

Project Instructions

Before you start working on the project work you should complete and present the laboratory work.

As project work, you should create a nonelementary Android application for smartphones & tablets by completing the instructions in this document. The instructions have been divided into different parts that should be completed in order. We recommend you read through all the parts before you start working on the project.

The PDF file `project-grading-guidelines.pdf` on Ping Pong contains some guidelines for how your work will be graded. We recommend you read through it every now and then to double check that you fulfill the requirements for the grade you are aiming for.

In your work you should also write a report explaining how your application is implemented. The report will be a living document throughout the course, meaning that you will continuously improve and add more content to it as the course proceeds. The Word file `project-report-template.docx` on Ping Pong contains more information about what the report should cover.

Table of Contents

Part 1: Joining a Project Group	2
Part 2: Writing the Application Idea	3
Suggestions	3
Requirements.....	3
Part 3: Creating Mockups.....	5
Part 4: Implementing the Application.....	6
Part 5: Demonstrating the Application	7
Part 6: Submitting your Work	8

Part 1: Joining a Project Group

The project work should be carried out in groups of 2-3 students each. Even if you're just two students in the group you are expected to work just as much as three students. Let the examiner know who you are working with by joining one of the project groups on Ping Pong.

If you can't find someone to work with, or if you're a pair and want to find a third student to work with, send the course coordinator an email at Peter.Larsson-Green@ju.se with the following information:

- Your name(s)
- Your JU email(s)
- The grade you are aiming for in this course
- The grade(s) you got in the following pre-requisites courses (or similar):
 - Introduction to Programming
 - Data Structures and Algorithms
 - Object-Oriented Programming
 - Object-Oriented Software Development **OR** Object-Oriented Software Development with Design Patterns

The course coordinator will then try to match you up with students with similar ambitions and skills as you.

Part 2: Writing the Application Idea

The next thing you should do is to decide what the application you want to create should do. Try to be creative and come up with an application someone wants to use, i.e. an application that solves a real-world problem of some kind. You could for example create an application for an organization you are member of (e.g. a football organization that needs to keep track of members, leaders, teams, training sessions, etc.), or simply an application you would like to use yourself. You can also try asking your family and friends if they need an Android application with some functionality that doesn't exist today.

Another way to come up with an application is to think of something you find boring/dangerous/time consuming, and then try to make it fun/safe/avoid it completely using an Android application. For example, most people find it boring to wait in line. Maybe you can create an application that makes it fun to wait in line? Or an application that entirely eliminates the need to wait in line?

Suggestions

The PDF file `project-idea-flea-market.pdf` on Ping Pong describes a real-world problem you can try to solve by creating an Android application. It can also be used as a guideline for how "complicated"/"complex" the application you create need to be.

Another idea is to create an application that can control an embedded system of some kind, for example using Bluetooth. Feel free to create your own embedded system. It can for example be a simple prototype used as a proof of concept. For example, your application might be about controlling a smart lock on a door (a lock that can be locked/unlocked with a smartphone using Bluetooth), but you don't have any smart lock, so then you create your own embedded system that simulates one, e.g. one that consists of a Bluetooth module and a LED lamp that is turned on/off to indicate if the door is locked or not.

Note: You will not get any credits for learning things not related to the intended learning outcomes in this course. To build an embedded system is out of scope for this course, so only learn and do that if you are prepared to spend your own time on learning it.

If you need some special equipment (such as Beacons, modules for an embedded system, some smart devices, etc.), the school might be able to provide you with it (must be returned to the school at the end of the course). Email your needs to the course coordinator at Peter.Larsson-Green@ju.se and he will look into it and get back to you.

Requirements

The application you create must make use of at least one other device in addition to the device your application is running on. This could be:

- Any other device you can communicate with using Bluetooth.
- Bluetooth beacons.
- Another Android device running the same application, e.g. using Bluetooth to sync data.
- A server on the Internet, such as:
 - The backend application you implemented in the course *Internet and Cloud Services* (for those of you who took that course).
 - A [Firestore](#) database (or any Backend-as-a-Service you want).

The application may also not be too simple. For example, creating the simplest possible all-to-all chat application synchronizing chat messages using Firestore will not be accepted. But often ideas like this one can quite easily be made more complex. In this case one could also add support for one-to-one chats, group chats, friends chat, etc.

Describe your application by starting to write the project report. Especially, visualize which components your solution consists of (android device, running your application + at least one other device) and how they make use of each other. You should especially write as much as possible for now in the Introduction chapter and the Overview chapter.

Note that your report is a living document throughout the course. What you write now will most likely be changed later when you come up with idea for improvements, so do not spend too much on details now. Just write it detailed enough so the reader gets a rough understanding of everything, e.g. start out simple. **But write it properly!**

Part 3: Creating Mockups

Create mockups of the graphical user interface. When you are done, show your mockups to a lab assistant at one of the lab sessions. If the lab assistant doesn't understand how your application will work in the end, they are not good enough and need to be improved.

