateg_forFavor = "all_point_array";

    checkCateg_forFavor = category; //категорія в якій працюємо

    StringBuilder stringBuilderCategor;

    stringBuilderCategor = new StringBuilder();

    String tempCat = pref.getString(category, "none");

    if(tempCat != null){

        if(tempCat.equals("none"))

        {

            for (int i = 0; i < array.length; i++)

            {

                stringBuilderCategor.append("0");

            }

            Log.d("MyLogCategory", category + " " +
stringBuilderCategor.toString());

            saveString(stringBuilderCategor.toString());

        }
}

```

```

        else{}

    }

    //для заповнення усіх блоків новим вмістом
    List<ListItem> list = new ArrayList<>();

    for(int i = 0; i< array.length; i++)

    {

        ListItem item = new ListItem();

        item.setText(array[i]);

        item.setCategory(category);

        item.setPosition(i);

        list.add(item);

    }

    adapterF.updateList(list, false);

}

//для додавання до вподобань
private void setRecOnClickListener()

{

    recOnClickListener = new RecOnClickListener() {

        @Override

        public void onItemClick(int position)

        {

            String tempCat = pref.getString(checkCateg_forFavor,

"none");

            Log.d("MyLOG", "Saved data favorite: " +

pref.getString(checkCateg_forFavor, "none"));

            if(tempCat != null)

            {

```

```

        if(tempCat.charAt(position) == '0')
            { saveString(replaceCharAtPosition(position, '1',
tempCat)); }

        if(tempCat.charAt(position) == '1')
            {saveString(replaceCharAtPosition(position, '0',
tempCat));}

            }

        }

    } ;

}

{

private String replaceCharAtPosition(int position, char ch, String
st)

{

    char [] charArray = st.toCharArray(); //розкладання слова на
СИМВОЛИ

    charArray[position] = ch;

    return new String(charArray);

}

private void saveString(String stToSave)

{

    SharedPreferences.Editor editor = pref.edit();

    editor.putString(checkCateg_forFavor, stToSave);

    editor.apply();

    Log.d("MyLOG", "Saved data favorite: " +
pref.getString(checkCateg_forFavor, "none"));

}

}

```


com.example.vacab_ivanturist.Text_Content_Activity

```
package com.example.vacab_ivanturist;

import android.content.Intent;

import android.graphics.Typeface;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.AdapterView;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

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.BitmapDescriptorFactory;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.firebase.database.DataSnapshot;
```

```

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

import java.util.List;

public class Text_Content_Activity extends AppCompatActivity implements
OnMapReadyCallback

{

    private TextView text_Content_point; //увесь опис

    private Typeface my_front_1; //тип шрифту

    private TextView point_name; //назва точки

    private ImageView imagePoint;

    private GoogleMap map;

    private LatLng locat_point;

    private int position =0;

    private int category =0;

    private int [] array_point = {R.string.point_1, R.string.point_2,
R.string.point_3,

        R.string.point_4, R.string.point_5, R.string.point_6,
R.string.point_7, R.string.point_8, R.string.point_9, R.string.point_10};

    private int [] array_image_point = {R.drawable.kurgan_2,
R.drawable.dolina, R.drawable.kruchi,

        R.drawable.point_4, R.drawable.point_5, R.drawable.point_6,
R.drawable.point7, R.drawable.point8, R.drawable.point9, R.drawable.point10};

    private int [] array_name_point = {R.string.name_point_1,
R.string.name_point_2, R.string.name_point_3,

        R.string.name_point_4, R.string.name_point_5,
R.string.name_point_6, R.string.name_point_7, R.string.name_point_8,
R.string.name_point_9, R.string.name_point_10};

```

```

        private int text_for_us = R.string.author_team_text; //потім заміню
на текст про команду

        private int author_team_img = R.drawable.author_team;

        private int user_point_back = R.drawable.user_point_back;

        private double LAT;

        private double LNG;

        private String title_point;

        private DatabaseReference myDataBase;

        private String USER_KEY = "US_POINT";

        private String COMENT_KEY = "COMENT_KEY";

        private EditText coment_text;

        private ListView listViewComent;

        private ArrayAdapter<String> adapterComent;

        private List<String> listDataComent;

        private List<UserPoint> listUserComent;

