



SAPIENZA  
UNIVERSITÀ DI ROMA



# SuggesTrip

- Find the location that is most suited for you!



# The Team



**Valerio Longo**

1655653



**Luigi Sigillo**

1761017



**Giuliano Martinelli**

1915652

# Table of Contents



**1**

## **Introduction**

Idea  
Approach

**2**

## **Architecture**

Technologies  
Structure

**3**

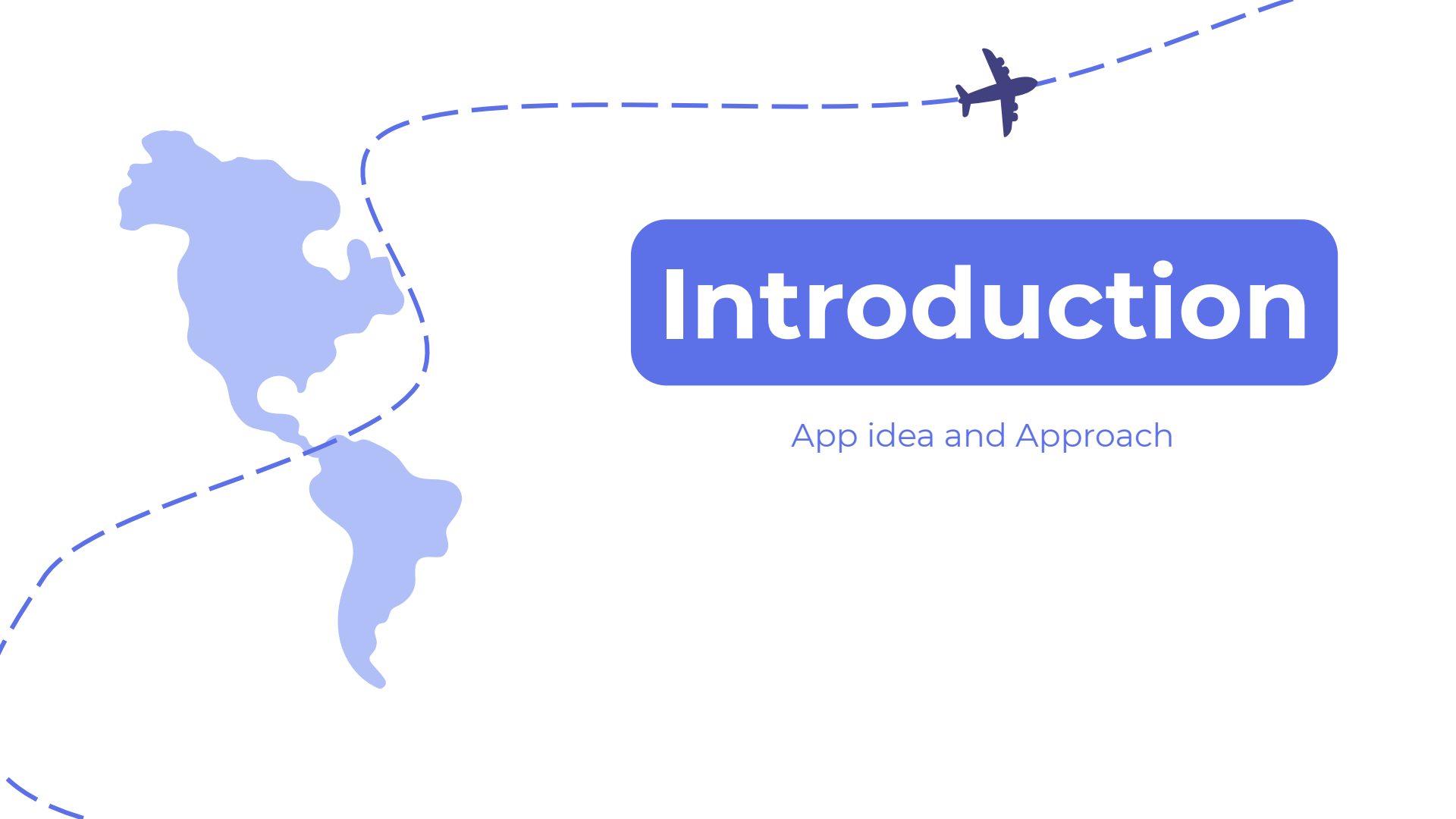
## **Functionalities**

Login, City Algorithm  
Favourites, CRUD

**4**

## **Conclusions**

Future work



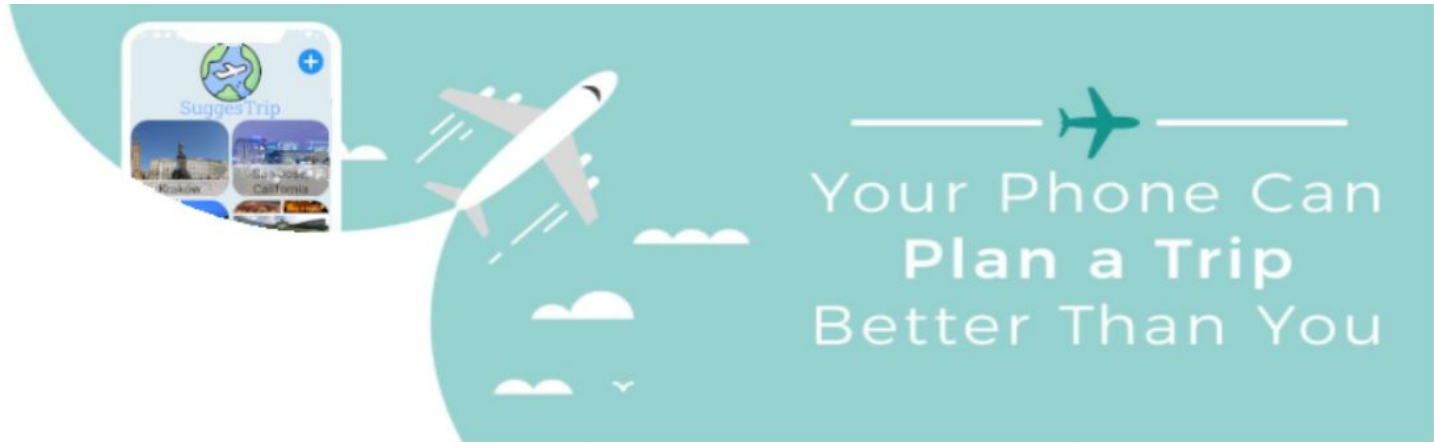
# Introduction

App idea and Approach

# Idea







- Our application idea is Suggestrip, a service to find the location that best suits your tastes and needs.



- This application allows you to report what your needs and preferences are when it comes to travel through a survey during the research phase, in order to find the best destinations for the user.

# Approach



- The destination search algorithm bases its operation on different matching criteria between the user's needs and what the different locations offer.
- It also takes into account, through the use of external APIs:
  -  Weather conditions
  -  Information about the current Coronavirus situation for the different locations, in order to provide updated information when a user is suggested a location.
- On the Android application side we have provided:
  - Responsive UI
  -  Use of the GPS sensor to calculate the distance between the various destinations
  -  Accelerometer to return to a random city, whenever a user shakes the phone.
- We have implemented the cloud structure using AWS and the authentication with Firebase.

