CTPCJ - A Bus Management Application

1. *A short description. Note that the client is not having a technical background.*

CTPCJ is the public transport company of Cluj-Napoca - one of the most digitalised cities in Romania. Following the trends of the city, it was imminent that the company switched from an analog way of keeping track of its vehicles to a digital one. Thus, the need of an application which allows having an overview of its buses, trolleybuses and tramways. This way, the company can manage the new acquisitions and the changes of the fleet.

1. *Domain details. Present the details of the entities that will be persisted. Each field should have a description*.

At this moment, the entity of the domain is the Vehicle, which has the following 5 fields:

* The Picture (it is an image of the respective vehicle)
* The Line (on what line does the vehicle run at the moment e.g 47, 101..)
* The Start (Capăt traseu 1 - the departing station of the vehicle)
* The End (Capăt traseu 2 - the arriving station of the vehicle)
* The Type (what kind of vehicle it is e.g tram, diesel bus…)

1. *CRUD. Present the details of each crud operation. - create. - read. - update. - delete.*

**Create:** The user has to press the + (plus) button on the main page (*Screen 1*) in order to add a new vehicle. After this step, a new page opens (*Screen 2*), where the user must add an image of the vehicle. Besides that, they should select the line, the start, the end, and the type of the vehicle from each drop down list. For the entity to be created, the user must press the Save button at the bottom of the screen.

**Read:** The Read operation is performed twice. Once, when all the entries are retrieved and displayed on the main page (*Screen 1*), and then when the user presses a specific vehicle in order to perform modifications (*Screen 3*). The difference between these two operations is that the first one retrieves all the entities, while the second one retrieves only a specific entity.

**Update:** The user has to press a specific vehicle in order to update it. After they do so, a new page opens (*Screen 3*), where they can notice all the completed fields of the given vehicle. For it to be updated, the user must choose a different entry for the desired field(s), and then press the Save button at the bottom of the screen.

**Delete:** The user has to press a specific vehicle in order to delete it. After they do so, a new page opens (*Screen 3*), where they can notice all the completed fields of the given vehicle. For it to be deleted, the user must press the Trash bin button, on the top right corner of the screen. After pressing it, the user will be redirected to the main page automatically.

1. *Persistence details, what crud operations are persisted on the local db and on the server.*

*Ideally, all operations are persisted both on the local db + the server. The only difference between local and remote occur when the application goes offline. see the next section.*

**Create:** Whenever a new entity is created, it is added both in the database and on the server.

**Read:** When online, the application reads the entities from the server. When offline, it reads them from the database.

**Update:** Whenever an entity is updated, the updates are transmitted both to the database and to the server.

**Update:** Whenever an entity is deleted, it gets deleted both from the database and from the server.

1. *Details on what is happening when the device is offline*.