        @Override

        protected void onCreate(@Nullable Bundle savedInstanceState) {

            super.onCreate(savedInstanceState);

            setContentView(R.layout.text_content);

            //locat_point = new LatLng(LAT, LNG);

            point_name=findViewById(R.id.point_name);

            init();

            receiveIntent();

            SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map_fr_2);

            mapFragment.getMapAsync(this);

            //getDataFromDB();

```

```

    }

    private void receiveIntent()
    {
        Intent i = getIntent();

        if (i != null)
        {
            category = i.getIntExtra("category", 0);
            position = i.getIntExtra("position", 0);
        }

        switch (category)
        {
            case (1):
                text_Content_point.setText(array_point[position]);

                imagePoint.setImageResource(array_image_point[position]);

                point_name.setText(array_name_point[position]);

                LAT=i.getDoubleExtra("lat", 0);
                LNG=i.getDoubleExtra("lng", 0);

                title_point = getString(array_name_point[position]);

                break;

            case (2):
                Toast.makeText(this, "Про регіони нічого поки не має",
                Toast.LENGTH_SHORT).show();

                break;

            case (3):
                point_name.setText(i.getStringExtra("point_name"));

                text_Content_point.setText(i.getStringExtra("point_descriotion"));

```

```

        imagePoint.setImageResource(user_point_back);

        LAT = i.getDoubleExtra("lat", 45);

        LNG = i.getDoubleExtra("lng", 45);

        title_point = i.getStringExtra("point_name");

        break;

    case (4):

        text_Content_point.setText(text_for_us);

        imagePoint.setImageResource(author_team_img);

        point_name.setText("Інформація про нашу команду");

        break;

    }

}

private void init()

{

    point_name = findViewById(R.id.point_name); //назва точки

    text_Content_point = findViewById(R.id.text_main_content);
//опис точки

    imagePoint = findViewById(R.id.image_point_id); //фото

    my_front_1 =
Typeface.createFromAsset(this.getAssets(),"fonts/VollkornSC-Bold.ttf");

    text_Content_point.setTypeface(my_front_1);

    coment_text = findViewById(R.id.editTextTextPersonName);
//считування коментаря

    myDataBase =
FirebaseDatabase.getInstance().getReference(COMENT_KEY);

    //ініціалізація списку коментарів

    listViewComent = findViewById(R.id.coments_list); //місце куди
записувати

```

```

        listDataComent = new ArrayList<>(); //створення масиву

        listUserComent= new ArrayList<>(); //створення масиву

        adapterComent = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, listDataComent);

        listViewComent.setAdapter(adapterComent);

        getDataFromDB();

    }

    @Override

    public void onMapReady(GoogleMap googleMap) {

        map = googleMap;

        locat_point = new LatLng(LAT, LNG);

        map.addMarker(

            new MarkerOptions()

                .position(locat_point)

                .title(title_point)

                .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE
)))

        CameraUpdate cameraUpdate =
CameraUpdateFactory.newLatLngZoom(locat_point,12);

        googleMap.moveCamera(cameraUpdate);

    }

    public void onClick_saveComents(View view)

    {

        Toast.makeText(this, "position " + position + " category " +
category, Toast.LENGTH_SHORT).show();

        String coment_id= myDataBase.getKey();

        String coment_text_BD = coment_text.getText().toString();

```

```

        UserComent newcoment = new UserComent(coment_id, coment_text_BD,
String.valueOf(category) , String.valueOf(position)); //передае в конструктор
нового користувача

        if (!TextUtils.isEmpty(coment_text_BD))

            {

                myDataBase.push().setValue(newcoment);

                Toast.makeText(this, "Коментар записано, дякуємо",
Toast.LENGTH_SHORT).show();

