



Group Project

AMIV UX: From API to UI

Recent advances in the *application programming interface (API)* of the AMIV [1] significantly facilitated the implementation of common use-cases which the AMIV faces in their day to day business. Even though the majority of people working at AMIV have a tendency to favor the lower levels of the application stack (e.g., hardware, or hardware near protocols), using the API directly for a longer period of time turned out to be rather cumbersome. It has therefore been suggested [2] that a *user interface (UI)* should be developed that increases the *user experience (UX)*.

It has been noted on many occasions, that a simply having a user interface does not mean that said user interface satisfies users. Thus, the term user experience was coined, which addresses common issues, and focuses on a satisfying user experience, rather than exposing the underlying mechanisms in a 1 : 1 fashion. The difference between UI and UX is illustrated in Figure 1.

We therefore claim that it is time to address these issues, and to develop a new, user-friendly user interface, which works on top of the existing AMIV REST API.

The goal of this project is to analyze the functional requirements of the AMIV front-end, to select appropriate tools for development, and also for documentation, and to implement the required features.

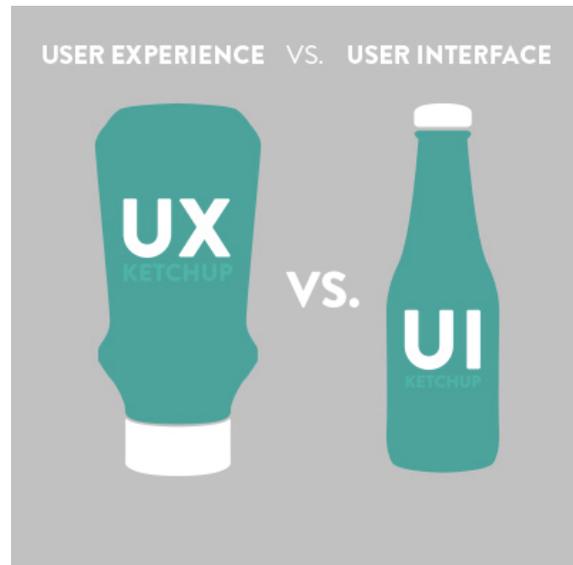


Figure 1: Simply allowing a user to interact with a service does not automatically lead to a good user experience.

[1] <http://www.amiv.ethz.ch>

[2] A RESTful API for the AMIV. *H. Blum, C. Buchert, A. Dietmiller (2015).*

Contacts

- Roger Wattenhofer: roger.wattenhofer@tik.ee.ethz.ch, ETZ G96
- David Stolz: david.stolz@tik.ee.ethz.ch, ETZ G94
- Laura Peer: laura.peer@tik.ee.ethz.ch, ETZ G97

Detailed Project Outline

In the following, we outline the tasks of the students.

- Analyze the functional requirements of the frontend, and categorize them.
- Search and evaluate tools for successful developments, including: Project planning, project documentation.
- Search and choose an appropriate development environment, including: Programming language, libraries, . . .
- Design a responsive user interface.
- Implement the required functionality.
- Test the required functionality.

The Students' Duties

- One meeting per week with the advisors.
- A final presentation (15 min) of the work and results obtained.
- A final report (English or German), presenting work and results.