Team Name	
Member 1	YAO LI
what	**what we evaluate**
Quality of Code	
	Compilable & Executable. We must be able to run your code without any problems on standard lab computers and using standard emulators. All libraries must be included or properly linked via Gradle
	Code style, naming, format (consistent format regarding class and variable naming as well as indentation, consistent format regarding class and variable naming as well as indentation)
	Exception (Proper use and handling of Exceptions in your code)
	Code Organization (e.g. use of Java packages - dcs.shef.ac.uk.com4280.net, distribution of directories etc).

Code: Details	
Querying:	
General Planning and Code Organisation	Separation of concerns
	Use of MVVM model
1.1. 1. Taking photos and Map integration	Ability to take pictures with the camera and upload existing pictures
	if the phone does not have a camera, only picture uploading must be available
1.1.2 Capturing a visit	Tracking of a visit (path) is started and stopped by the user
	Date and time captured automatically when the activity is started
	A title can be associated to the visit
	The tracking is implemented as a service tracking geolocation (gps coordinates), temperature and barometric pressure every 20 seconds
	When the user takes a picture the location is associated to the location
	The App shows a map when the tracking is on
	the map shows the geolocated path taken up to that point
	The map shows the current position
	The map shows the locations of the pictures taken during the trip

	The service works even if the screen is off
	The service works the app is swiped out from the list of recent apps
1.1.3 Visually browse previews of photos	The app allows visualising previews of the pictures
	The interface can show a grid of pictures sorted by date in ascending order
	The interface can show a grid of pictures' thumbnails sorted by path (grid format)
	Allows selecting a picture for further detailed inspection
	The interface is efficient and able to cope with a library of thousands of photos
1.1.4 Persisting data in a local database	all the data is persisted into a Room database
1.1.5.Showing details of a photo including its location on a map	When clicking on a location with photo, shows the details of the photo
	Implementing searching for locations or showing a preview of all the images shown
Additional Features	Have you implemented any feature that was not required by the brief but that show the quality of your solution?
	Quality of the solution

Quality consideration	
	Quality of the interface (e.g. use of appropriate elements, no confusing views, etc.)
	Appropriate use of views, transitions and navigation item
	Portability (what versions of Android are you covering?)
	any other
Documentation	
	Code documentation: JavaDoc (in-line commenting, javadoc)
Division of work	
	detail the division of work of each member. Use the same level of details described in the assignment text

	1
	2
	3
	4
	5
Insert your value in the drop down cells (1- low, 2 - a little bit, 3 fair, 4 - quite good, 5 - high)	**Comments**
5	It works without any stress
5	**All Java part is followed with Oracles' naming convention(https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html), all layout is named by lowercase with words separated**
5	**Exception catchers can be seen among most classes where possibly exist any**
5	**Yes , its called uk.ac.shef.dcs.travelguider**

	Required note: how did you implement the separation of concerns between interface, sensors and database? Describe the details of your model: have you implemented the MVVM model? How did you implement it? In what way the model is correctly implemented? Add a diagram if necessary
5	*This implementation strictly followed the concept of MVVM, all views are in the directory of \(*
5	It works perfectly
5	No problem found
5	There is a noticeable button for user to use
5	Yes, the date and time will be automatically fetched and a timer will be generated when a new visit starts.
5	**The application will prompt the user to enter a title when the user try to**
5	Yes, it's automatically tracking and the temperature/pressure values will be fed from 3-rd party APISs. (https://openweathermap.org/api). The location will be updated in every 10 seconds or every 10 meters the user moved.
5	**Yes and the location will be stored into the room database as a set of values of latitude and longitude**
5	**Yes, and the home page is also the map**
5	Yes, it's achieved by using Polyline
5	Yes, a button is also provided for users manually locate them and play around
5	Yes, and the user can click on it to see the clearer version of the picture taken

3	**Not sure if it works, but a location service is implemented**
2	*Not sure*

5	**Yes, both on the map and the gallery fragment**
3	**It can only display in descending order by default**
1	No, don't have this feature
5	When a thumbnail is clicked on the gallery fragment, the title of visit, data of taking the photo and the photo file path will show along
3	**Haven't tested thousand of photos but it's reliable since all photos are loaded as thumbnails.**

5	**Yes, and all files related to the database in the /database folder**
5	Works as desired
4	**When a photo is clicked on the gallery fragment, a map will be provided to show where the photo was taken.**
4	**A floating button is provided to change from normal mode to the night mode. A customised theme file is implemented as well as some icons. The UI is followed with the concept of material design. (https://material.io)**
4	*add your note here*

4	**A bottom navigation bar is provided with 3 fragments. Whenever there could be confusing to the user, a toast message will present with detailed information.**
5	*There’re 3 fragments clearly designed and labeled; a bottom navigation bar is*
3	Only tested on Android API 30 and 31 as my computer is not compatible with lower android
3	
4	**All confusing parts were well-commented with in-line comments**
	I implemented this solution all on my own

no					
a little bit					
fair					
quite good					
perfect					

