

# Comments on the reports

Overall, I am very impressed by the quality of your submissions. It is possible to read from the reports how proud you are of your achievements and this is something we like to hear. I have summarised a few issues with the reports that will hopefully help you to improve your technical writing skills and will ultimately result in better grades for your other projects (especially thesis). Please note that none of the examples in this document are actually taken directly from your reports but I have paraphrased existing problems slightly to emphasise the point. Also, do take these comments as mere hints and suggestions. The list is by no means complete and you won't be able to fit everything into a short report such as the one required for this exercise.

## Too much prose

A lot of you tend to write prose. This makes for an easy read and is very adequate for a publication targeting the general public. However, when writing reports for engineers, phrases like "And then the culprit would probably go home" or "that's why our app is so cool" are somewhat off the mark. Prose will lead your readers to miss a lot of the detail of your implementation and that is no good if you want to convince them of your great features. Most markers, supervisors, project managers ... will have very little time to read your report. So be as concise as possible. Think about what you include. And never, ever ask the marker for leniency in your report - send us an email if you had any problems.

## No sectioning

An alternative to writing large chunks of text, which inevitably lead to a more prosaic style, is starting by putting down headers. This enables you to plan ahead what to include into the report. One possible organisation could be to discuss the project in terms of a simple project life cycle. We have given you the requirements, so you can start by discussing design and implementation (with the use of appropriate diagrams) and then go on to testing. Make sure sections and paragraphs are tied together by matching first and last sentences. This gives the report a better "flow".

Alternatively, you could split the report up and discuss "features" in separate sections. But here again, you should start with motivation and design choices before you move on to implementation issues. Always remember to test and evaluate your features.

## Little space dedicated to design considerations

I enjoyed reading your comparisons of Android to other systems such as J2ME or the various Symbian flavours. In a real technical report, this part probably fits well into the design stage where you motivate your choice of platform and programming language. It also gives the reader an insight into what to expect when it comes to the implementation. So maybe instead of just adding it to the end of the report as some kind of afterthought - start with it!

The problem is that a lot of you have jumped straight into implementation without giving much room to the design stage. Of course our requirements were quite rigid and left little room for creativity, but there were a lot of small gaps where your design choices made the difference. Since you were required to design the AntiTheft algorithm, it would have been nice to have a little diagram (state chart), describing its operation. Some of you used quite sophisticated methods, which would have made for a very impressive report. When you just describe them textually, some of the cool design decisions get lost for the reader.

Also, when evaluating your design, mention other ideas or solutions. Criticise your own approach. You can always write it in a way that makes your own solution come out top. But do evaluate your ideas. Nothing is more than annoying than reading "our solution is good because it is well cool".

## Testing

Only a very few of you have actually mentioned testing in the report. I know a lot of you had bugs to fix (or to fight the peculiarities of the Android platform...) - so how did you go about doing it? And have you considered tools such as unit testing? Remember, even the use of the emulator can be part of the testing process. And since this application is very dependent on the UI and the usability of the alarm, you could have described (or even conducted), a little field experiment to test it.

## Figures / Listings

Diagrams were far and few between in your submissions. As mentioned in the section on design considerations, you should really try to express your ideas with simple diagrams. That way both you and your reader can follow the text much better. You can use UML diagrams to highlight your components and/or the control flow of the program. For example, during the initial design phase, you can start with a use-case diagram to tell yourself and the reader about the main objectives of your program.

If you feel that it is important to discuss screenshots, be aware that the report is not a user-manual. Only discuss features that are novel and specific to your design. A user manual can always be added separately - see next section.

## Code listings

The same applies to pseudo (real) code examples in your reports. They appeared more often but in many cases without an accompanying description. It is an absolute no-no to include a code listing without explaining what it does. Diagrams and code listings must be referenced and discussed in the text.

## Appendix

I liked the level of detail in which you explained how to use and install the app - as some of you used external libraries this is a very important point and should not be missed out in a technical report. However, if you wish to include these, please move them into an appendix. We don't mind you writing more than a single page if you feel the need to explain how to use and install the app. Just make sure this part is actually separated from the main report.

## References / Bibliography

A small minority of you has referenced sources in the report. I can't believe that all of you have finished this task merely on the live-hacking session and the slides. If you do use other resources, you should reference them accordingly. Especially for the "sensor readings in standby" problem there are a lot of resources on the Internet which any meaningful discussion should include.

Of course, references are to be collected in bibliographic style and appended to the end of the report. They do not count towards the 1 page limit.

**Last but not least: Administrative issues**

**Name and group no** Always include your full name and group number on the report. The group number is the number written on the back of your phone.

**English or German** Whatever you feel most comfortable with. Spellcheck both.