

SEBA Master: Web Application Engineering

0. Introduction

Prof. Dr. Florian Matthes, SS21

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
www.matthes.in.tum.de

After successful participation in this lecture, you can

- identify, explain and develop B2B & B2C **business models** on the web
 - select and apply **patterns** for web application design
 - explain a **user-centered, iterative development process** and apply it in a team
 - explain and apply concepts of **modern web technologies**
 - **present** software projects and software deliverables to an expert audience
-
- **apply** concepts and technologies of a **JavaScript based web application development framework** to build a small web solution
 - **develop** small web solution based on the **MERN** stack (**M**ongoDB, **E**xpress JS, **R**eact JS, **N**ode JS)
 - have sufficient knowledge to contribute to the development of projects in this area.

Outline of the Lecture (1)

1. Web Site Genres

- Business Models on the Web
- Web 2.0 and Social Software
- Anything as a Service (XaaS)

2. Web Site Design Process

- Key Issues of Customer-Centered Web Design
- An Overview of Web Design Patterns
- Knowing Your Customers
- Involving Customers with Iterative Design
- Processes for Developing Customer-Centered Sites
- Additional Viewpoints and Aspects

Outline of the Lecture (2)

3. Patterns for Web Site Design

- Explaining your Value Proposition
- Creating a Navigation Framework
- Basic E-Commerce
- Advanced E-Commerce
- Helping Customers Complete Tasks
- Designing Effective Page Layouts
- Making Site Search Fast and Effective
- Making Navigation Easy
- Speeding Up Web Sites
- Supporting a Wide Range of Devices (Responsive design)
- Integrating Multiple Touchpoints
(Mobile apps, progressive web apps, and hybrid apps)
- Integrating Independently Developed Web Services,
Content, and Processes

- We will **skip** this section in this to align the lecture better to the practical part
- However, these patterns are **relevant** for the design of your web applications
- We will upload the slide deck on Moodle as additional material

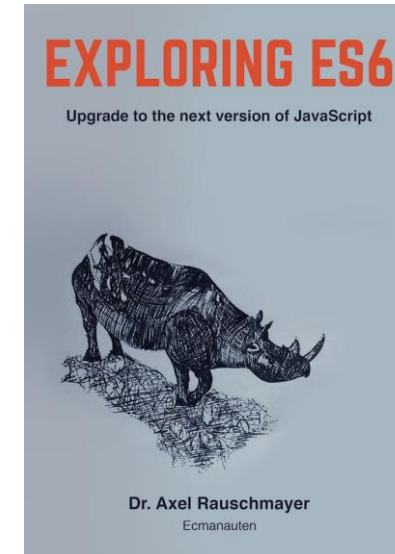
"The Design of Sites. Patterns for Creating Winning Websites" - Douglas K. van Duyne, James A. Landay, Jason I. Hong, Prentice Hall; 2nd edition, 2006



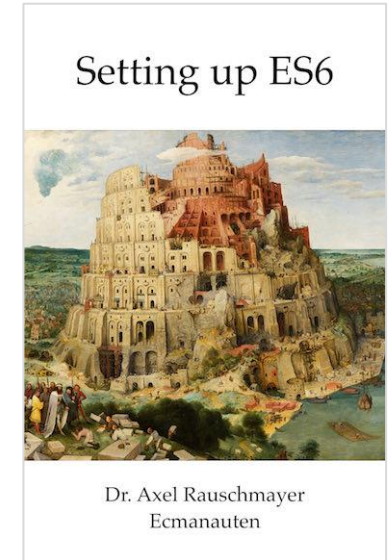
Outline of the Lecture (3)

4. Developing Serverless Single-Page Web Applications

- HTML Concepts and Evolution
- CSS Concepts and their Link to HTML
- JavaScript Basics
- SVG and Canvas
- Web Components
- Single-Page Web Applications
- Component-based Frameworks



<http://exploringjs.com/es6/>



<https://leanpub.com/setting-up-es6/>

5. Building Single-Page Web Applications with a Specific Web Application Framework

- Web Application Framework Comparison
- Basic Concepts of the Chosen Framework
- State Management
- Navigation and Routing
- Client-Server Communication
- Consuming RESTful Services

Outline of the Lecture (4)

6. Building REST-Enabled Backend Services

- Target Architecture and Development Environment
- Event-driven Architecture and Asynchronous I/O operations With Node.js
- Creating REST Interfaces
- Using Document-oriented Database Storage
- Enabling User Authentication on the Web Service