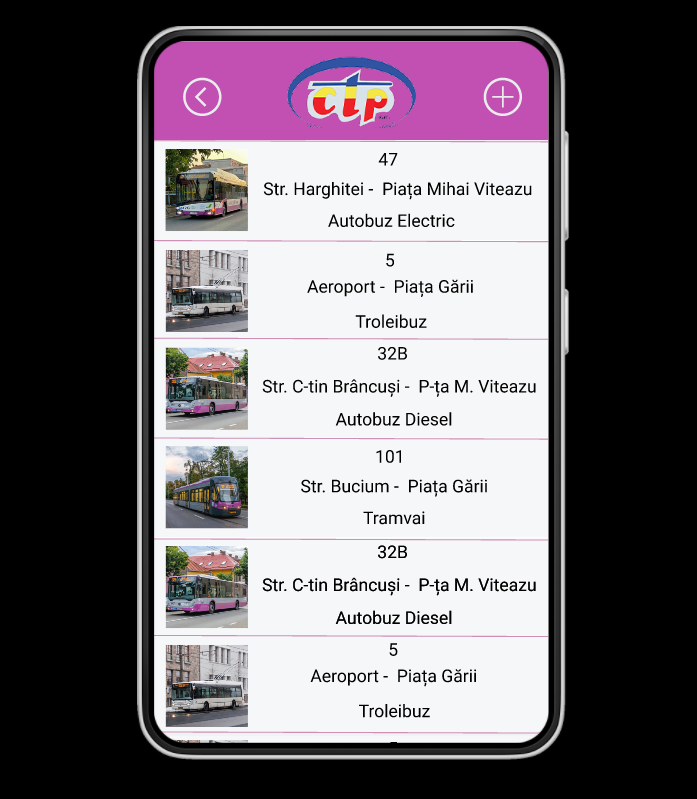
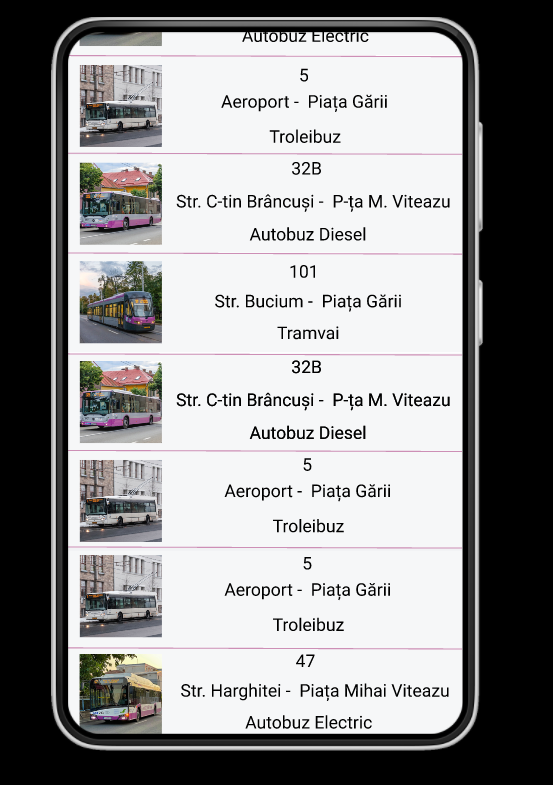
**For this scenario, we suppose that, at this point, there is one single user using this application, in order to avoid conflicts. We also assume that the local db and the remote server are connected.**

When the application is used offline, all the operations performed store the information locally:

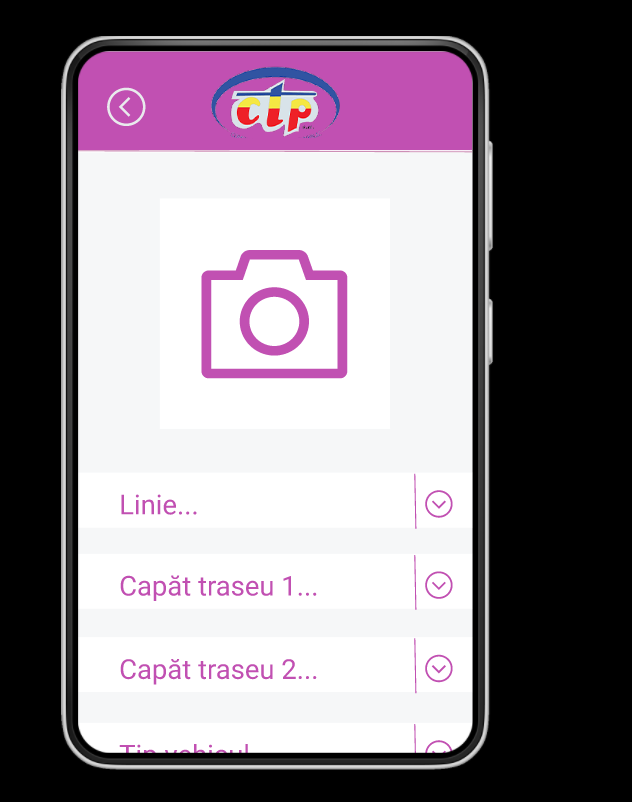
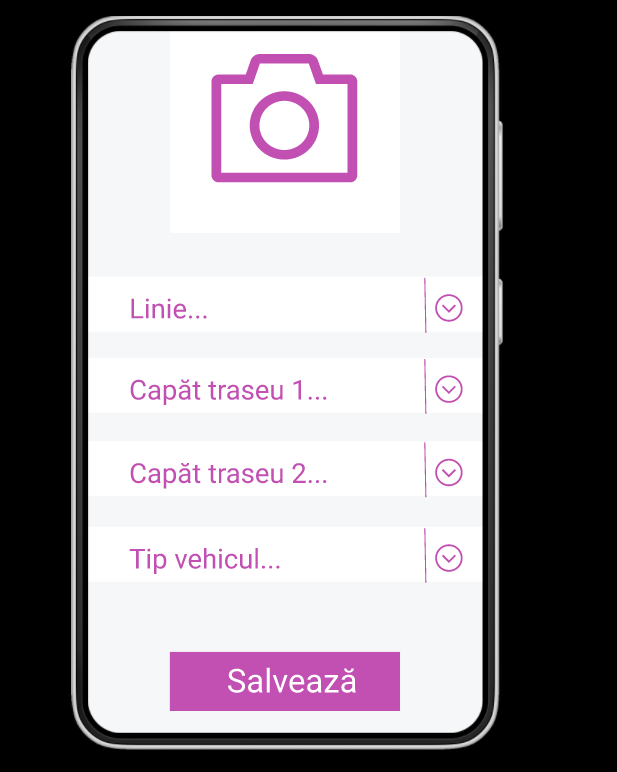
* the entities created are stored in the local database
* the read operation is performed on the locally saved entities
* the update operation goes for the locally saved entities
* the delete operation is performed on the local database

