



# Review report of a final thesis

**Reviewer:** Ing. Jiří Daněček  
**Student:** Azad Mamiyev  
**Thesis title:** Comparison of different architecture approaches on Android OS  
**Branch / specialization:** Web and Software Engineering  
**Created on:** 31 May 2021

## Evaluation criteria

### 1. Fulfillment of the assignment

- ▶ [1] assignment fulfilled
- [2] assignment fulfilled with minor objections
- [3] assignment fulfilled with major objections
- [4] assignment not fulfilled

Assignment fulfilled.

### 2. Main written part

70/100 (C)

The description of individual UI models is insufficient and messy. The student failed to present the information which he apparently drew from different materials by a clear and uniform form. He uses a different way to describe individual models: once UML diagrams, sometimes informal diagrams. The text lacks one unifying simple example, on which the principles of individual models would be illustrated. Such an example ("setting username") is listed on page 10 only for the MVVM architecture, but without exact specification, no illustrations, and no source code examples. Student should explain the principle of all models independently of any implementation environment and only then explain peculiarities for the Android environment. The class diagrams for the MVP and MVVM models are dedicated for Android, they are almost identical and thus almost useless to explain difference of these two model. In addition they contain the name collision of nodes "Presenter". Particularly the presentation of the most advance model MVI which should be described especially clearly and carefully is unsatisfied.

The symbolism used in diagram 1.6 is not clearly defined and the diagram 1.8 is essentially a fragment of the diagram 1.7.

It is not mentioned that the idea of non-modifiable states corresponds to a functional paradigm, which

is especially a pity, because the implementation language Kotlin strongly supports this paradigm.

There is no discussion whether the reactive and therefore non-blocking nature of MVI represents an advantage over

classic trampoline pattern, which is also non-blocking.

The final comparison of individual models is unconvincing. It is obvious that the MVI model in particular cannot be assessed by number

of lines of code, but especially in terms of the application of modern paradigms of functional and reactive programming.

### **3. Non-written part, attachments**

100 /100 (A)

The student presented that the result application is functional.

### **4. Evaluation of results, publication outputs and awards**

0 /100 (F)

NA

## **The overall evaluation**

70 /100 (C)

The final grade C is mainly given by objections about the textual part of the project.

## **Instructions**

### **Fulfillment of the assignment**

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

### **Main written part**

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 26/2017, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

### **Non-written part, attachments**

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

### **Evaluation of results, publication outputs and awards**

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

### **The overall evaluation**

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.