Backend

* Aufgaben des Backends
  + Wird die Adresse des Backends in die Aderszeile des Browsers eingegeben, liefert das Backend die Anwendung bzw. Die SPA (Single Page Applikation) an den Browser aus.
  + Das Backend nimmt API-Aufrufe von dem Client entgegen und liefert Daten im JSON-Format.
    - Beispiel: User klickt auf einen Button, wodurch eine Frage und Antwort im Quiz gelöscht werden soll. Der Client sendet dann eine HTTP-Anfrage an das Backend, dieses löscht die entsprechende Frage und Antwort aus der Datenbank und sendet die aktualisierte Liste an Fragen und Antworten an den Client zurück. Der Client rendert die aktualisierte Liste und stellt sie dar.
  + Damit ein User die Privilegien hat Änderung vorzunehmen, muss das Backend wissen, von wem die Anfragen kommen. Der User kann sich Einloggen, damit das Backend erkennt, welche Daten ausgeliefert werden müssen und können.
    - Dazu dient die Session auf dem Backend.
* Plattform
  + Das Backend wird auf der Node.js Plattform ausgeführt.
    - Node.js ermöglicht es, dass JavaScript außerhalb des Browsers ausgeführt werden kann.
* Module
  + HTTP
    - Anfragen und Antworten (Request, Response) über das HTTP-Protokoll.
  + Express-session
    - Damit sich User einloggen können
  + Body-parser
    - Eingehende Anfragen werden in ein Objekt geparst. Der Zugriff auf die Daten ist dadurch sehr einfach
      * Zugriff bspw.: Request.body.password
  + Express.js
    - Automatische Generierung der HTTP-Header.
    - Einfache Verarbeitung der Anfragen und Erstellung von Antworten.

Frontend

* Aufgaben des Frontends
  + Darstellung des GUI.
  + Tätigen der API-Anfragen an das Backend bzw. Web-API.
  + Darstellung des GUI auf Basis der Backend Antworten aktualisieren.
* Framework Vue.js

Datenbank

* SQL (Relationale Datenbank)
  + Die Datenbanksprache ist SQL, die für Relationale Datenbanken verwendet wird.
* DBMS (Datenbank Management System)
  + MySQL

Entwicklung

* Versionierung mit GIT.
  + Hat ein/e Entwickler/in eine Änderung bspw. Am Quellcode vorgenommen, kann diese im git-repository als Version gespeichert werden. Anschließend kann die Änderung auf den Server hochgeladen werden, auf dem das Projekt existiert, bspw. Auf [www.github.com](http://www.github.com). Das Team kann diese Version mit: $ git pull, in das Arbeitsverzeichnis laden.
    - Eine Einführung kann gerne gegeben werden.
* Frontend
  + Requirements: npm (node package manager).
  + Mit Hilfe des Paketmanagers npm wird automatisch die Entwicklungsumgebung erstellt.
    - Änderungen an den Komponenten werden im Browser, ohne aktualisieren zu müssen, sofort dargestellt.
    - Ein lokaler Server stellt die Vue-Applikation bereit.
      * Das Kommando, um den Server zu starten lautet: $ npm run dev
    - Im Browser werden Fehlermeldungen dargestellt.
    - Integration in IDE (nicht notwendig).
* Backend
  + Requirements: npm, Node.js
  + Der HTTP-Server wird einfach durch das Modul HTTP implementiert (nur wenige Zeilen Code).
* Allgemeine Reuirements
  + Npm, zur Installation und Initialisierung des Projekts.
  + Node.js, um das Backend bereitzustellen.
  + GIT, um als Team zusammen arbeiten zu können.
    - GIT gibt es als GUI und als CUI.

Deployment

* Das Bereitstellen der Applikation auf einen Server.
  + Die Applikation mit allen Abhängigkeiten wie Frontend, Backend und Datenbank wird auf dem vorher konfigurierten Server bereitgestellt.
* Die Vue-Applikation (Frontend) erstellen.
  + Wenn die Entwicklung auf dem Development-Server abgeschlossen ist, wird die Vue-Applikation mit einem einfachen Befehl erstellt: $npm run build.
    - Das Resultat ist die Vue-Applikation in Form einer JS-Datei, einer index.html-Datei und Assets wie Bilder. Alles wird in einem Ordner Namens /dist gespeichert. Das Frontend liefert letzten Endes die index.html-Datei, und somit auch die JS-Datei und Assets, aus.