Guest Lecture

- TBD

7. Advanced Topics in Web Application Engineering

- Real-time Web Applications
- Virtual DOM
- Hybrid Web Applications
- Other Web Frameworks
- Traditional Web Stacks, Server-Side Rendering

8. High Performance Web Applications

- Database Options for Web Applications
- Scaling Web Applications

Additional Reading (Online & Print)

- “Business Model Generation: A Handbook for Visionaries, Game Changes, and Challenges” - Alexander Osterwalder, Yves Pigneur, John Wiley & Sons, 2010
- „The Back of the Napkin: Solving Problems and Selling Ideas with Pictures“ – D. Roam, Portfolio Hardcover, 2012
- “Don't make me think! Web Usability: Das intuitive Web” - Steve Krug, New Riders Press; 2 edition, 2005
- “Designing the Obvious. A Commonsense Approach to Web Application Design” - Robert Hoekman, New Riders Press; 1 edition, 2006
- “Building Scalable Web Sites” – Cal Henderson, O'Reilly; 2006

- “Exploring ES6” – Axel Rauschmayer, Ecmanauten, 2018 <http://exploringjs.com/es6/>
- “Setting Up ES6” – Axel Rauschmayer, Ecmanauten, 2018 <https://leanpub.com/setting-up-es6/>
- „Express – Web Framework for Node.js“ – <http://expressjs.com/>
- „React Official Tutorial “ – ReactJS.org, <https://reactjs.org/tutorial/tutorial.html>
- „React.js Essential Training“ – Eve Porcello, Lynda.com, 2017, <https://www.lynda.com/React-js-tutorials/React-js-Essential-Training/496905-2.html> (free access for TUM students)

- **Module number:** IN2087
- **ECTS:** 8 Credits
- **Self-study hours:** 180
- **Participation hours:** 60
- **Target audience**
 - Master Information Systems (compulsory course, priority)
 - Master Informatics & others (elective course)
- **Prerequisites**
 - Bachelor-level knowledge of programming, software engineering, networks and databases.
- **Lecture materials**
 - **Lecture recordings**, without personal information about students participating in the stream, will be available on [Moodle](#)
 - Slides, tutorials, and assignments will be available on [Moodle](#)
 - To get access to the course in Moodle, make sure to be registered to the lecture in [TUMonline](#)
- **Type of Assessment: Project Work**
 - Development of a web application in a team of 4 students
 - Assessment based on presentations, (written) deliverables, team work, etc., and answers to questions to topics presented in the lecture in form of...
 - **a final written exam (45 minutes)**

- **Lecture Format:**
 - The lecture is divided into a theoretical and a practical (tutorial with demo) part. The practical lessons are starting on **4th of May**.
 - **Theoretical part: Mondays 14:15 – 15:45** on <https://www.twitch.tv/tumsebis>
 - **Practical part: certain Tuesdays 12:15 – 13:45** on <https://www.twitch.tv/tumsebis>
 - See lecture dates: <https://www.matthes.in.tum.de/pages/1mqqqoqe7gapz/SEBA-Master-Web-Application-Engineering>
- **Submission of deliverables and group presentations at the end of the following phases. (Each phase grade contributes to the final grade.)**
 1. Business idea, BMC, and VPC (Presentations: 05./06./07.05.2021)
 2. Use-cases, mockups, and data model (Presentations: 19./20./21.05.2021)
 3. Final Prototype (Presentations: 27./28./29./30.07.2021)
 4. Written exam (TBD)
- **Special guest who talks about web application engineering in the industry – 21.06.2021 (not final, TBD)**
 - TBD
- **Two optional prototype reviews conducted by your mentors to give you feedback and provide guidance. Please contact your mentor for scheduling an appointment.**
 1. First prototype review period (between 14.06.2021 and 18.06.2021)
 2. Second prototype review period (between 05.07.2021 and 09.07.2021)

- **The registration for the lecture ends today (23:59)!**
 - Until then, you can only register for the **waiting list**.
 - Places are limited to 220 due to capacity constraints.
 - **Master Information Systems** students get a fixed place first.
 - If you start your master's program in Information Systems (Wirtschaftsinformatik) in the summer semester 2021, but you are not yet officially registered as a master student in Information Systems on TUMonline, please send an email with an official **notice of admission (Zulassungsbescheinigung)** to nektarios.machner@tum.de until 15.04.2021 to receive a fixed place.
 - On 16.04.2021, all remaining **fixed places** will be assigned.
- In the first week of the lecture, students **with fixed places** will be **assigned to teams** by the organizer.
- In case you want to form **teams on your own**, please write an email containing the **names** and **matriculation numbers** of all team members to the organizer (Nektarios Machner)
 - **Subject** of the mail: *SEBA Master 21 – team proposal*
 - **Deadline** for team proposals: 15.04.2021 at 23:59
 - Only **one mail per team** (put team members in **CC**)

