Università
della
Svizzera
italiana

Faculty
of Informatics

Lecture Mobile Computing, fall term 2016
Android Programming Project – Project description

Group name:	Yet Another Project Team
Group members:	Jacques Dafflon, Samantha Rosso

Instructions

- Please answer the questions reported in the following two pages using the foreseen text boxes.
- Do not change the format, font size or any other elements of this template.
- Be concise and precise.
- Do not exceed the given limit of maximal number of characters. The given limits are intended including white spaces (e.g., the sentence “*This is a nice sentence*” contains 23 characters).
- “The app” mentioned in the questions refers to the Android-based application that you develop in the context of the Mobile Computing class.
- Once compiled, please upload this form in iCorsi in the **"Upload APP Project Description Form"**.
- The deadline for uploading the document is Monday, November 14, 2016, 23:59 CEST
- If you have questions: post your question(s) in the iCorsi forum.

1. **What is the name of your app?**

PARC (Phone As a Remote Control)

2. **Which problem does the app solve?** (Max. 200 characters)

When sitting at your pc, your phone is left unused, PARC solves this by turning it into an extra input for your pc.

3. **How does your app solve this problem?** (Max. 650 characters)

The app detect hovers of the hand in proximity of the phone. Different motions will trigger different commands which are sent to the computer over bluetooth. A small program on the computer will execute those commands.

Optionally if the bluetooth connection is lost, the computer will automatically lock itself. Once the connection is restored, the computer will unlock.

4. **Why is this problem relevant?** (Max. 300 characters)

The phone has interesting sensors which can augment the user's interaction with the computer using a keyboard and mouse. So far, those sensors remain unused on the desk right next to the computer.

5. **Do other apps exist to solve this (or a very similar) problem?**

Yes ● No ○

6. **If you answered *yes* to question 5, list the existing apps that are most related to yours and explain how these solutions differ from your own. If you answered *no* to question 5, explain why do you think nobody else has solved this problem before.** (Max. 650 characters)

Chrome Remote Desktop and Unified Remote are exampels of apps which allow a user to control the computer from an Android phone. However those try to either replace the keyboard and mouse for a true remote control or emulate buttons on the screen for the user to press. Both options require the user to stop and pick up the phone to perform an action.
The idea behind PARC is to complement the keyboard and mouse and allow the user to blindly use it without picking the phone or even looking at it.

7. **Which of the built-in sensors of your phone does the app make use of?** (Max. 200 char.)

Proximity, Bluetooth, (optional: Accelerometer, Touch, whichever "sensor" used to detect if the phone is charging)

8. **Which of the built-in actuators of your phone does the app make use of?** (Max. 200 char.)

Vibrator, Bluetooth (to send data), (optional: speaker, screen)

9. **Does your app store sensor data locally, remotely, or in both ways?**

Locally ● Remotely ○ Both ○

10. **Motivate your answer to question 9 (i.e., explain why your app stores data and why it does so only locally/remotely or in both ways).** (Max. 650 characters)

The app does not store sensor data per say, as readings from the sensor will trigger actions which are sent to the computer over bluetooth. Settings to know which action to send based on the sensor readings will be stored locally
on the phone.
Since those are specific settings for the app which do not need to be shared, it makes sense to simply store them locally on the phone.

11. **Does your app use any type of data visualization?**

Yes ○ No ●

12. **If you answered *yes* to question 11, explain which type of data visualization your app offers and why. If you answered *no* to question 11, explain why data visualization is not necessary for your app.** (Max. 650 characters)

As mentioned before, the app does not store sensor data, there is nothing much to visualize, however some sensor readings could generate activities on the phone. For example loosing the bluetooth connection could make the phone vibrate or even send a sound.

13. **Does your app perform any type of data processing on the collected sensor data?**

Yes ● No ○

14. **If you answered *yes* to question 13 explain which type of data processing your app performs on the collected data and why. If you answered *no* to question 13 explain why data processing is not necessary for your app.** (Max. 650 characters)

The app will need to process data from the proximity sensor to determine if the hand swiped over the phone once, twice or remained static over the phone.
The app will also need to analyse the strength of the bluetooth signal and use the strength as an indication of the distance between the phone and the computer. This wil be used to lock the computer once the phone reaches a certain distance away from the computer.

15. **How do you evaluate whether your app performs correctly and achieves its goal (i.e., solves the problem described in question 2)?** (Max. 650 characters)

By measuring the number of time gestures trigger the correct action on the phone as well as the number of time the computer locks when it is supposed to and does not when it is not supposed to.

16. **Does your app use any third-party services? (E.g., Google maps, Google App Engine, etc.)**

Yes ○ No ●

17. **If you answered *yes* to question 16, list the services your app makes use of.** (Max. 200 char.)

18. **Which permissions does your app require to be granted by the user?** (Max. 200 characters)

BATTERY_STATS, BLUETOOTH_ADMIN, BLUETOOTH_PRIVILEDGED, VIBRATE, (optional: WAKE_LOCK)

19. **Which are the main challenges that you need to overcome in order to build your app?** (Max. 300 characters)

Properly process data from the proximity sensor.
Properly correlate the bluetooth signal strength with distance.
Handle bringing the app from the backgroud to the foreground and back seamlessly in every situation.