

# Welcome & Organization

On this site you will find some **general information**, a **script** and the **exercises** for the lab of the lecture **Human Computer Interaction (HCI)** of Prof. Dr. Ute Trapp at Hochschule Darmstadt, Germany. The workload of this module for a typical efficient working student with the corresponding pre knowledge is 5 ECTS, i.e. ~115h (about 3 weeks).

There are video playlists addressing flutter

- [Flutter tutorial with Riverpod and GoRouter](#)
- [Flutter-Tutorial \(deutsch\) mit Riverpod und GoRouter](#)

## Objective of the Lab

In this lab you will ideate, validate, plan and develop your own mobile app. The focus of the lab is not on the best software architecture. However, we expect you to apply your knowledge of **OOAD** and the parallel course **SWE** -- [here](#), you find the lecture slides of both modules. We focus on the **users perspective**.

You will go through the typical user centered development cycle, reflect your work and improve your skills to work in a team. During the semester you will develop a web and mobile app using **Flutter**. However, the programming part accounts for a little less than 50% of the total grade. The implementation itself will be done in two labs only, the three other labs address **user experience** issues. But a new framework and programming language need more than two labs to learn it, hence you will work on little tasks on your own **before** you start to implement your project.

## Project

You are free to choose a project of your interest. It is only requested that you address a real problem for users you have access to. Alternatively, you may develop a serious game. After the first lab it is not possible to change the project idea. You may also extend an existing app. If you choose to do so, make sure the existing project is still in active development and the license covers your involvement. You should add features with UI, design and implementation parts. Here are some projects you may have a look at

- projects using riverpod for state management (as preferred in this lab)
  - [movie app](#)
  - [time tracking](#)
  - [feel - workout app](#)
- projects using provider (not riverpod)
  - [open food facts app](#) -- complex, very active
  - [wonderous](#)
  - [smart home](#)
- projects using bloc, another widely used state management library
  - [ai based card game](#)
  - [ToDo-App](#)

Or search by yourself or check [FDRÖID flutter apps](#) or have a look at one of the following lists

- [Open Source projects in Flutter](#)
- [Top 12 Flutter Open source projects](#)

#### Tip

Here are some hints of the students of the previous semester

- früher anfangen, Flutter zu lernen
- Eigeninitiative zeigen
- Termine mit Team vereinbaren
- nicht zu komplizierte Projektidee wählen
- offenes Labor besuchen
- stellt früh Fragen, dann kann man frühzeitig Fehler beheben
- einigt euch im Team, keine Einzelgänge
- erklärt euch den Code im Team
- niemals aufgeben bei Praktika
- es ist aufwändig und teilweise komplex, aber es lohnt sich

## Backend

The development of a backend is not part of this course, hence do not invest time to do so. However, there are great tools and available APIs to use in your project.

- [API generator](#): Tool used in the videos and the [repository](#) provided. With the tool you may define one table with different columns, get a permanent URL and modify the data with REST-calls. [Low-code backend](#) seems to be similar, but I did not test it.
- [bund - administration](#): APIs provided by the administration of Germany, e.g. waste management.
- [Google Sheets API](#) and [Build a CRUD API using the Google Sheets API](#)
- [Fake store rest API for your e-commerce or shopping website prototype](#)
- [OpenFoodFacts](#)
- [Public APIs Developers Can Use in Their Projects](#)
- [Big List of Free and Open Public APIs \(No Auth Needed\)](#)

Flutter plays well with [Firebase](#) and many packages makes it easy to access firebase in flutter applications. If you are familiar with firebase. Feel free to use it as your backend.

## Lecture and theory

For the lecture you need to come prepared (see Moodle). Sometimes I will give a short input. Most of the time, we discuss special issues and work either on the lab itself or on preparing tasks for the lab. I will do some live coding as

well. Although you will find a script in the [theory](#) part of this site, I expect you to come to the lectures and fill the gaps and to discuss different approaches. As cognitive learning requires reflection and discussion.

## Topics of the Lab sessions

The lab sessions will cover the following topics

- [Lab 1: Teams & Problem Space](#)
- [Lab 2: User Research](#)
- [Lab 3: Prototypes & Design](#)
- [Lab 4: Architecture & Implementation GUI](#)
- [Lab 5: Implementation MVP](#)
- [Lab 6: UX Evaluation / Presentation](#)

## Carrying out the Lab

You are expected to use your private notebook and a gitlab repository, which you will get during the first lab session (lab 1). The provided repository is based on [learning-flutter](#).

## Preparation and Follow-up

In each lab description you will find a preparation. It is essential that you have dealt extensively with that preparation **before** the lab date. You will notice that the tasks are formulated very openly. It's up to you how deeply you want to understand the topics. As usual, understanding comes with using, reading is not sufficient, reading is just the start;)

## Feedback

We are happy to give you feedback

- 3 days after your lab (a tutor will write a gitlab issue)
- at the beginning of the following lab for the documentation of the last lab

## Teamwork

Each of you will work on a project in groups of four or six at most. It is recommended to address either iOS or Android, not necessarily both -- web is mandatory. As in reality, it is up to you to distribute the workload fairly. However, if one or more do not contribute as planned, do not hesitate to contact me. Do so as early as possible -- and not after the grading. **Work together as a team. Be reliable and fair to each other, and communicate.**

## Communication

For general questions use the Moodle Forum. For individual questions use the lab, lecture and on demand consultation hour.

**Do not hesitate to communicate with the lecturer in case of any problems or questions. A lively communication is explicitly encouraged.**

## Thanks

The source code and infrastructure of the page as well as some content is based on the material of Ralf Hahn, Stefan Zander, Stefan Ruehl and others I am not aware of.

---

This website was created using [MkDocs](#), [Markdown](#) und [Visual Studio Code](#) and rendered with [Material Theme for MkDocs](#).