- **Each team** will be assigned to an **advisor** (employees/research assistants of the sebis chair).
- **Advisors** will provide support by advising you throughout the course. Feel free to contact them.
- **However:**
 1. They are not your "tech support". Please try to solve technical problems within the team and only reach out to the advisors regarding such issues if you can't solve the problem on your own.
 2. Don't ask the advisors to grade your submissions in advance. If you want feedback on your submission before the submission deadline, please contact your mentor **early**.
Example: Deadline is on Sunday 23:59 → Contact advisor on Wednesday, not on Sunday 2 hours before the deadline.
- When contacting your advisor **via mail**, please mention your **team number** and put the remaining **team members in cc**.

Finding Your Idea

Call for sophisticated project ideas by experienced students

If you already have some **experience** in web application development, do not hesitate to approach more **sophisticated topics**, e.g., by implementing/integrating

- Real-time collaboration
- Interactive visualizations and maps
- Interactive video players & augmented reality
- Machine learning, natural language processing, etc.
- ...

Successfully implementing **ambitious** and **innovative** web applications can

- have a positive impact on the final grading of your project (we would ignore certain technical issues if the innovative core functionality is implemented satisfactorily)
- potentially lead to interesting follow-up projects (e.g., master's thesis, cooperation with industry partners, etc.)
- be an interesting challenge for you

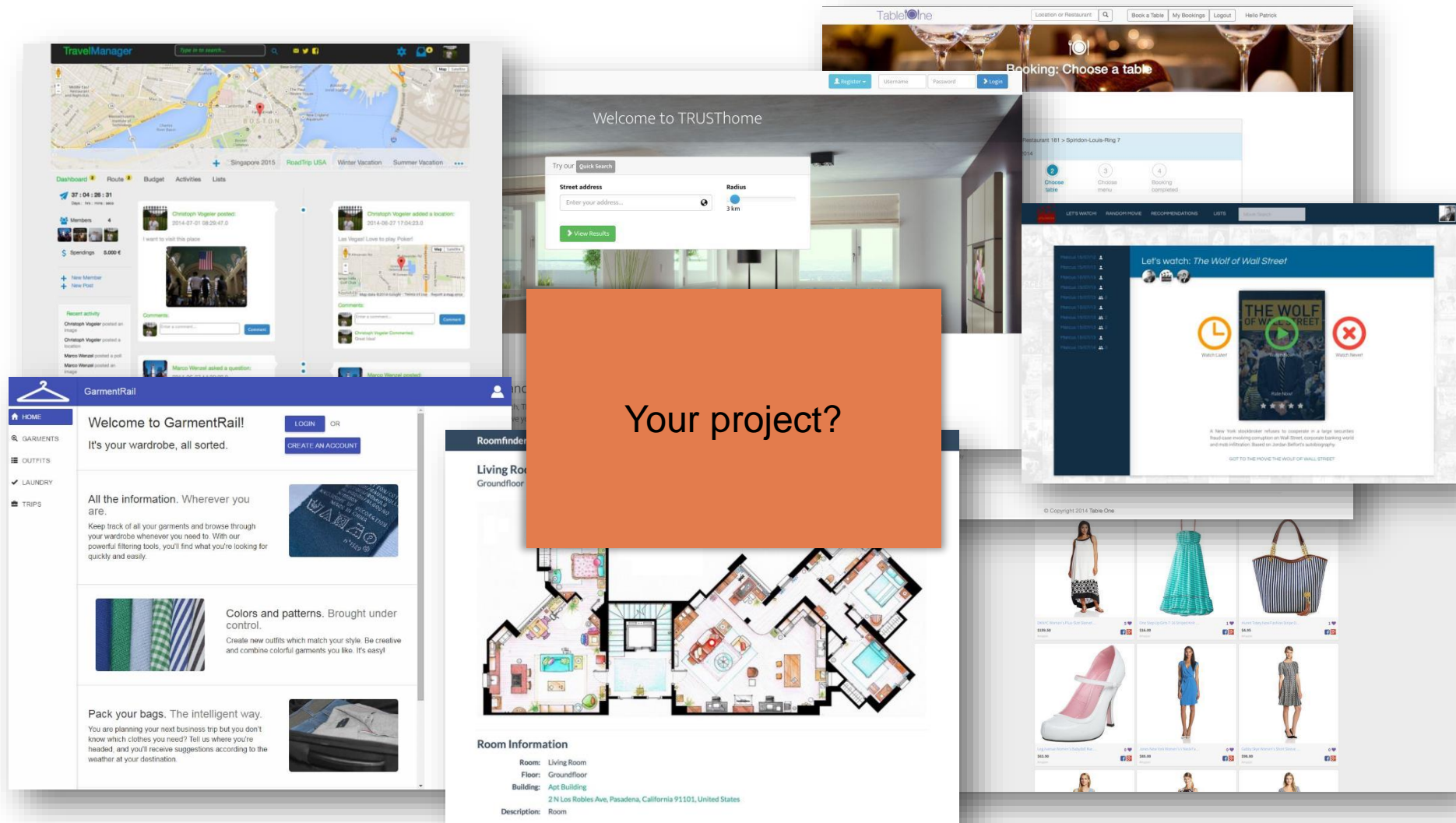
Currently, there are some **related research projects** at the sebis chair to which your idea could contribute, e.g., [LegalTech](#), [Next-Generation IT Organizations](#), [Service Platform and Ecosystems](#), [Blockchain-based Systems Engineering](#), and more. (Check the sebis page for further details.)