# Architecture













Technologies  
Structure



# Technologies

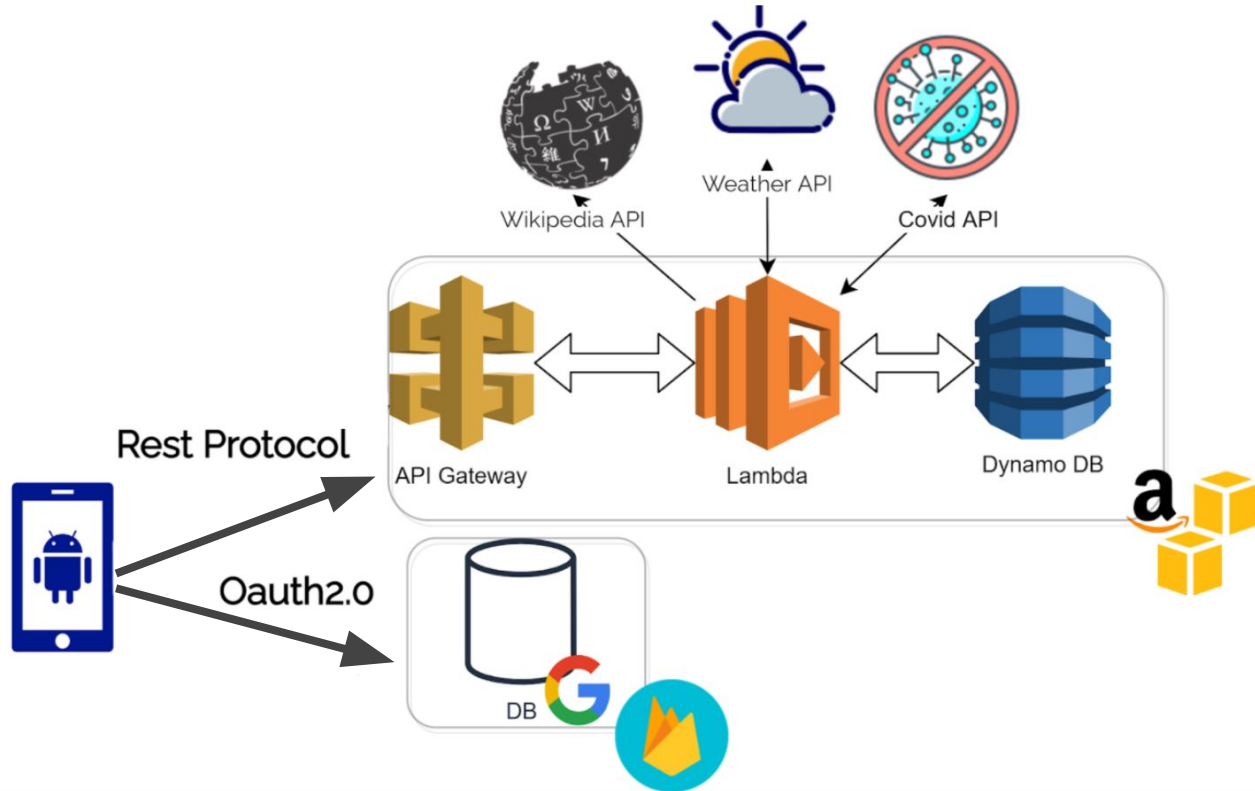


Web-API

Backend		Application	
 API Gateway	 Firebase	 android studio	 Kotlin
 Lambda	 Oauth	API	
 Dynamo DB	 Python	 OpenWeather	 Covid API
 AWS			 Wikipedia API



