

ANALYSE DER AUFGABENSTELLUNG, RECHERCHE ZUM THEMA.
ABSCHLUSS MIT EINER DETAILLIERTEN VISION

Recherchebericht

Projektleitung: Otto Bittner

12.12.2016

gbaw17

Inhaltsverzeichnis

1	Begriffe.....	3
1.1	JavaScript.....	3
1.2	Framework.....	3
1.3	Frontend/ Backend.....	4
1.4	API.....	4
1.5	WebApp.....	4
1.6	Plugin.....	4
1.7	JSON.....	5
1.8	SQL.....	5
1.9	OpenSource.....	5
1.10	Unit Test.....	5
1.11	Integrationstest.....	5
2	Konzepte.....	5
2.1	Client-Server-Architektur.....	5
2.2	Datenbanksystem.....	6
2.3	Microservices/Microservice Architektur.....	6
2.4	Ajax (Asynchronous JavaScript and XML).....	6
2.5	DOM.....	7
2.6	Pomodoro.....	7
2.7	Gamification.....	7
3	Aspekte.....	7
3.1	Ausgangslage.....	7
3.2	Ähnliche Projekte.....	8
3.2.1	Duolingo.....	8
3.2.2	Stackoverflow.....	8
3.3	Zielsetzung.....	8
3.3.1	Motivationssteigerung.....	8
3.3.2	Plattformunabhängigkeit.....	8
3.3.3	Webseitenunabhängigkeit.....	8
3.3.4	Barrierefreiheit.....	9
3.4	Probleme.....	9
3.5	Umsetzung.....	9
4	Quellen.....	9

1 Begriffe

1.1 JavaScript

JavaScript ist eine Skriptsprache, die ursprünglich für dynamisches HTML in Webbrowsern entwickelt wurde, um Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren und so die Möglichkeiten von HTML und CSS zu erweitern. Heute findet JavaScript auch außerhalb von Browsern Anwendung, so etwa auf Servern und in Microcontrollern.

1.2 Framework

Ein Framework (Rahmenstruktur) ist ein Programmiergerüst, das in der Softwaretechnik, insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen, verwendet wird. Im allgemeineren Sinne bezeichnet man mit Framework auch einen Ordnungsrahmen. Ein Framework ist selbst noch kein fertiges Programm, sondern stellt den Rahmen zur Verfügung, innerhalb dessen der Programmierer eine Anwendung erstellt, wobei u. a. durch die in dem Framework verwendeten Entwurfsmuster auch die Struktur der individuellen Anwendung beeinflusst wird.

1.3 Frontend/ Backend

Die Begriffe Front-End und Back-End werden in der Informationstechnik an verschiedenen Stellen in Verbindung mit einer Schichteneinteilung verwendet. Dabei ist typischerweise das Front-End näher am Benutzer, das Back-End näher am System. In manchen Fällen ist diese Interpretation nicht anwendbar, es gilt aber prinzipiell, dass das Front-End näher an der Eingabe und das Back-End näher an der Verarbeitung ist. Front-End bezeichnet auch grundsätzlich die Benutzeroberfläche, die z. B. in Form einer grafischen Benutzeroberfläche (GUI) oder mittels Bildschirmmasken implementiert sein kann. Bei Client-Server-Anwendungen wird das auf dem Client laufende Programm als Front-End (also hier: Dienstanutzer), das auf dem Server laufende als Back-End (hier: Dienstleister) bezeichnet.

1.4 API

Eine Programmierschnittstelle, genauer Schnittstelle zur Anwendungsprogrammierung, häufig nur kurz API genannt (application programming interface), ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. Im Gegensatz zu einer Binärschnittstelle (ABI) definiert eine Programmierschnittstelle nur die Programmanbindung auf Quelltext-Ebene. Neben dem Zugriff auf Datenbanken oder Hardware wie Festplatte oder Grafikkarte kann eine Programmierschnittstelle auch das Erstellen von Komponenten der grafischen Benutzeroberfläche ermöglichen oder vereinfachen. Heutzutage stellen auch viele Internetdienste Programmierschnittstellen zur Verfügung. Im weiteren Sinne wird die Schnittstelle jeder Bibliothek als Programmierschnittstelle bezeichnet.

