

Exercise 3

The goal of this exercise is to start the development of your web application and to present your initial prototype.

1. Description of the Exercise

Implement an initial prototype of your web application based on the MEAN-Stack consisting of:

- MongoDB (<https://www.mongodb.com>) as a database
- Express.js (<http://expressjs.com/>) as a web framework
- AngularJS (<https://angularjs.org/>, we recommend version 1.x) as a client-side framework
- Node.js (<https://nodejs.org/>) as a runtime environment for JavaScript

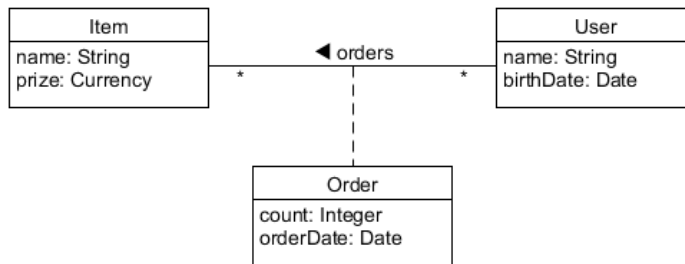
Furthermore, you should use a UI framework like Twitter Bootstrap (<http://getbootstrap.com/>), Semantic UI (<http://semantic-ui.com/>), or Angular Material (<https://material.angularjs.org/latest/>) for designing your web site. Basics and key concepts of those technologies are presented in the lecture. However, if you are not yet familiar with them, we highly recommend to do the official tutorials, e.g., <http://expressjs.com/en/starter/basic-routing.html>, or <https://docs.angularjs.org/tutorial>. We recommend to use Webstorm by JetBrains as an IDE which is free of charge for students and for academic use (<https://www.jetbrains.com/student/>). To enable collaborative development of your web application, we highly recommend to use a version control system like Git (<https://git-scm.com/>) and to host your remote repository on Bitbucket (<http://bitbucket.org/>) or GitHub (<https://github.com/>).

You can use the MovieApp as presented in our introduction talks to Node.js and AngularJS as an inspiration and starting point. We've published the backend (<https://bitbucket.org/sebischair/sebamaster-movie-backend>) as well as the frontend (<https://bitbucket.org/sebischair/sebamaster-movie-frontend>) project on Bitbucket. Those repositories also contain some basic information on how to set up your environment.

We would like to encourage you to follow a component-based approach to your web application's architecture, i.e., do not separate your code by layers (model, controllers, routes, etc.), but by semantic components. Both the backend and the frontend part of the exemplary MovieApp implement and demonstrate a component-based architecture. Also have a look at the slides of the introduction talks to Node.js and AngularJS which we uploaded/will upload to Moodle.

On a general note and as presented in the first lecture: Do not distribute the workload strictly along the layers of your web application (e.g., UI, application logic, database, etc.), but in a way that each team member gets familiar with each layer of the web application. This includes models, controllers and views on both the client- and server-side part of the web application.

Of course, the UI of the web application should comply with the mock-ups of exercise 2. Deviations have to be justified accordingly. Furthermore, thoroughly think about the conceptual data model of your web application: Which are the basic entities your web application has to deal with, and how they are related to each other? For this purpose, use a UML class diagram (https://en.wikipedia.org/wiki/Class_diagram) to design such a **conceptual data model**. For example, an excerpt of the conceptual data model of a shopping platform might look as follows:



For creating UML class diagrams, you can use the free UML tool UMLet (<http://www.umlet.com/>), or the online-tool gliffy (<https://www.gliffy.com>) which provides a free account (see pricing page).

For the initial prototype, make sure that the web application

- already has a **nice-looking landing page** which gives an overview over your web application's value proposition,
- implements user **registration** and **login**, and
- supports at least **one of the business idea-specific use cases** as defined in exercise 1.

2. Presentation Session

Right before your presentation, make sure that your **presentation notebook** is **prepared**, i.e., start the web application and a proper browser, and open your slides. Also make sure that your presentation and live demo works for lower resolutions (e.g., 1280x800). We can only guarantee a VGA-connection to the beamer, i.e., take care of respective adapters by yourselves!