# Architecture





# Functionalities and Design

Navigation Path

Login

Login and Homepage

City

Algorithm

ExploreView and CityView

Favourites

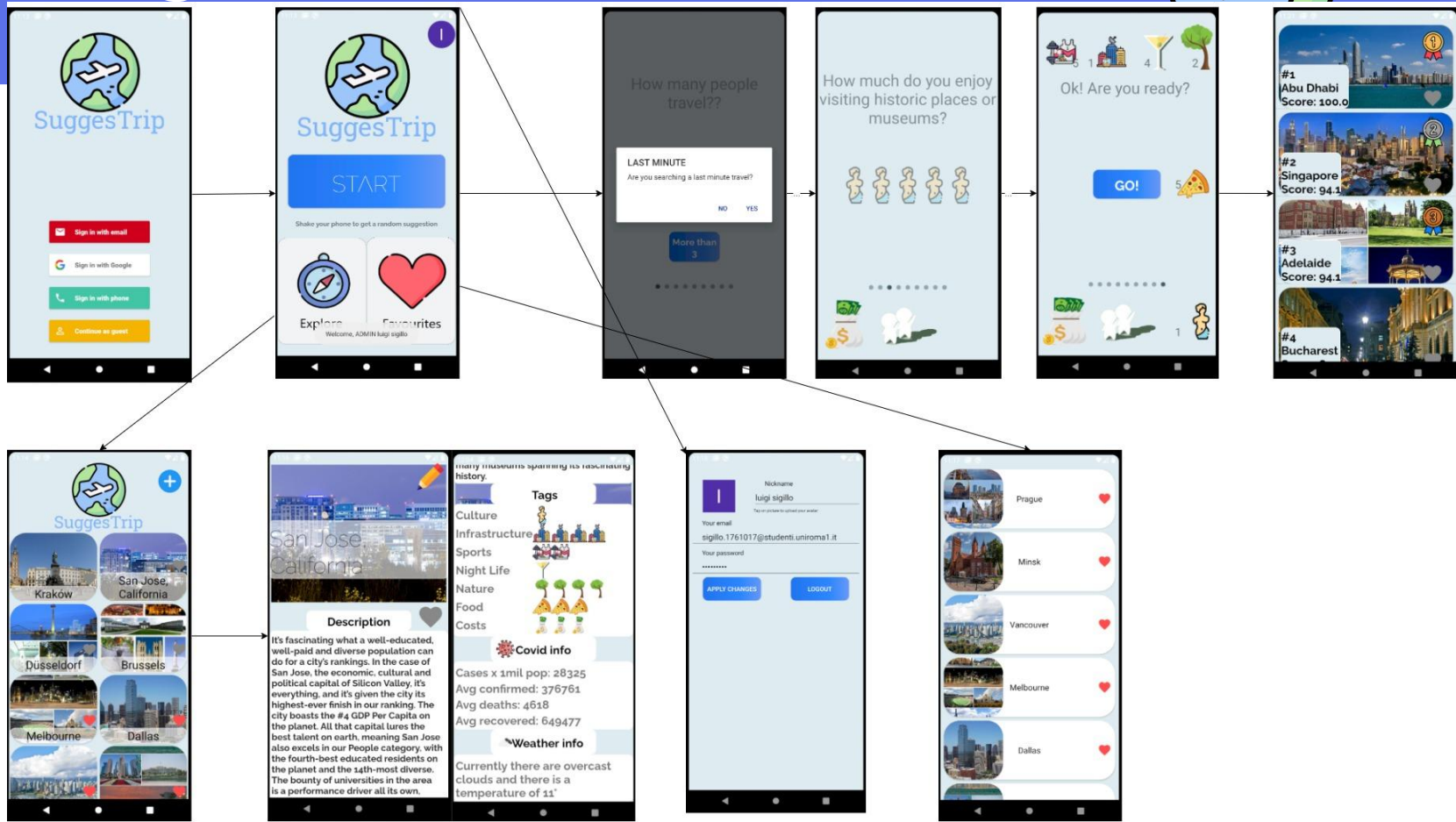
FavouritesView

CRUD

Admin views



# Navigation Path



# Login



AUTHENTICATION



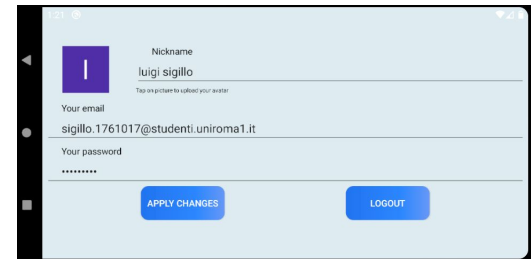
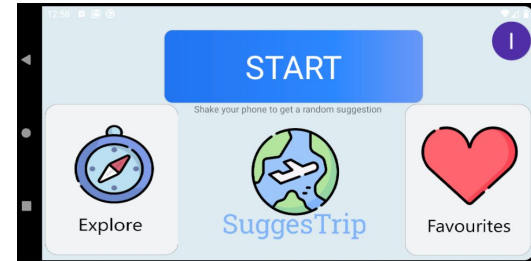
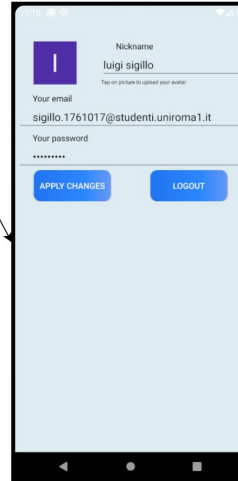
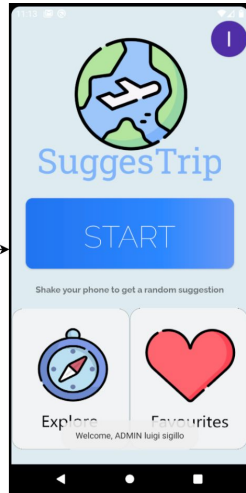
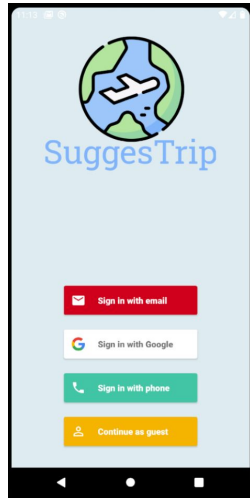
- Firebase Authentication
- We use Firebase AuthUi to allow users to log into our app using various providers:
  - Generic Email
  - Google account
  - Phone number
  - Anonymous (Guest)