1.5 WebApp

Eine Webanwendung (auch Online-Anwendung, Webapplikation oder kurz Web-App) ist ein Anwendungsprogramm nach dem Client-Server-Modell. Anders als klassische Desktopanwendungen werden Webanwendungen also nicht lokal auf dem Rechner des Benutzers installiert und dort ausgeführt. Die Datenverarbeitung und -auswertung findet stattdessen hauptsächlich auf einem entfernten Webserver statt. Nur das Ergebnis der Datenverarbeitung wird zur Anzeige oder Ausgabe an den lokalen Client-Rechner des Benutzers übertragen (Thin Client). Genutzt wird eine Webanwendung dabei in der Regel über einen Webbrowser.

1.6 Plugin

Ein Plugin ist ein optionales Software-Modul, das eine bestehende Software erweitert bzw. verändert. Plugins werden meist vom Benutzer installiert und dann von der entsprechenden Hauptanwendung während der Laufzeit eingebunden. Plug-ins können nicht ohne die Hauptanwendung ausgeführt werden. Softwarehersteller definieren oft Programmierschnittstellen (API) zu ihren Produkten, mit deren Hilfe Dritte Erweiterungen (Plug-ins) für diese Softwareprodukte programmieren können.

1.7 JSON

Die JavaScript Object Notation, kurz JSON, ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen. Jedes gültige JSON-Dokument soll ein gültiges JavaScript sein und per eval() interpretiert werden können. Davon abgesehen ist JSON aber unabhängig von der Programmiersprache.

1.8 SQL

SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen.

1.9 OpenSource

Als Open Source wird Software bezeichnet, deren Quelltext öffentlich und von Dritten eingesehen werden kann.

1.10 Unit Test

Ein Modultest wird in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (Module) von Computerprogrammen zu testen, d. h., sie auf korrekte Funktionalität zu prüfen.

1.11 Integrationstest

Der Begriff Integrationstest bezeichnet in der Softwareentwicklung eine aufeinander abgestimmte Reihe von Einzeltests, die dazu dienen, verschiedene voneinander abhängige Komponenten eines komplexen Systems im Zusammenspiel miteinander zu testen. Die erstmals im gemeinsamen Kontext zu testenden

Komponenten haben im Idealfall jeweilige Modultests erfolgreich bestanden und sind für sich isoliert fehlerfrei funktionsfähig.

2 Konzepte

2.1 Client-Server-Architektur

Die Client-Server-Architektur beschreibt eine Möglichkeit, Aufgaben, die einem gemeinsamen Ziel zuarbeiten, auf verschiedene Rechner innerhalb eines Netzwerks zu verteilen. Hierbei findet eine Unterscheidung in die Rollen des Servers und des Clients statt. Beide Rollen werden letztlich durch einen Prozess realisiert, sie können also auch auf einem Rechner ausgeführt werden.

Ziel dieser Architektur ist es zum einen, Dienste an einer zentralen Stelle verfügbar zu machen, also die Möglichkeit zu schaffen, dass mehrere Clients mit der gleichen Software oder den gleichen Daten arbeiten können, ohne diese selbst ausführen/vorhalten zu müssen. Außerdem können so die nötigen Ressourcen für einen Client teilweise stark gesenkt werden.

2.2 Datenbanksystem

Datenbanksysteme werden dazu genutzt elektronische Daten dauerhaft, effizient und widerspruchsfrei zu speichern. Besonderes Merkmal dieser Systeme ist die Möglichkeit die Daten auf verschiedene Arten darzustellen und entsprechend für verschiedene Anwender passend aufzubereiten. Außerdem stellen sie Möglichkeiten zur automatisierten Zugriffskontrolle zur Verfügung und sind im Allgemeinen wartungsfrei.

Es ist bei der Betrachtung solcher Systeme in den tatsächlichen Datenbestand (= Datenbank) und das Datenbankmanagementsystem, also die Verwaltungssoftware für ebendiese zu unterscheiden

2.3 Microservices/Microservice Architektur

Microservices sind eine Architekturstil im Bereich der Softwareentwicklung. Dabei wird eine komplexe Anwendungssoftware in autonome, kleine Komponenten aufgeteilt, die unabhängig voneinander programmiert werden und über sprachneutrale Programmierschnittstellen kommunizieren.

