Semestral project specification

To obtain credits for this course, students are required to deliver a working Android mobile application as a semestral project, developed using Native Android or Kotlin Multiplatform (KMP).

Timeline

October 1 – December 1: Forming project groups (1-3 students) and creating project specifications

December 1: The project specification has to be accepted by a lecturer

February 28: Final version of the project

April 15: Issues identified by lecturers fixed

Project specification (Due by December 1)

Send your specification via email to:

- jan.kofron@d3s.mff.cuni.cz and
- kurnavova.simona+mff@gmail.com

Your email should include:

- **Team information** - Names of 1 - 3 members of the group. Groups of 2-3 people are encouraged.
- **Purpose of the application** - What does the app do?
- **Feature list and functionality description** - What screens will it have? What will the user be able to do?
- (*Optional*) Technical stack & implementation details - Libraries used, architecture patterns, etc.
- (*Optional*) Feel free to use this as an opportunity for consultations

Project specification criteria

To be accepted, your project must:

1. Contain at least 2 distinct screens (UI) with navigation between screens
2. Include network communication (e.g. API calls) **OR** local data persistence (or similar)
3. Have an appropriate scope and complexity for your team size

Technical requirements

Technology: Native Android **or** Kotlin multiplatform for Android + iOS
Programming language: Kotlin (Java projects are not accepted as it is deprecated language for Android development).
UI of the application: Preferred UI framework is Jetpack Compose. XML layouts will be accepted, but are discouraged.
Network communication: Retrofit or Ktor are preferred..
Data Persistence: Room database or Data Store should be used (Shared preferences are discouraged). In certain cases persistence can be done via saving data to the file.
MVVM, MVI: App follows one of the patterns - *Model-View-ViewModel* or *Model-View-Intent*.

Project submission and corrections

Please submit your finished project via email using GitHub, GitLab or Bitbucket repository link. Ensure access is granted to lecturers or repositories are public.

Before submitting the project, we recommend consulting “Best practices” section in last year’s lecture presentation:
https://github.com/avast/android-lectures/blob/2024/Lecture%206/Lecture_6.pdf

Review and corrections

After the project is handed over, it will be reviewed by the lecturers. The review will take the form of a **code review** – you will receive a list of identified issues to fix. This will be an **iterative process**, so there may be several rounds of back-and-forth. Once all identified issues are resolved and accepted, you will be granted credits for the course.

❗ *Please ensure that you dedicate sufficient time and effort to addressing these issues.*

AI Guidelines

AI is allowed to be used for generating small code snippets (not large portions of the project or the entire project), for consultation, help with bug fixing, generating unit tests, images, icons, and similar tasks. Students must fully understand the codebase they are handing over, and the code must be correct and follow the best practices discussed during lessons.