            }

            else{Toast.makeText(this, "Поле пугте",
Toast.LENGTH_SHORT).show();}

        }

        private void getDataFromDB(){

            ValueEventListener vlistener = new ValueEventListener() {

                @Override

                public void onDataChange(@NonNull DataSnapshot dataSnapshot)

                {

                    if(listDataComent.size()>0) listDataComent.clear();

                    if(listUserComent.size()>0) listUserComent.clear();

                    for(DataSnapshot ds : dataSnapshot.getChildren())

                    {

                        UserComent coment = ds.getValue(UserComent.class);

                        assert coment != null;

                        //if(coment.co)

                        listDataComent.add(coment.coment_text);

                        //listUserComent.add(coment_text);

                    }

                    adapterComent.notifyDataSetChanged();

```

```

        }

        @Override

        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }

    };

    myDataBase.addValueEventListener(vListener);

}

}

```

com.example.vacab_ivanturist.User

```

package com.example.vacab_ivanturist;

public class User {

    public String user_id, user_name, user_email, user_password;

    public User() {

    }

    public User(String user_id, String user_name, String user_email,
String user_password) {

        this.user_id = user_id;

        this.user_name = user_name;

        this.user_email = user_email;

        this.user_password = user_password;

    }

}

```

com.example.vacab_ivanturist.UserPoint

```

public class UserPoint {

    public String user_point_id, user_point_name,
user_point_discription, user_point_lan, user_point_lng;

```



```

public UserPoint() {

}

public UserPoint(String user_point_id, String user_point_name,
String user_point_discription, String user_point_lan, String user_point_lng)
{

    this.user_point_id = user_point_id;

    this.user_point_name = user_point_name;

    this.user_point_discription = user_point_discription;

    this.user_point_lan = user_point_lan;

    this.user_point_lng= user_point_lng;

}

}

```

com.example.vacab_ivanturist.adapter_favor.DataAdapter

```

package com.example.vacab_ivanturist.adapter_favor;

public class DataAdapter extends RecyclerView.Adapter
<DataAdapter.DataHolder>

{

    private Context context;

    private RecOnClickListener recOnClickListener;

    private List <ListItem> listItemArrey;

    private SharedPreferences preferences;

    private boolean isFavorBlock;

    public DataAdapter(Context context, RecOnClickListener
recOnClickListener, List<ListItem> listItemArrey)

    {

        this.context = context;

        this.recOnClickListener = recOnClickListener;

```

```

        this.listItemArrey = listItemArrey;

        preferences = context.getSharedPreferences("CAT",
Context.MODE_PRIVATE);

    }

    public DataHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType)

    {

        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_layout ,
parent, false);

        return new DataHolder(view);

    }

    public void onBindViewHolder(@NonNull DataHolder holder, int
position)

    {

        holder.setData(listItemArrey.get(position));

    }

    public int getItemCount()

    {

        return listItemArrey.size();

    }

    public class DataHolder extends RecyclerView.ViewHolder implements
View.OnClickListener

    {

        private boolean isFavCheck = false;

        private TextView tvText;

        private ImageButton imButtonfavor;

        public DataHolder(@NonNull View itemView)

```

```

    {
        super(itemView);

        tvText = itemView.findViewById(R.id.tvText); //блок з
вподобанням

        imButtonfavor = itemView.findViewById(R.id.imButton); //мій
прапорець

        imButtonfavor.setOnClickListener(this);
    }

    public void setData(ListItem item)
    {
        tvText.setText(item.getText());

        if(!isFavorBlock)
        {
            setFav(item, getAdapterPosition()); //дивимося яка
позиція у елементу з вподобайкою

        }

        else {setFavAll();}

    }

    @Override

    public void onClick(View v)
    {
        isFavCheck = !isFavCheck;

        if (isFavCheck)
        {
            imButtonfavor.setImageResource(R.drawable.marker_yellow);
        }

        else
        {

```

```

imButtonfavor.setImageResource(R.drawable.marker_white);

        }

        if(!isFavorBlock) {

            recOnClickListener.onItemClicked(getAdapterPosition());