Den konzeptuellen Gegenpol stellen Monolithische Software-Architekturen dar, welche ihre funktionalen Elemente in einem einzigen, untrennbaren sowie homogenen Gebilde verbinden. Durch die starke Entkopplung zwischen den Komponenten werden Wartung und die Einführung neuer Programmteile erheblich vereinfacht, da hierzu nur ein einzelner Microservice erneuert/verändert werden muss.

Durch die hohe Modularität können in kleineren Teams „single-purpose“, Service orientierte Softwarebausteine entwickelt werden. Hierbei können Dinge wie Programmiersprache und Datenstruktur den Anforderungen des konkreten Microservice angepasst werden. Die Kommunikation im Team wird durch die Größe erleichtert, die Teammitglieder haben ein Überblick über den Code und der Einstieg für Neuzugänge ist einfacher. Das Zusammenspiel der einzelnen Komponente wird durch verschiedene Protokolle sichergestellt.

Dennoch bringt die verteilte Architektur eine erhöhte Komplexität, vorallem auf Netzwerkebene, mit sich. Auch die Sicherung der Datenintegrität kann eine größere Herausforderung bedeuten.

2.4 Ajax (Asynchronous JavaScript and XML)

Im Gegensatz zu einem klassischen Modell einer Webanwendung, bei der nach einer Serveranfrage die komplette Seite neu erstellt wird, kann im Ajax Modell eine asynchrone Datenübertragung zwischen dem Browser und Server stattfinden. So wird nur ein bestimmter Teil der Website nach einer Abfrage geändert während der restliche Teil unverändert angezeigt wird. Dabei wird JavaScript zur Manipulation des DOM und der dynamischen Darstellung der Inhalte genutzt. Der XML/HTTP-Request erlaubt dabei die Daten dynamisch vom Webserver aufzurufen.

2.5 DOM

Document Object Model ist eine Spezifikation für HTML und XML die eine Struktur im Sinne der objektorientierten Programmierung darstellt. Ziel ist dabei auf die strukturierten Daten einfach und einheitlich zugreifen zu können.

2.6 Pomodoro

Die Pomodoro-Technik ist ein Zeitmanagement-Konzept welches von Francesco Cirillo in den 80er Jahren entwickelt wurde. Dabei werden die Arbeiten in zumeist 25 Minuten Abschnitte unterteilt die von kleinen Pausen (meist 5 Minuten) unterbrochen werden, aller 4 Abschnitte ist dabei eine größere Pause von 20-30 min vorgesehen. Während der 25 min soll sich dabei rein auf die Aufgabe Konzentriert werden. Störungen und Unterbrechungen sind zu notieren um sich später darum kümmern zu können. Am Ende der 25 Minuten setzt man einen Schlußstrich unter das Aktuelle Pomodoro , was den unmittelbaren Effekt hat zu sehen wie viel man erledigt hat. Außerdem stellt man so Daten zur späteren Selbstreflexion bzw. für das Projektmanagement bereit. Vorteile dieses Konzeptes sind einerseits, dass erkennbar wird wie produktiv die Zeit zum Arbeiten genutzt wird. Andererseits ein Rückgang der Unterbrechungen und Ablenkungen spürbar ist, da man sich Problemlos auch nach den 25 Minuten Abschnitten darum kümmern kann beispielsweise einen Freund anzurufen oder eine E-Mail zu beantworten. Diese Zeitspanne ist auch ein guter Zeitwert um Konzentriert an etwas heranzugehen , nicht zu lang um unkonzentriert zu werden aber auch nicht zu kurz um etwas zu erreichen.

