

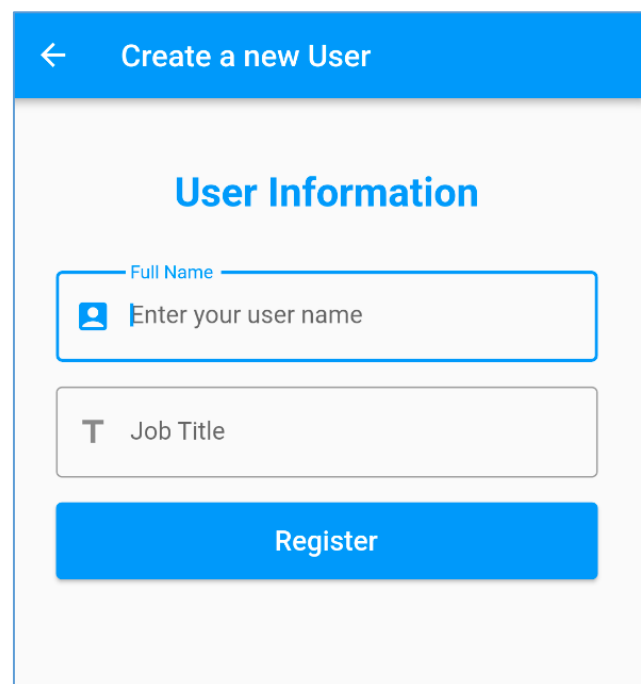
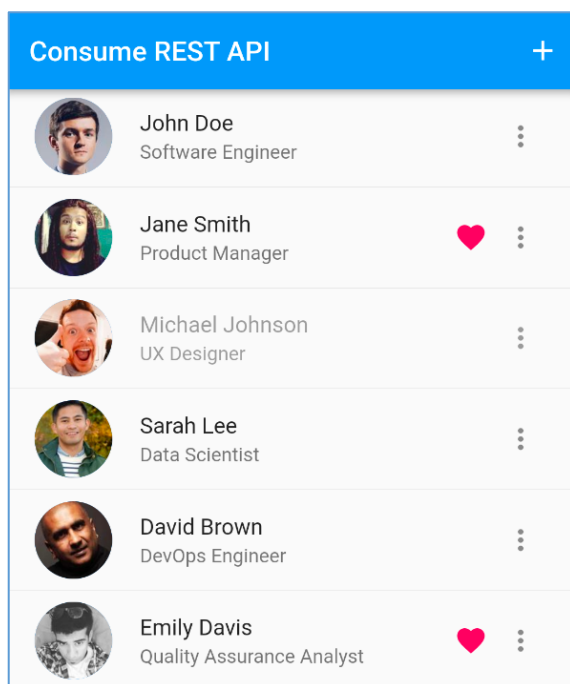
CROSS-PLATFORM MOBILE APP DEVELOPMENT

(503107)

LAB 9

EXERCISE 1

Based on the given source code, you should develop a user management application with basic functions such as displaying a list of users, adding a new user, updating a user and deleting a user. The application does not store data locally, but retrieves data from the server through the restful web service.



Use the given data.json file as data for the restful web service. You will not need to manually code a web service from scratch, instead you will use the JSON Server tool to initialize a web service with the data.json file provided. To install and use JSON Server you need to follow the instructions below.

```
npm install -g json-server # install JSON Server (nodejs required)
json-server /path/to/data.json -d 1000 # delay 1 second
```

By default, the server will be available at <http://localhost:3000/>. Note that if you use the Android Emulator for running your application, you must use <http://10.0.2.2:3000/> instead of localhost. To interact with the api endpoints, send http requests to the address in the summary table below:

GET	/users	# read all users
GET	/users /1	# read user by id
POST	/users	# add a new user (json body)
PUT	/users/1	# replace a user by id
PATCH	/users/1	# update a user property
DELETE	/users/1	# remove a user

When using the given source code, you need to run the `flutter pub add cached_network_image` command to install the required library. In addition to consuming the web service, you also need to install the `http` package or `retrofit` package (and related packages) with the following commands:

```
flutter pub add http dio retrofit json_annotation
flutter pub add --dev build_runner retrofit_generator
```

Because there is a delay when sending requests to the api server, you need to use `FutureBuilder` widget, `CircularProgressIndicator` widget or other types of progress notifications to make the interface more intuitive. Make sure to let the user confirm via an alert dialog before actually deleting a user.

Everytime make changes to the ApiService class, make sure to run the following commands:

```
dart run build_runner build
flutter pub run build_runner build
```

EXERCISE 2

Use the **LayoutBuilder** widget to create a **responsive** interface that displays a **GridView** with a different number of items per line depending on the size of the screen. Every time the application window is resized horizontally, the number of items per line will change and their background color will also change accordingly.



An example of flutter application using gridview displaying 1, 2 and 4 items in a row, depending on the current horizontal size of the application window.