The structure of your presentation should be as follows:

- Short description of your business idea. (1 slide)
- Overview over the conceptual data model of your web application (1 slide)
- Short description of the implemented use case as well as the corresponding mock-up (1 slide)

Thereafter, you have to do a live demo of your prototype. Again, the demo should be based on the use cases of exercise 1.

The presentation (inclusive the live demo) must not exceed 10 minutes.

The subsequent discussion, will last additional 10 minutes at most. During the discussion, you might be asked to do a short code review, e.g., to explain certain parts of the web application's implementation. For this purpose, think of which parts of the code do you consider as the most interesting/challenging ones.

The presentation of the slides and the live demo has to be done by exactly two students. These two presenters will be assigned randomly immediately before the presentation. Therefore, make sure that each of the team members is familiar with the web application and its implementation. However, in the discussion session and thus for the code review, each team member might be asked and take part.

3. Questioning Session

20 minutes before your presentation, your team has to be in the "Question room" (01.12.034) in order give a written answer to a randomly chosen question out of a public question catalog. The questions cover the content of the lecture

until part 5 – *Implementation of Web Application Servers* (apart from the parts which were already covered by the first questioning session). There won't be any questions about chapter "3 – Web Design Patterns", which we skipped in the lecture.

The question catalog will be published on May 31th, 2016.

4. Deliverables

For this exercise, there are 4 deliverables which you have to submit:

1. A **Paper** (in PDF format) containing an **UML class diagram** of your web application's **conceptual data model** as described in Section 1. *Description of the Exercise*.
2. At least one **Screenshot** (or a series of Screenshots) of your web application (in PDF format) which shows how one of the use cases of exercise 1 is supported by your current prototype. Also add some description of which use case is covered by the screenshot.
3. The **Presentation** (in PDF format) as defined in Section 2. *Presentation Session*.
4. A **Paper** (in PDF format) describing the individual contribution to the exercise of each team member

5. Submission of Deliverables

You have to submit deliverables via e-mail with the following guidelines:

- Deliverables must be sent via e-mail in PDF format. Recipient is: seba-master.sebis@tum.de.
- The subject line must conform to the following format:
 "[SEBA-Master] Exercise 3 - Group number <Number of your group> - v<Version>".
Please replace the parts enclosed in <> (and delete thereafter the symbols '<' and '>'). Provide the group number with a leading '0' in case it is less than 10, i.e., '01' instead of '1'.
- The e-mail body must contain the following information about each group member:
 - last name,
 - first name,
 - matriculation number,
 - your project group number
- The e-mail with the highest version number before the deadline is considered as final.

Only submissions conforming to the stated form will be accepted.

Submission deadline for the third exercise is Monday, June 06th, 2016, 12:00 (noon).

6. Evaluation

The exercise is evaluated according to the following guidelines:

- Documents and slides were delivered in time (before deadline)
- Quality and UML-conformity of the submitted conceptual data model
- Quality of the implementation of the selected use case as well as conformity with the mock-ups from exercise 2 (Deviations of the UI from the mock-ups of exercise 2 have to be justified accordingly)
- Individual contribution of each team member to the exercise

SEBA Master - Web Application Engineering

Summer term 2016



Technische Universität München

- Style and structure of the document
- Presentation of the slides
 - Make sure to comply with the schedule. Overtime (exclusive interruptions due to questions) will affect your assessment negatively!
- Style of presentation
 - Presentation skills (confident appearance, fluent speech, etc.)
 - Dealing with short interruptions and questions
 - Slide design and appearance (no animations, pleasant colors, “corporate design”, etc.)
- Correctness of the submitted answer in the questioning session.

This exercise is worth **22 points**:

- 8 for the questioning session
- 2 for the presentation
- 12 for the deliverables

In total, there are 100 points over the course of the semester.

7. Outlook

As the last exercise, you have to finish the implementation of your web application, i.e., you have to implement all use cases as defined in the first exercise.



Fakultät für Informatik
Lehrstuhl für Informatik 19