2.7 Gamification

(englisch für game = Spiel und zu deutsch als Spielifikation) bezeichnet das Einbinden von spieltypischen Elementen in eine spielfremde Umgebung. Dazu zählen z.B. Fortschrittsbalken, Erfahrungspunkte, Highscores, Ranglisten und virtuelle Güter oder Auszeichnungen. Dies soll in erster Linie zur Motivationssteigerung beitragen um wenig herausfordernde monotone oder zu komplexe Aufgaben besser bzw. motivierter zu erledigen. Gamification wurde Anfangs vor allem im Unterhaltungs- und Werbebereich verwendet, findet heute aber auch bei Fitness, im Shopping-Bereich, bei betrieblichen Anwendungen und in

Lernumgebungen häufig Verwendung. Der Erfolg von Gamification ist vorallem von der Haltung des Anwenders und die persönliche Affinität zu Spielen abhängig.

3 Aspekte

3.1 Ausgangslage

Slidewiki ist eine Webseite, die dazu dient Nutzern eine Umgebung zu schaffen in der sie, im Stil von Github gemeinsam mit anderen Nutzern, Präsentationen erstellen können. Dabei gibt es bereits erste Gamification-Ansätze in Form von Abzeichen. Diese erhält ein Nutzer bspw. für die erfolgreiche Übersetzung von 50 bzw. 100 Slides. Durch diese Errungenschaften wird ein Nutzer motiviert mehr zu arbeiten.

Unser Projekt befasst sich damit einen interaktiven "gamified bot" zu implementieren um eine Webseite ansprechender zu gestalten und die Interaktion des Nutzers mit der Webseite zu verstärken. Dieser Bot soll Nutzern die Möglichkeit bieten, nach einer gewissen Arbeitszeit Fragen zu beantworten und somit eine Arbeitspause einzulegen.

3.2 Ähnliche Projekte

3.2.1 Duolingo

Auf Duolingo können Nutzer kostenlos Sprachen lernen. Dabei wird ein spielerischer Ansatz verfolgt, wobei man bei mehrtägigem Lernen einer Sprache eine sogenannte "Streak" erhält. Dadurch kann der Nutzer verfolgen wie viele Tage er am Stück bereits eine Sprache gelernt hat und ist motiviert diese Streak fortzusetzen. Außerdem hat jeder Nutzer eine bestimmte Anzahl an Herzen, wobei man bei einer falsch beantworteten Frage ein Herz verliert. Hat man keine Herzen mehr, muss die Lektion wiederholt werden.

3.2.2 Stackoverflow

Stackoverflow ist eine online Community, auf der sich Entwickler gegenseitig beim lösen von Programmierproblemen helfen können. Die Gamification auf dieser Seite besteht darin, dass Nutzer für bestimmte Dinge, für jeden ersichtliche, Abzeichen erhalten. Wird bspw. eine Antwort auf eine Frage mindestens 100 mal positiv bewertet, erhält der Nutzer für diese Antwort ein goldenes Abzeichen ("great Answer"). Dadurch sind Nutzer eher gewillt hilfreiche und positive Antworten beizutragen.

3.3 Zielsetzung

3.3.1 Motivationssteigerung

Wir wollen Gamification mit den Grundgedanken der Pomodoro-Methode verknüpfen. Regelmäßige Pausen vom konzentrierten Arbeiten sollen in einen spielerischen Kontext eingebettet werden. Die Motivation hierbei stellen Punkte in Ranglisten und eventuell zusätzliche Belohnungen dar.

3.3.2 Plattformunabhängigkeit

Hierbei geht es uns um eine plattformübergreifende Applikation die mit unterschiedlichen Hard- und Software-Konstellationen kompatibel sein soll. Sowohl Betriebssysteme jeglicher Art wie Windows, Linux oder Mac OS, als auch verschiedene Rechnerarchitekturen sollen unterstützt werden.

3.3.3 Webseitenunabhängigkeit

Es soll eine autarke Webapplikation erstellt werden um vielfältige Einbindungsmöglichkeiten sicherzustellen und eventuell über das EU-Projekt hinaus, eine ansprechende Quiz-Applikation zur Auflockerung des Arbeitsprozesses für Jedermann zu gestalten.

3.3.4 Barrierefreiheit

Die Webapplikation sollte mit einem ansprechenden Design überzeugen und den aktuellen Standards einer Webapp entsprechen. Eine intuitive Bedienung ist sehr wichtig, da eine möglichst große Zielgruppe erreicht werden soll. Hinzu kommt natürlich erneut die Idee des Gamification-Konzepts, da ein ansprechendes Spiele-Interface zusätzlich motiviert.

