CityScape – a mobile application for City breaks management

1. Short description

People's lack of time is becoming an impediment when it comes to scheduling and planning trips these days. More and more people seem to be feeling overwhelmed by the multitude of data they must consider while making travel-related decisions. Having all of these in mind, we want to take advantage of technology and simplify the process of organizing and monitoring people's wish lists for city breaks. For this reason, CityScape comes as a rescue for helping people to keep track and organize their vacations as well as making them fall in love with travelling and exploring the world.

1. Domain details

CityBreak: city, country, startDate, endDate, description, accommodation, budget

- city: represents the city where the vacation takes place

- country: represents the country where the city is located

- startDate: represents the starting day of the city break

- endDate: represents the ending day of the city break

- description: represents a brief description of the city and its attractions

- accommodation: represents the hotel where the user will be staying during the specific city break

- budget: represents the amount of money the user has decided to spend during the city break

1. CRUD: details of each crud operation

- create: this operation allows the user to create a new city break and can be acquired by clicking on the plus icon button situated in the bottom right corner. The user will be redirected to a new page where the information for the city break needs to be completed. The newly created city break will then be submitted to the list by pressing on the “Add” button. There is also an option available for canceling the adding by clicking on the “Cancel” button.

- read: this operation allows the user to view the contents of a specific city break. This can be obtained by clicking on a specific city break from the whole list that is displayed on the home screen and see its information

- update: this operation allows the user to change the contents of the city break fields. This will be acquired by clicking on the update button, situated at the right of the city break that is supposed to be modified and then being redirected to a page where any field of the city break can be modified and then submitted by pressing on the “Update” button.

- delete: this operation allows the user to remove a city break by clicking on the garbage icon next to the specific city break. Then a window will pop up and ask the user if he wants to cancel the operation or continue deleting the item. If the user clicks “Yes” and persists the operation the city break will be deleted, otherwise the operation will be cancelled and the list will remain the same.

4. Persistence details

The persistence is a very important aspect when it comes to this application so it should be linked to both a local database and a server making it accessible online and offline too. We want to save changes made while adding, editing, or removing a city break in both directions so they will be maintained on both the local database and the server. When the user is online the updates are loaded immediately, in real time, from the server. Also, when the user is offline, the temporary data is retrieved from the local database.

1. Details on what is happening when the device is online

Having the possibility of being offline in mind, the goal is to create an application that enables local entry addition, deletion and modification by the user. Changes made in this manner will only be stored locally in the database.

Every action taken while using the application offline saves the data locally.

- The newly created entities are stored on the local database.

- The read operation is executed on the locally saved entities.

- The update operation is working for the locally saved entities

- The delete operation is executed on the local database

When the application goes back online, all the changes on the local database are received by the server.

When there is no internet, you can only access the city breaks with the properties previously saved in the database.

1. Mock-up