The credentials will be passed to the Firebase Authentication SDK and after getting a response from their servers, the user's access information will be saved into **Firestore**.



```
private fun showSignInOptions() {  
    providers = Arrays.asList<AuthUI.IdpConfig>(  
        AuthUI.IdpConfig.EmailBuilder().build(),  
        AuthUI.IdpConfig.GoogleBuilder().build(),  
        AuthUI.IdpConfig.PhoneBuilder().build(),  
        AuthUI.IdpConfig.AnonymousBuilder().build()  
    )  
  
    startActivityForResult(AuthUI.getInstance()  
        .createSignInIntentBuilder()  
        .setAvailableProviders(providers)  
        .setIsSmartLockEnabled(false)  
        .setTheme(R.style.AppTheme)  
        .setLogo(R.drawable.logo)  
        .build(), OK_RESPONSE_CODE)  
}
```

# Login and HomePage



- Oauth











# City



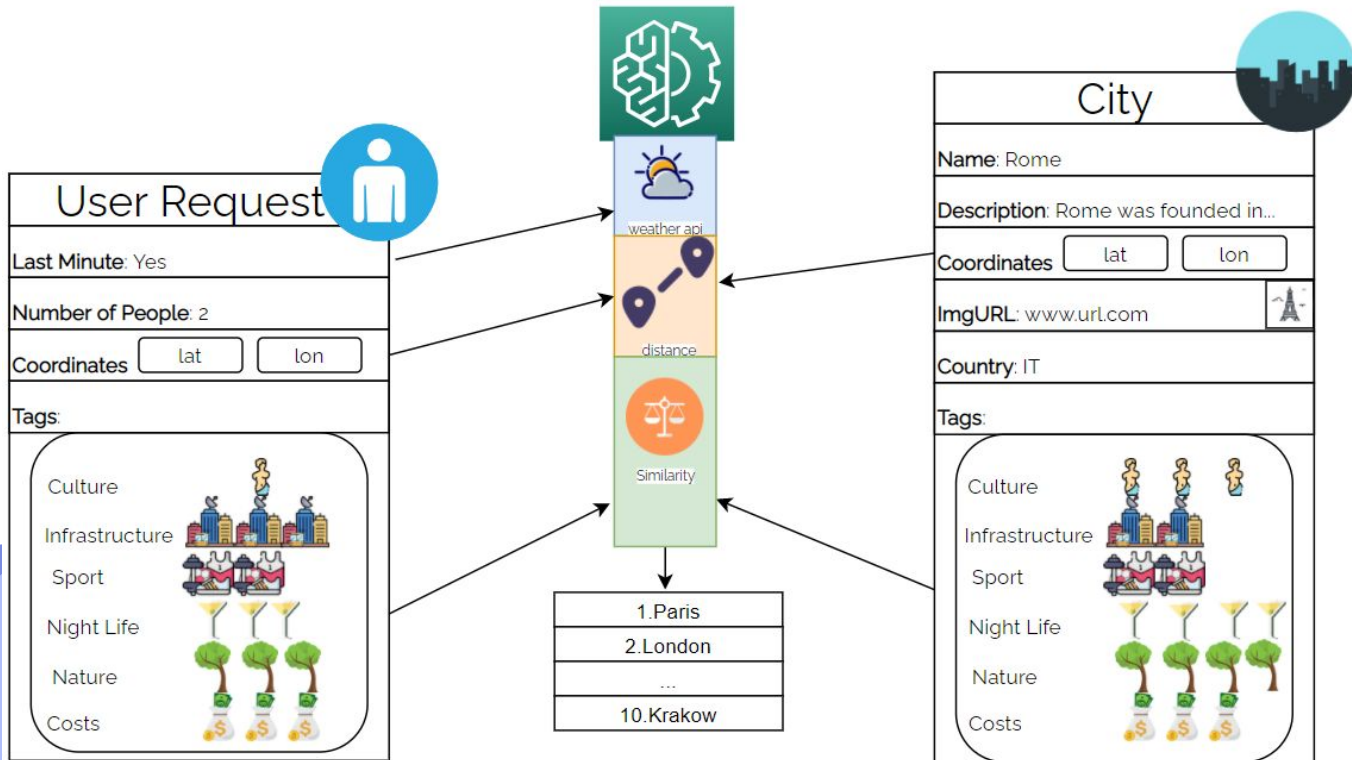
To retrieve informations:

- Scraped websites
- Used wikipedia API

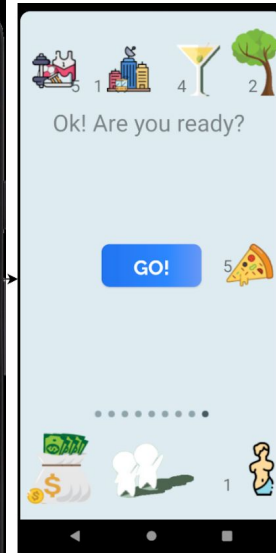
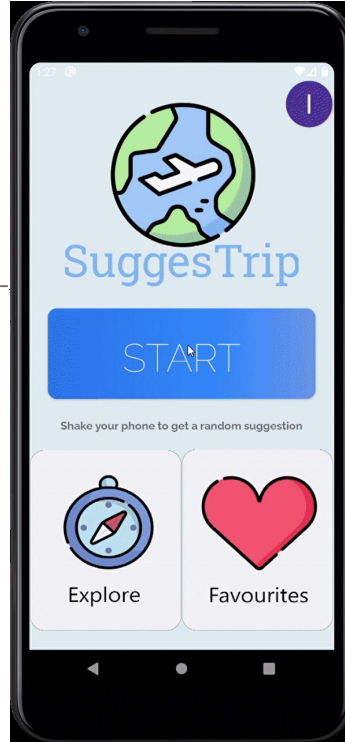
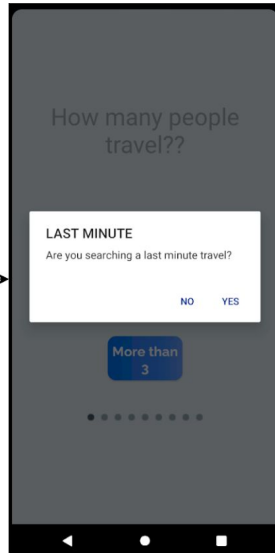
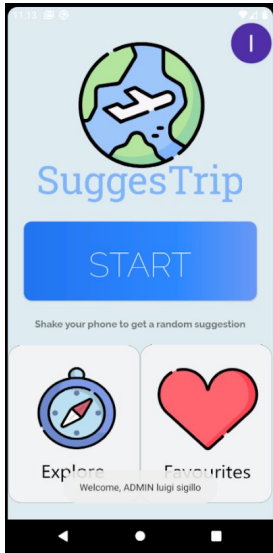


City	
Name: Rome	
Description: Rome was founded in...	
Coordinates	<input type="text" value="lat"/> <input type="text" value="lon"/>
ImgURL: <input type="text" value="www.url.com"/>	
Country: IT	
Tags:	
Culture	
Infrastructure	
Sport	
Night Life	
Nature	
Costs	