When the application goes back online, all the changes on the local database are transmitted to the server.

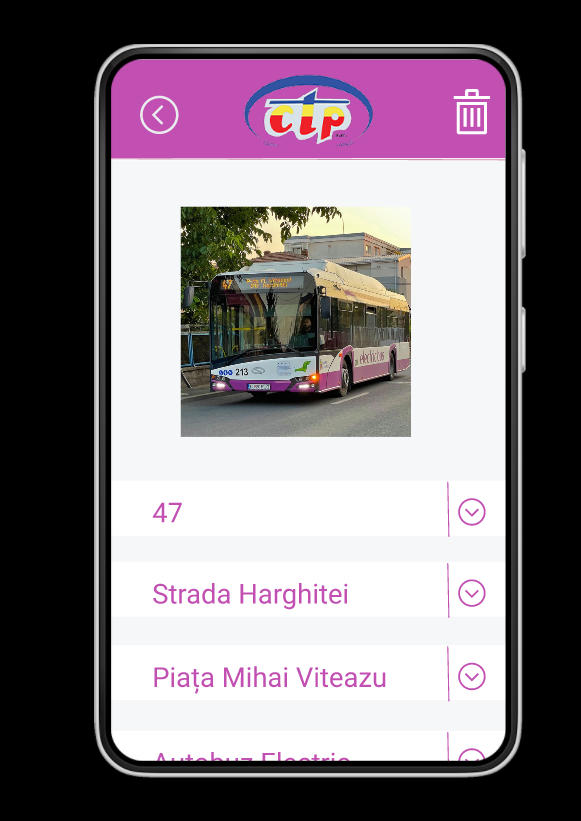
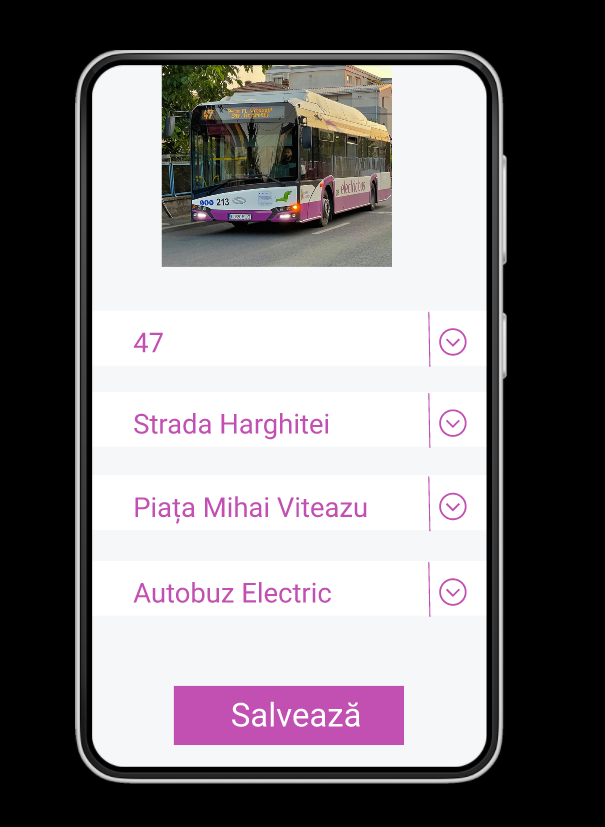
1. *App mockup. Eg. using figma.com or sketch.com or something equivalent. The screenshots should be attached to the assignment.*

* *

*Screen 1*

* *

*Screen 2*

* *

*Screen 3*