Mobile Application Development 2020-21

A car pooling application

Lab4 – Position, ratings and aesthetic

Learning objectives

- Using maps
- Add a rating system
- Take care of aesthetics details

Description

You will improve the app created in the previous lab to show locations on maps and rate of user as driver and as passenger.
Furthermore you will finalize your app improving its aesthetic to make it consistent and appealing.
The project will be kept on bitbucket as a private repository.

Steps

1. Open the project you have created in the previous lab.
2. Add two new fragments to the navigation bar.
    a. TripsOfInterestListFragment contains a list of all the trips the logged user is interested in
    b. BoughtTripsListFragment is a list of all the trips the user bought.
    c. Commit the project. Push it onto the remote repository.
3. Update your application using the osmdroid library for maps `https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library-(Kotlin)`.
    a. When a user edits the departure, arrival and eventually intermediate location in an offered trip, he can also pinpoint the position on a map which will be visible to other users when browsing the trip.

    In order to get the geographic location of the tapped point, use the

following code, adapting it as necessary:

```
map.overlays.add(object:Overlay() {
            override fun onSingleTapConfirmed(e: MotionEvent,
                                              mapView: MapView): Boolean {
                val projection =  mapView.projection
                val geoPoint = projection.fromPixels(e.x.toInt(),
                                                     e.y.toInt())
                Log.d("OSM", "${geoPoint.latitude},
                                ${geoPoint.longitude}")
                return true
            }
        })
```

b. Replace the departure, arrival and intermediate stops locations of the trip with a position shown on the map when in “show mode”.

c. When the user is consulting a trip, with a click on the map he can see the route from departure to the arrival location of the trip; for this purpose, use another fragment. The trip is represented as a line from the source location to the destination, possibly passing through intermediate stops.

d. Commit the project. Push it onto the remote repository.

4. Rate the users of the app.

   a. When a trip is terminated, the user can rate the driver, with a number (collected using the five-star methodology) and, optionally, a textual comment. This rate can be given only once for each trip.

   b. Similarly, the driver can rate the passenger, with a number (collected using the five-star methodology) and, optionally, a textual comment. This rate, too, can be assigned only once for each trip.

   c. Update the user profile so that user rating is shown, both as driver and as passenger.

   d. Commit the project. Push it onto the remote repository.

5. Finally, improve the aesthetics.
    a. Choose a logo if you still don't have one.
    b. Be sure that your product has consistency, both in each single page and through pages
        i. icons, buttons, colours, text, language, …
    c. Make the contents simple and flowing.
    d. Make some tests with potential users and use their suggestions to improve your app.
    e. Commit the project. Push it onto the remote repository.
6. Now your project is ready to be presented.

Submission rules

- Lab must be submitted by June, 3th at 23:59
- The functionalities implemented in your code as well as the design of the user interface will be evaluated
- Before submitting, clean the project using *Build -> Clean Project*
- Create a zip file with your project and name it groupXX_lab4.zip
- Upload it on the Polito web portal (only one student of the group must upload it); for multiple uploads, only the most recent file will be evaluated