        }

        else

        {

            deleteItem();

        }

    }

    private String replaceCharAtPosition(int position, char ch,
String st)

    {

        char [] charArray = st.toCharArray(); //розкладання
слова на символи

        charArray[position] = ch;

        return new String(charArray);

    }

    private void saveString(String stToSave)

    {

        ListItem item = listItemArrey.get(getAdapterPosition());

        SharedPreferences.Editor editor = preferences.edit();

        editor.putString(item.getCategory(), stToSave);

        editor.apply();

    }

    private void deleteItem() //знімаємо вподобання усюди і
видаляємо з Упадобаного, перезаписуємо списки вподобаних у всіх блоках.

```

```

        {

            ListItem item = listItemArrey.get(getAdapterPosition());
//або дізнатися на якому місці він був у рідному списку

            String dataForChange =
preferences.getString(item.getCategory(), "none");

            if (dataForChange == null) return;

            String replecedData =
replaceCharAtPosition(item.getPosition(), '0', dataForChange);

            saveString(replecedData);

            listItemArrey.remove(getAdapterPosition());

            notifyItemRemoved(getAdapterPosition());

            notifyItemRangeChanged(getAdapterPosition(),
listItemArrey.size());

        }

        private void setFavAll()

        {

            imButtonfavor.setImageResource(R.drawable.marker_yellow);

        }

        private void setFav(ListItem item, int position) //ставить
відмітки на вже вподобаних екземплярах

        {

            String fav_data = preferences.getString(item.getCategory(),
"none"); //отримуємо стрінг 01 який записаний у пам'яті

            //ставимо вподабайку або ні, в залежності від значень попередньої сесії

            if(fav_data != null)

            {

                char[] charArray = fav_data.toCharArray();

                switch (charArray[position])

                {

```

```

        case '0':

imButtonfavor.setImageResource(R.drawable.marker_white);

            isFavCheck = false;

            break;

        case '1':

imButtonfavor.setImageResource(R.drawable.marker_yellow);

            isFavCheck = true;

            break;

        }

    }

}

public void updateList(List<ListItem> listMyArray, boolean isFav)
{
    isFavorBlock = isFav;

    listItemArrey.clear();

    listItemArrey.addAll(listMyArray);

    notifyDataSetChanged();

}
}

```

com.example.vacab_ivanturist.adapter_favor.RecOnClickListe ner

```

package com.example.vacab_ivanturist.adapter_favor;

public interface RecOnClickListener

```

```

{
    public void onItemClick (int position);
}

```

com.example.vacab_ivanturist.For_us_Activity

```

public class For_us_Activity extends AppCompatActivity
{
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.for_us);
    }
}

```

com.example.vacab_ivanturist.Registration_Activity

```

public class Registration_Activity extends AppCompatActivity {

    private EditText regName, regEmail, regPas;

    private DatabaseReference myDataBase;

    private String USER_KEY = "USER";

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.registretion);

        init();

    }

    private void init()

    {

```

```

        regName=findViewById(R.id.ed_userName);

        regEmail=findViewById(R.id.ed_mail);

        regPas=findViewById(R.id.ed_Password);

        myDataBase =
        FirebaseDatabase.getInstance().getReference(USER_KEY);
    }

    public void onClick_save_us2(View view)
    {

        String id= myDataBase.getKey();

        String name = regName.getText().toString();

        String email = regEmail.getText().toString();

        String password = regPas.getText().toString();

        User newUser = new User(id, name, email, password);
        //передає в конструктор нового користувача

        if (!TextUtils.isEmpty(name) && !TextUtils.isEmpty(email) &&
        !TextUtils.isEmpty(password))
        {

            myDataBase.push().setValue(newUser);

            Toast.makeText(this, "Запит реєстрації надіслано",
            Toast.LENGTH_SHORT).show();

        }

        else{Toast.makeText(this, "Не всі поля заповненні",
        Toast.LENGTH_SHORT).show();}