However, **do not overreach yourself**, especially if you do not have any experience in web application development.

Finding Your Idea

Projects from previous years

Over the last few years, we have published a selection of great students projects on [our website](#)



Details on the Overall Assessment

The final grade consists of **five partial assessments**:

- Over the course of the semester, we will have **three assignments**
 - **A1** – Business idea evaluation: Business idea, BMC, and VPC (10%)
 - **A2** – Technical concept: Use-cases, mockups, and data model (10%)
 - **A3** – Final prototype (50%)
- Individual assessment regarding **presentation skills, rhetoric**, and performance in **discussion** (10%)
- **Written final examination** regarding lecture content (20%)

Assessment	Contribution	Example
A1 – Business Idea Evaluation	10%	1.7
A2 – Technical Concept	10%	1.3
A3 – Final Prototype	50%	1.0
Individual presentation and discussion skills	10%	1.3
Written final exam (45 min)	20%	2.3
	Final Grade	1.3

- **However:** In each assignment, we assess the individual contribution of each team member. Should a team member not contribute to the group work, we will assign that team member an overall final grade of 5.0, which results in failing the course regardless of potentially already acquired partial assessments of ≥ 4.0 .

The SEBA Movie Web Application

- We are providing a working example web application, the SEBA Movie App.
- The application is available on Github.
 - <https://github.com/sebischair/sebamastrer-movie-backend>
 - <https://github.com/sebischair/sebamastrer-movie-frontend>
- ReactJS Frontend
- Node.js/Express Backend
- Please try to get the application running on your machine according to the instructions in the README files until 04.05.2021.

On 04.05.21 12:15-13:45, we will have a Q&A session on Twitch!
If you have questions or problems regarding the setup of the example app,
join this Q&A session.

The Microsoft Frontend Bootcamp

- Microsoft provides a free "Bootcamp" for learning ReactJS basics.
- You can use this Bootcamp as addition to the lecture.
- The Bootcamp consists of two "days".
- The first day covers the basics whereas the second day introduces more advanced (e.g. Redux) and some for our course optional (e.g., TypeScript, ...) aspects of frontend programming
- The Bootcamp is available under <https://microsoft.github.io/frontend-bootcamp/>

Structure of the Presentations

- Over the course of the semester, we will have **3 presentation blocks**.
- Each presentation block lasts **3-4 days** and has approx. 14 slots in total (**~5-6 slots per day**)
- We will assign each student team to one of these 14 slots for each presentation block. Each team must be **present only in its assigned slot**.
- In each slot there will be 4 student teams, each of them having **max. 20 min for presentation and discussion**.
- This semester, the presentations will be performed **via Zoom**.
 - Each team of a certain slot will receive an invitation to a Zoom meeting session.
 - The **whole team joins the meeting at the start time and stays until the end** of the last presentation in that session.
 - During the discussion of each presentation, the remaining three teams can ask questions as well.
 - The schedule for the 3 presentation blocks will be published roughly a week before the respective block takes place.

