MOBILE COMPUTING

First call - Example

(total 20.0) **duration: 2h**

1. [4.0] *Android* applications can contain several components, and between them several activities, that can be invoked from within the application or even from other applications.

a) Activity invocation in *Android* uses a specific mechanism. Give a description of that mechanism, considering the several variants, especially when the invocation comes from within the application or externally.

b) Many times, an initiating activity receives data needed for its internal state. Explain the possible origin of that data and the mechanism of its transmission. Are there any limitations?

c) Also, in the reverse sense, when an activity ends and returns to the one that has invoked it, it could be convenient and possible to transmit some data. Describe some ways of doing that transmission.

d) How can we share information between application components? And if the components are in different applications?

2. [3.7] Many activity screens in *Android* are built using predefined interface controls. But, sometimes, the programmer needs interface functionality not available in those controls.

a) Describe one way, available in *Android*, to build activity screens by drawing them directly. How can the activity cause its redrawing, for instance when there is a change in its internal state? Explain.

b) Explain how the user interacts with that kind of screen. Which appropriate events should be handled?

c) To execute a scaling of a visualized drawing in the screen a positive scale factor greater than 1 (*zoom-in*) or less than 1 (*zoom-out*) should multiply the vertex coordinates. Supposing that you know the position of all fingers in contact with the screen, propose a simple calculation that, in each finger movement, provides an appropriate scale factor value. Explain.

3. [3.8] Many graphic user interface (GUI) *frameworks*, including the mobile platforms, use the *Inversion of Control* (IoC) pattern, where the main process *thread* detects screen user interactions and invokes *handlers* (or *callbacks*) available in the application code. But, in some situations, this pattern can become inefficient and not respondent.

a) State some circumstances where those possible inefficiencies appear and explain why. Give some concrete examples.

b) What is the usual way of solving this kind of inefficiencies?

c) When an application processes these user interactions, new data can be obtained, leading to a change in the screen content. In the way you have indicated in the previous answer, is there any difficulty in dealing with this situation? Explain. If yes, how can you solve the difficulty?

4. [3.5] In many development platforms for mobile the use of the pattern known as MVC+S is very frequent.

a) Briefly describe that pattern and relate it with the data transfer between the user interface and the application business logic. Also describe the transfer mechanism.

b) Many mobile applications show collections of items to the user, allowing operations such as adding, modifying or eliminating an item. Describe how *Flutter* can ease the implementation of those operations.

(please turn the page)

5. [5.0] A museum shows a big set of art objects in several rooms, that can be observed by following one of the available room routes. Within each room several objects are exhibited. For each of them a set of multimedia material was prepared, which can be played in a mobile device accessing the museum server. For each object there is also a small survey consisting in yes/no questions or choosing one option from a small set. It is intended that each visitor answers the survey after observing the corresponding object.

When a visitor arrives at the museum he can borrow a device already running the app prepared, or he can install it in his own device, from the museum wireless network. The app allows the user to choose a route and observe any object within each room (by any order). After identifying the object (there should be an expedite way to do that) he can play the corresponding multimedia material and answer the survey. At the end, the borrowed devices are returned. The museum is interested in the statistics calculated from the survey answers.

i. Draw the architecture and interface diagrams for this app, stating the possible user actions, and the app navigation graph.

ii. What operations should be implemented in the museum server? How the app and server should work together?

iii. Discuss the information needed to circulate between app and server, and its format. What information should be persisted and processed by the server?

iv. Discuss other relevant aspects namely, how to orient the visitor inside the museum, and how to automatically play the multimedia content associated to an object. Also state any needed device characteristics, in the implementation of these aspects.