# Funct: algorithm



# Profiling and Ranking views



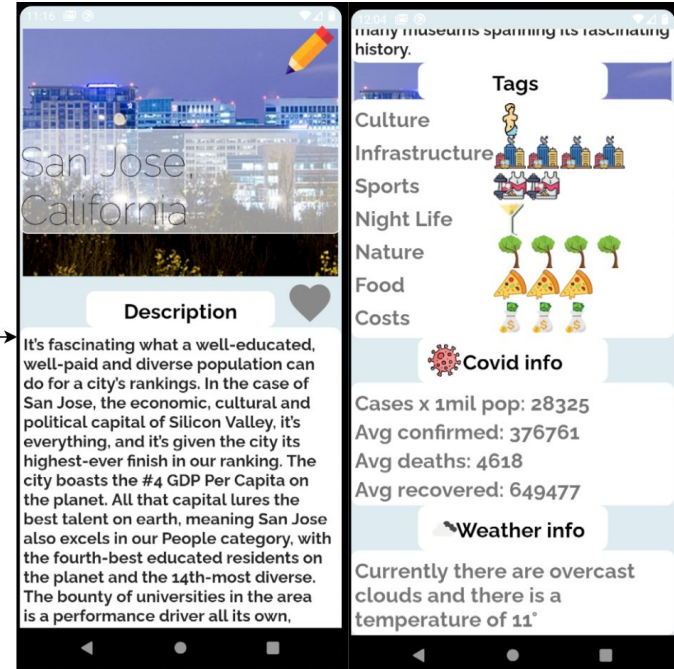
- Slide between fragments using ViewPager



# ExploreView and CityView



- RecyclerView
- Scrollview



# Funct: explore favourites



- SQL lite database is used to store locally the favorited cities

```
fun insertIntoTheDatabase(name: String, url_img: String, id: String?, fav_status: String)
{
    val db: SQLiteDatabase
    db = this.writableDatabase
    val cv = ContentValues()
    cv.put(NAME, name)
    cv.put(IMG_URL, url_img)
    cv.put(ID, id)
    cv.put(FAVORITE_STATUS, fav_status)
    db.insert(TABLE_NAME, null, cv)
    Log.d("FavDB Status", "$name, favstatus - $fav_status - . $cv")
}
```

- After inserted, the item saved is shown in the favourite activity. The user can also delete it from the db

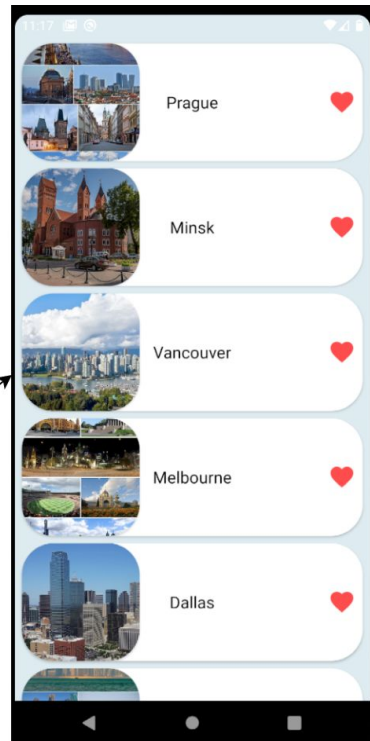
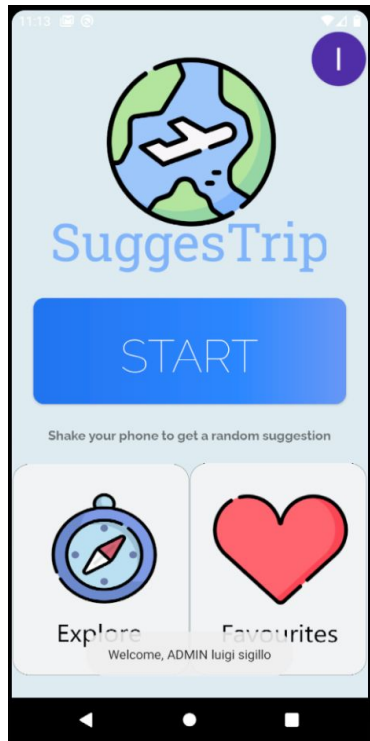
```
fun remove_fav(id: String) {
    val db = this.writableDatabase
    val sql = "DELETE FROM " + TABLE_NAME + " WHERE ID = " + id
    db.execSQL(sql)
    Log.d("remove", id)
}
```