    } }

```

com.example.vacab_ivanturist.Shere_Point_Activity

```

public class Shere_Point_Activity extends AppCompatActivity {

    private EditText edName_point, edDescription_point, lag_p, lng_p;

```



```

private DatabaseReference myDataBase;

private String USER_KEY = "US_POINT";

@Override

protected void onCreate(@Nullable Bundle savedInstanceState)

{

    super.onCreate(savedInstanceState);

    setContentView(R.layout.form_recomend_point);

    init();

}

private void init()

{

    edName_point=findViewById(R.id.Text_1_recPointName);

    edDescription_point=findViewById(R.id.editText_2_about);

    lag_p=findViewById(R.id.editText_3_lon);

    lng_p=findViewById(R.id.editText_3_lng);

    myDataBase =
FirebaseDatabase.getInstance().getReference(USER_KEY);

}

public void onClick_save(View view)

{

    String id= myDataBase.getKey();

    String name = edName_point.getText().toString();

    String description =
edDescription_point.getText().toString();

    String lan = lag_p.getText().toString();

    String lng = lng_p.getText().toString();

```

```

        UserPoint newUser = new UserPoint(id, name, description, lan,
lng); //передає в конструктор нового користувача

        if (!TextUtils.isEmpty(name) &&
!TextUtils.isEmpty(description) && !TextUtils.isEmpty(lan) &&
!TextUtils.isEmpty(lng))

        {

            myDataBase.push().setValue(newUser);

            Toast.makeText(this, "Точка зареєстрована, дякуємо",
Toast.LENGTH_SHORT).show();

        }

        else{Toast.makeText(this, "Не всі поля заповненні",
Toast.LENGTH_SHORT).show();}

    }

}

```

com.example.vacab_ivanturist.Watch_users_point_Activity

```

public class Watch_users_point_Activity extends AppCompatActivity {

    private ListView listViewUs;

    private ArrayAdapter<String> adapterUs;

    private List<String> listDataUs;

    private List<UserPoint> listUserPoint;

    private DatabaseReference myDataBase;

    private String USER_KEY = "US_POINT";

    private int category_index=3;

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.recomend_point);

        init();

        getDataFromDB();

    }

    private void init()

    {

        listViewUs = findViewById(R.id.id_list_RecomUserPoint);
//місце куди записувати

        listDataUs = new ArrayList<>(); //створення масиву

        listUserPoint = new ArrayList<>(); //створення масиву

        adapterUs = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, listDataUs);

        listViewUs.setAdapter(adapterUs);

        myDataBase =
FirebaseDatabase.getInstance().getReference(USER_KEY);

        setOnCkickItem();

    }

    private void getDataFromDB(){

        ValueEventListener vListener = new ValueEventListener() {

            @Override

            public void onDataChange(@NonNull dataSnapshot)

dataSnapshot)

            {

                if(listDataUs.size()>0) listDataUs.clear();

                if(listUserPoint.size()>0) listUserPoint.clear();

                for(DataSnapshot ds : dataSnapshot.getChildren())

                {

```

```

        UserPoint point_for_list =
ds.getValue(UserPoint.class);

        assert point_for_list != null;

        listDataUs.add(point_for_list.user_point_name);

        listUserPoint.add(point_for_list);

    }

    adapterUs.notifyDataSetChanged();

}

@Override

public void onCancelled(@NonNull DatabaseError
databaseError) {

}

};

myDataBase.addValueEventListener(vListener);

}

private void setOnCkickItem() {

    listViewUs.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

        @Override

        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

            UserPoint userPoint = listUserPoint.get(position);

            Intent intent = new
Intent(Watch_users_point_Activity.this, Text_Content_Activity.class);

            intent.putExtra("category", category_index);

            intent.putExtra("position", position);

            intent.putExtra("point_name",
userPoint.user_point_name);

```

```
        intent.putExtra("point_description",
userPoint.user_point_discription);

        intent.putExtra("lat",
Double.valueOf(userPoint.user_point_lan));

        intent.putExtra("lgn",
Double.valueOf(userPoint.user_point_lng));

        startActivity(intent);

    }

});

}

}
```