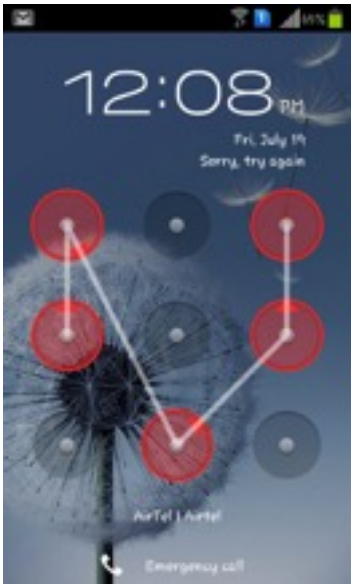
Pattern lock problem

Test exercise is to create a pattern lock.

Pattern lock will work the same as Android pattern lock:



The problem consists of two parts:

- Unlocking the application
- Changing a pattern

Unlocking the application

When the application is started, it will ask for a pattern. After user enters a pattern, this is sent to a backend server for verification. Meanwhile the application needs to lock itself (that is not allow any more input) until the backend responds (or until the request fails).

Response from the backend will simply indicate whether the pattern was entered correctly or not.

Application must be able to handle both positive and negative responses from the backend as well as request errors (e.g. connection problems).

Changing a pattern

The application will allow to change the unlock pattern. This will be possible only after unlocking the application with a valid pattern. The default pattern when the application starts is the same as on the image above.

For the pattern to be successfully changed, the same pattern has to be entered twice in a similar manner as Android phones work.

When the same pattern is entered twice, the new pattern should be sent to the backend server to be stored. If the new pattern matches certain criteria, the backend server will accept the change. Otherwise it will refuse to store the pattern and will provide a reason. (e.g. “Pattern too short.”).

Backend server

The backend server is not part of this task and the frontend application for the purpose of this test exercise should contain a mocked version of it.

The backend mock should simulate a real server, that is:

- Be able to respond both positive and negative answers for unlocking the application
- Keep the state information if application is locked or unlocked
- Have a delay before negative response when unlocking the application
- Be able to accept a pattern change request, when
    - Application is unlocked
    - Pattern meets the criteria

The API between the application and the backend server can be defined at will.

Pattern criteria

A pattern is an undivided continuous line connecting dots from a starting point to an end point and no dot can be used more than once.

These criteria should be enforced both by frontend and backend part.

Moreover, backend part will restrict the pattern to be at least 3 characters long. Shorter patterns will be refused with a reason.