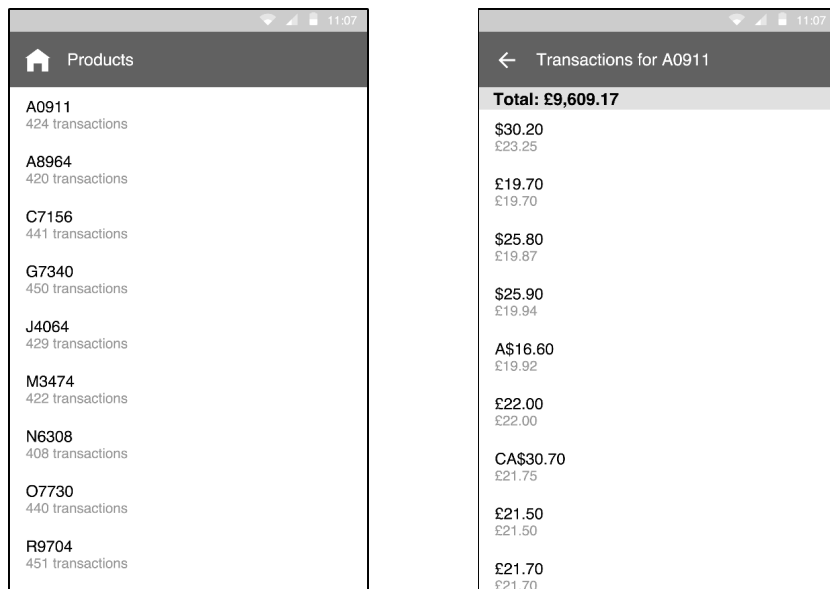# Transaction viewer

Your manager has asked you to design and implement a mobile application to help the firm executives who are always flying around the globe. They need a list of every product the company trades with, and the sales of those products, which are made in different currencies.

Your application should be composed of two screens:
1. In the first one, your application should
   - give the users the choice of which product they want to see, and
   - show the count of transactions for that product.
2. When the user selects a product, the application must show
   - each of the transactions related to that product,
   - the amount of each transaction converted to GBP and
   - the sum of all of these transactions in GBP.

11:07	11:07
🏠 Products	**← Transactions for A0911**
A0911 424 transactions	**Total: £9,609.17**
A8964 420 transactions	$30.20 £23.25
C7156 441 transactions	£19.70 £19.70
G7340 450 transactions	$25.80 £19.87
J4064 429 transactions	$25.90 £19.94
M3474 422 transactions	A$16.60 £19.92
N6308 408 transactions	£22.00 £22.00
O7730 440 transactions	CA$30.70 £21.75
R9704 451 transactions	£21.50 £21.50
	£21.70 £21.70

In a real environment, you will probably download all the data you need from the Internet.

To simplify this task, so you don't need to worry about network management, your application should be designed to be able to work with two JSON files:
- **transactions.json:** holds an array of dictionaries where each one represents a transaction related to a product, indicated by its stock keeping unit (SKU), with a given currency and amount.
- **rates.json:** specifies the conversion rate from one currency to another.

Please note that:
- You should aim to develop a working solution with the simplest possible user interface.
- Some conversions may not be specified in rates.json. In case they are needed, they should be calculated using the known conversions.
- Two different sample data sets are included so you can test your solution with different scenarios.
- Other sets of data will be used to validate your solution, which should be general enough to handle all the possible different scenarios, including empty data files and other common errors.

Good luck!