**Web-Engineering 2**

Book Universe

Maxime Fritzsch, Nick Schroeder

Inhalt

[How should the solution work? 3](#_Toc87018281)

[With what should the solution be built? 4](#_Toc87018282)

[Test Concept 5](#_Toc87018283)

# How should the solution work?

## How is the system structured and constructed?

Used technologies are included in the MEAN-Stack:

* MongoDB/(MySQL)
* Express
* Angular
* Node.js

Except for the database, which should be hosted in the Cloud, everything else is supposed to run locally.

## Which interfaces and boundary conditions are there?

Interfaces:

* Google Books API

## What applications and data are needed?

Data:

* Books:
  + Actual Text
  + Metadata (author, genre, category, …)
* User:
  + Name
  + Address
  + Paying Information

## What will the infrastructure look like?

## What standards are set?

*Specific Version of things (HTML 5, CSS 7,…)*

*Specific Browser (Google Chrome, Microsoft Edge, …)*

## How are the quality requirements achieved?

To maintain quality throughout the project, Eslint will be used to enforce a specific code style, which still is to be defined. Additionally, Unit and Acceptance Tests are going to be needed to ensure quality.

# With what should the solution be built?

## Which products and components (from which manufacturer) are required for the system?

AWS Cloud Service - AWS

Database Service - manufacturer

Google Books API - Google

## How is the system developed and rolled out?

Agile Project Management

Rolled out to cherry-picked customers

Expanding gradually -> adapting application upon feedback

Limited on Germany at first

## What verification methods are used?

Simple user verification via username/e-mail and password

## How is the solution operated?

Deployed as angular project/node app

## Who pays what?

Initial capital will be backed by investors, providing funds to get licences for books from publishers.

Users will be paying a monthly fee for a monthly subscription to lend books.

# Test Concept

Enforcing a special code style using Eslint, functions shall not get bigger than a certain size. Hence, preventing too nested and too complex functions.

## Which tests should be planned?

Tests should cover nearly 100% of the code.

Testing single methods in order of making sure edge cases are being handled properly.

## When are the tests performed?

The tests should be co-developed with each component, reducing the risk of accumulating a larger number of bugs in the code.

## Which steps should be tested?

## Which data should be entered?

Only data, which fits to a predefined scheme, making sure there will not be any type of unpredicted cases.

## What data is expected as output?