# Favourites View



- List view adapter



# Funct: admin CRUD operations



- Create city
- Update city
- Retrieve city
- Delete city



CREATE

C



READ

R



UPDATE

U



DELETE

D



# Admin views



Creation of the city

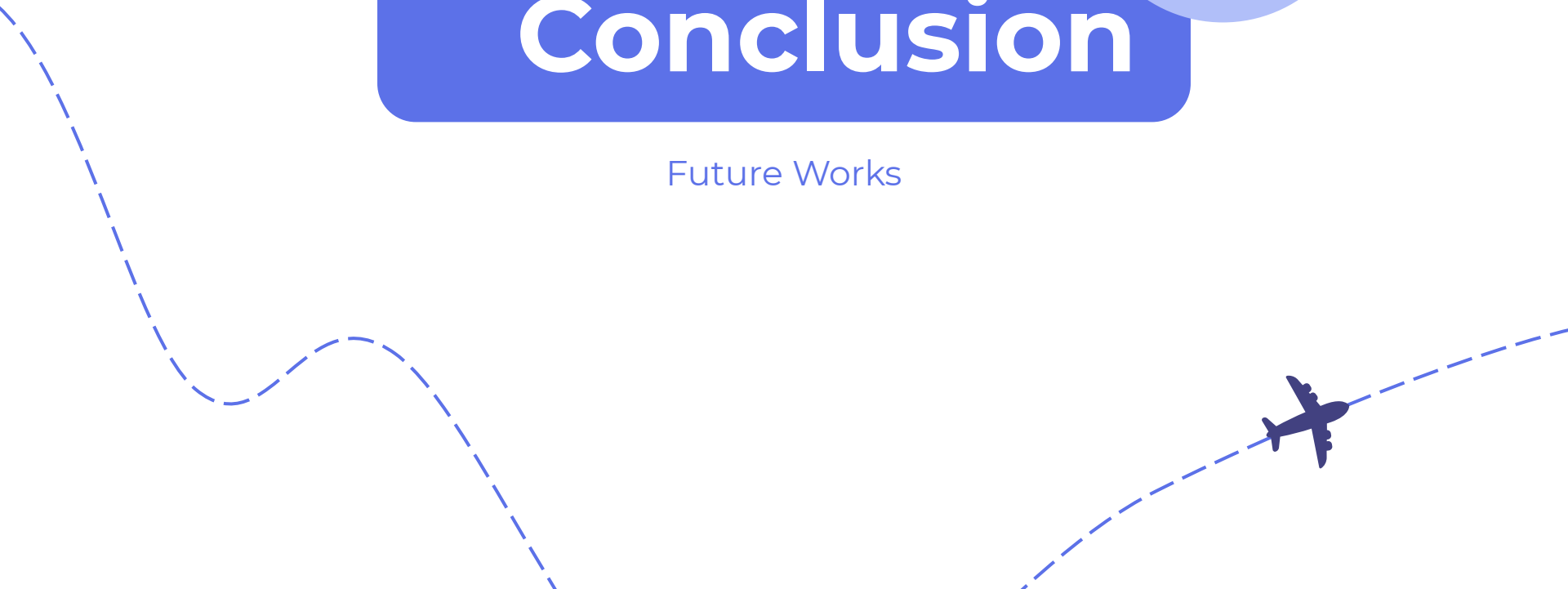


Update and Delete



# Conclusion

Future Works



# Future work



- Expand the cities database and add more interactive media (album, video etc..)
- Improvements of the destination search algorithm
- Add a general more customizable experience
- Publish SuggestTrip on Google Play Store





Thanks