When your mockups are good enough for the lab assistant to understand how the application will work in the end, go to the examiner's Doodle MeetMe page at <https://doodle.com/peterlg> and suggest **multiple** different **15 minutes** slots (do not suggest 30 minutes slots, or similar) you can meet him. The examiner will then get back to you once he has selected one of the slots. At that time, you should come to the examiner's office E3408 and present what you have written in the report so far, and what your mockups look like. You only have 15 minutes to do this, so be sure to come prepared and present it in an efficient way. Before the meeting, submit the assignment *Project Work* → *Application Idea* on Ping Pong with the current version of your report and your mockups.

The examiner will either approve or disapprove your application idea. If he approves it, he will approve you on the assignment *Project Work* → *Application Idea* on Ping Pong. If he disapproves it, you need to come back another time (use the examiner's Doodle MeetMe page again) after you have changed it according to the examiner's feedback. If you haven't been approved on it after the second meeting, you can still pass the project work, but the examiner will not spend any more time on trying to estimate if the application you create will be advanced enough for this course until you submit the final version of it at the end of the course.

Note: It may take some time before the examiner has time to see you. If you want, you can start implementing the application (continue on next part of the project work) before the examiner has approved your application idea. If he disapproves it, it is most likely because your application will be a bit too simple to implement, so the changes you need to do will most likely be to add (and not change existing) functionality.

Part 4: Implementing the Application

Start to implement your application using Android Studio. We recommend you to version control your code using a version control system, such as Git. Using a version control system is not a requirement for the project work, but it will make it much easier for you to share the code with each other. Use private repositories if possible. gitlab.com offers free private repositories you can use.

The first Android lectures will primarily focus on how to implement the graphical user interface in Android applications, so you are recommended to start by implementing that part of your application. Learning how to properly implement the GUI is also the most difficult part of learning Android.

Initially, see it as you are creating a prototype of your application. For example, if a user should be able to login to your application (which could be the case if you use Firebase), you need to have text boxes where the user can enter her credentials (e.g. its username and password) and a Login-button, but for the prototype you do not have to actually login the user when she presses the Login-button (e.g. you don't need to use Firebase at all). Instead, hardcode the correct credentials for a specific user in the prototype, and verify the entered credentials directly in your application. Or if you need to fetch data from a REST API, hardcode the data directly in the Android application. Or if you need to communicate with an embedded system using Bluetooth, emulate the communication with the embedded system without using Bluetooth at all.

As more Android features are covered at the lectures in the course, start to use them in your Android application, so your prototype application slowly turns into the final version of your application. For example, the first version of your prototype might not care about runtime configuration changes at all, but after that has been covered in the course, improve your activities by starting to handle them. Or maybe some of your code should run in a service instead of an activity? Then you can initially put that code in an activity, and then refactor it out and put it in a service when those have been covered in the course.

We recommend you to not start too advanced. To pass the course, it is more important that the functionality you implement works properly than that the graphical user interface looks beautiful. Therefore, it is probably better to start with a simple GUI and then improve it at the end of the course if you have time for it.

If you want to get grade 4 or 5, you must come and see the examiner for another meeting to discuss your progress. Your work will not be graded, but this is an extra chance for you to get some feedback on your work before the final submission at the end of the course. If you are aiming for grade 3 you are welcome to book one of these meetings as well, but it is only mandatory for grade 4 and 5. The meeting should take place after week 8/after 24th February. Use the examiners Doodle MeetMe page again the same way as before. After the meeting, the examiner will approve you on the assignment *Project Work* → *Halfway Meeting*.

Part 5: Demonstrating the Application

Your application should be demonstrated to the rest of the class (in smaller groups) at the end of course (time to be decided). At the demonstration, you should:

- Explain what your application can do
- Explain why someone would use it
- Show how users will use it

You may structure your demonstration however you want, but a recommendation is to start by explaining what the application is used for, and then have a quick demonstration. The demonstration is preferably carried out by having one person interacting with the application (shown on a big screen), and another one explaining what the first person is doing/telling the first person what to do.

The demonstration should take at most 15 minutes. There should be a short time for questions from the audience at the end (included in the 15 minutes). Plan and rehearse your demonstration in advance, so no time is wasted on *"Oh, what can I show you next? Hmm..."* or *"Oh, that should not have happened. It has never happened before. Hmm..."*.

The reasons you should demonstrate your application to the rest of the class are:

- To learn what type of Android applications you can create (you have all taken the same course, so you should all be able to create the applications the other students have created).
- To practice your presentation technique. This is not limited to improving your speaking skills, but also includes improving your skills to explain how something works to others, and to prepare and carry out demonstrations (your future customers will appreciate such skills).

Join a Project Demonstration Groups on Ping Pong to let the examiner know which project demonstration session you want to attend to.

After you have demonstrated the application the teacher will mark you as approved on the assignment *Project Work* → *Demonstration* on Ping Pong.

Part 6: Submitting your Work

When you are done with your project work, you should upload it to Ping Pong for grading by submitting the assignment *Project work* → *Final submission*. When you submit the assignment, you should attach:

- The source code for the application as a ZIP file.
- The application as an APK file.
- The report as a PDF file.

The deadline for doing this is Sunday 24th March 2019.