Each of the three assignment sheets is structured as follows:

SEBA Master - Web Application Engineering
Summer term 2020



Exercise 1

Motivation of the exercise

1. Description of the Exercise

A general description of the content and scope of the exercise

2. Presentation Session

Detailed description of what you have to present in the presentation session

3. Deliverables

Detailed description of what are the deliverables which have to be submitted before the presentation

4. Submission of Deliverables

Instructions how to submit the deliverables

5. Evaluation

Description of how the presentation and the deliverables will be assessed, and how much points the exercise is worth (Total number of points for the whole lecture: 100)

6. Outlook

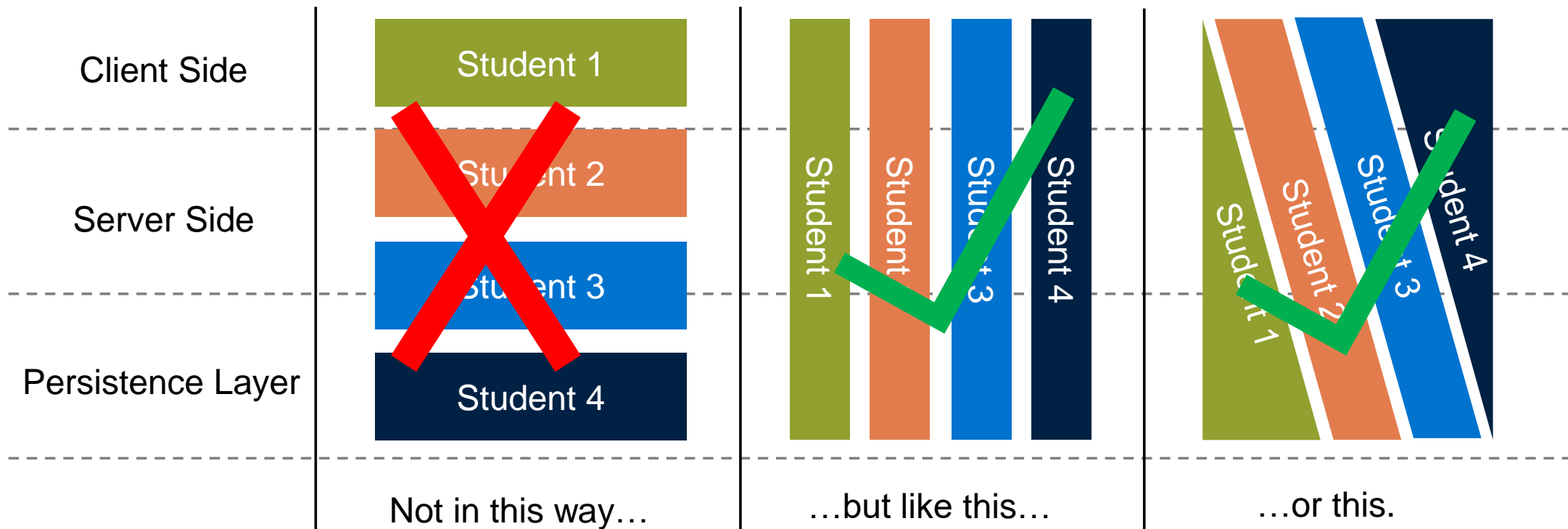
A brief outlook of what the next exercises are about, and how the current one forms a foundation for the future tasks

1. General **description** of what the content and the goal of the assignment is
2. Detailed description of what the **presentation** must cover
3. Detailed description of what the **deliverables** which must be submitted before the presentation are
4. Description of how to **submit** the deliverables
5. Explanation of how the presentation and the deliverables will be **evaluated**
6. An **Outlook** of what the next assignment will be about, and how the current one is the foundation for the future activities

- We will have **one final exam of 45 minutes**.
- We will publish an example of such an exam on Moodle during the semester.
- Date: **TBD**
- Format needs to be defined depending on the situational circumstances, most likely online with supervision and on-site alternative.
- The written exam counts 20% towards the overall final grade.

Distribution of Workload

- Distribute the workload of each assignment in a way that each student is familiar with each part of the deliverable, not only with those he or she is responsible for.
 - E.g., for the implementation of the prototype, do not distribute the workload horizontally along the abstraction layers, but vertically along the use cases.
 - In the discussion we might ask any student about any part of the deliverable



Technology Stack