3.4 Probleme

- Webanwendungen identifizieren angemeldete Benutzer meist per Session-ID. Daraus können sich Sicherheitsprobleme ergeben.
- Webanwendungen sollten im Idealfall mit allen Webbrowsern richtig funktionieren, was nicht selbstverständlich ist, da die Browser HTML teilweise unterschiedlich interpretieren.
- Durch JavaScript, neigt das System dazu sich stark aufzublasen und die Anwendung zu verlangsamen. Zusätzlich wird hierdurch die Auffindbarkeit durch Suchmaschinen verschlechtert.

3.5 Umsetzung

Da das Endprodukt eine funktionierende Webapplikation sein soll, ist die Plattformunabhängigkeit nahezu vorgegeben. Klassische Desktopanwendungen oder native Apps werden lokal auf dem Endgerät des Benutzers installiert und ausgeführt. Im Gegensatz dazu wird unsere Web-App über den Browser geöffnet und benötigt keinerlei Installationen. Sie passt sich automatisch an die jeweilige Nutzeroberfläche an und muss daher lediglich ein einziges Mal entwickelt werden, um auf allen Endgeräten zu laufen. Außerdem sollen eine ausreichende Dokumentation zur Nutzung und Einbindung unsere Applikation dafür sorgen, dass diese auch von Dritten in eigene Projekte und Webseiten eingebunden werden kann.

Darüber hinaus soll nicht nur Einbindung sondern auch Bedienung und Interface einfach gehalten werden. Hierbei lehnen wir uns an bekannte Fernsehshows und Quiz-Applikationen an. Man könne beispielsweise Antworten vorgeben, aus denen ausgewählt werden kann. Das Belohnungssystem kann dann durch eine Rangliste realisiert werden.

4 Quellen

https://de.wikipedia.org/wiki/Hypertext_Markup_Language

<https://wiki.selfhtml.org/wiki/CSS>

https://de.wikipedia.org/wiki/Cascading_Style_Sheets

<https://de.wikipedia.org/wiki/JavaScript>

<https://de.wikipedia.org/wiki/Framework>

<https://de.wikipedia.org/wiki/Microservices>

https://de.wikipedia.org/wiki/Front-End_und_Back-End

<https://de.wikipedia.org/wiki/Programmierschnittstelle>

https://de.wikipedia.org/wiki/Grafische_Benutzeroberfl%C3%A4che

<https://de.wikipedia.org/wiki/Pomodoro-Technik>

<https://de.wikipedia.org/wiki/Gamification>

<https://de.wikipedia.org/wiki/Bot>

<https://de.wikipedia.org/wiki/Webanwendung>

<http://www.startplatz.de/startup-wiki/mvp/>

<https://de.wikipedia.org/wiki/Plug-in>

https://de.wikipedia.org/wiki/JavaScript_Object_Notation

<https://de.wikipedia.org/wiki/SQL>

https://de.wikipedia.org/wiki/Open_Source

<https://de.wikipedia.org/wiki/Modultest>

<https://de.wikipedia.org/wiki/Integrationstest>

<https://de.wikipedia.org/wiki/Client-Server-Modell>

<https://de.wikipedia.org/wiki/Datenbank>

Prof. Dr. E. Rahm. WS16/17: Skript zu Datenbanksysteme 1. Universität Leipzig

<https://azure.microsoft.com/de-de/blog/microservices-an-application-revolution-powered-by-the-cloud/>

<http://www.itwissen.info/definition/lexikon/Monolithische-Software-Architektur.html>

<https://de.wikipedia.org/wiki/Microservices>

<https://de.wikipedia.org/wiki/XMLHttpRequest>

[https://de.wikipedia.org/wiki/Ajax_\(Programmierung\)](https://de.wikipedia.org/wiki/Ajax_(Programmierung))

<http://www.html-seminar.de/ajax-einfuehrung.htm>

<https://de.wikipedia.org/wiki/Gamification>

<http://slidewiki.org/>

<https://de.duolingo.com>

<http://stackoverflow.com/>