

## Dokumentation SunnyTravel

von Andrea, David, Felix, Fabio, Kai und Verena

### Business Case

Die Zielgruppe der Anwendung sind Spontanreisende, die sehr kurzfristig ein Urlaubsziel suchen, wo das Wetter gut und die Anfahrt günstig ist.

In der Regel ist es nicht leicht, innerhalb kürzester Zeit ein geeignetes Ziel zu einem günstigen Preis zu finden. Es gibt zu viele verschiedene Möglichkeiten, wo man den Urlaub verbringen kann. Auch müssen Wetterberichte eingeholt werden, wodurch noch mehr Zeit für die Entscheidung benötigt geht und weniger Zeit für die eigentliche Planung des Urlaubs bleibt. Auch kann es sein, dass einige interessante Urlaubsziele einfach nicht betrachtet werden, da man die Orte schlichtweg nicht kennt.

Somit wäre es von Vorteil, den Prozess der Entscheidung des Urlaubsziels softwaretechnisch zu unterstützen. Dazu sollen sowohl Kosten der Bahnfahrt als auch Wetterlage an beliebten Urlaubszielen automatisch eingeholt und bewertet werden, um den Nutzer bei der Entscheidungsfindung zu unterstützen.

## Anforderungen

### Qualitative Anforderungen an die Software

Das Design des User Interface soll ansprechend und klar verständlich für den Nutzer sein. Die Website muss ohne Erklärung genutzt werden können. Auch sollen dabei Benutzerhilfen, wie beispielsweise das automatische vervollständigen der Eingaben, zur Verfügung stehen.

Die Reiseoptionen sollen im Zeitraum der nächsten drei Tage zur Verfügung stehen. Weiter entfernte Reisezeiträume liegen außerhalb der Zielsetzung.

### Funktionale Anforderungen

Das Programm soll im Backend Informationen über das Wetter in bestimmten Städten in den kommenden fünf Tage speichern.

Bei Aufruf der Website soll der Nutzer sowohl den gewünschten Startbahnhof als auch den Tag der Abreise angeben. Hier sollen Eingabehilfen den Namen des Bahnhofs automatisch vervollständigen.

Nach Übergabe von Tag und Startbahnhof an das Backend wird dort das Wetter am Reisetag bewertet. Gleichzeitig wird eine Reiseroute der Deutschen Bahn zu allen Reisezielen angefordert. Anhand des Preises und des Wetters am Ziel werden alle Optionen bewertet und in absteigender Reihenfolge an das Frontend zurückgegeben. Dieser Prozess soll dabei in einer angemessenen Zeit durchgeführt werden.

Der Nutzer erhält nun eine sortierte Liste mit Reiseempfehlungen und kann anhand dieser ein Ziel auswählen. Dazu wird eine Karte implementiert, die mögliche Ziellokationen inklusive Preis- und Wetterinformationen anzeigt.

Zum Abschluss bietet die Anwendung eine Schaltfläche, die später an das Buchungsportal der Deutschen Bahn angeschlossen werden könnte.

### Rahmenbedingungen

Das Softwareprojekt soll als Gruppenarbeit im Zeitraum vom 13.05.2019 bis 24.07.2019 erstellt werden. Am Abgabedatum soll die Software präsentiert werden. Die Dokumentation sowie der Quellcode werden nachgereicht.

SunnyTravel wird als Website konzipiert. Es ist keine App-Version für Android oder iOS vorgesehen.

## Technologien

Das Backend wurde auf Basis von NodeJS entwickelt. NodeJS verwaltet Anfragen an und Antworten des Backends. Die Software wurde ausgewählt, da die Anwendung auf Basis einer Website erstellt werden soll und keine anderen Versionen geplant sind. Weiterhin ist NodeJS einfach zu implementieren und besitzt eine große Community, wodurch eine Fülle an Tutorials, Informationen, Dokumentationen und Support zur Verfügung steht.

Das Backend bezieht Daten aus einer PostgreSQL-Datenbank innerhalb eines Docker-Containers. Es wurde diese Datenbank ausgewählt, da die Implementierung dieser in die Architektur des Backends dank vorhandenem DockerImage effizient ist. Somit ist es lediglich nötig, die Tabellen in der Datenbank anzulegen und dieser einen Port zuzuweisen. Auch sind alle Gruppenmitglieder mit der Syntax von SQL vertraut, wodurch relativ wenig Einarbeitungszeit anfällt und Fehler minimiert werden können.

Die App verwendet die API der Deutschen Bahn und von OpenWeatherMap.

Die OpenWeatherMap API übergibt Informationen über Temperatur, Bewölkung, Windgeschwindigkeit, Luftdruck und Luftfeuchtigkeit an einem bestimmten Zeitpunkt an einem bestimmten Ort als JSON-Datei. Die Informationen werden in der PostgreSQL-Datenbank gespeichert. Es sind dabei 60 Aufrufe pro Minute kostenlos möglich. Somit ist es möglich, das Wetter an 60 definierten Zielorten durch die API jede Minute zu aktualisieren, oder bis zu 3600 Zielorte jede Stunde zu aktualisieren. Im Vergleich zu anderen Wetter-API wie beispielsweise AccuWeather mit 1000 Calls pro Tag oder WeatherBit mit 500 Calls pro Tag bietet OpenWeatherMap die beste Skalierbarkeit. Zusätzlich muss bei der Nutzung anderer APIs wie beispielsweise DarkSky eine Kreditkarte hinterlegt werden, die beim Überschreiten der kostenlosen Anzahl an Aufrufen für alle weiteren Calls eine kleine Gebühr berechnet.

Insgesamt bietet somit die OpenWeatherAPI eine ausreichende Menge an Calls und ist kostenlos ohne Hinterlegen eines Zahlungsmittels nutzbar.

Die API der Deutschen Bahn gibt Informationen über Route und Preis von einem Bahnhof zum anderen wieder. Dabei werden die Daten nicht in eine Datenbank geschrieben. Da die Anzahl möglicher Kombinationen aus Start- und Zielbahnhöfen mit der Fakultät wächst, würde die Menge der benötigten Anfragen bei weitem größer als die Zahl der kostenlos erlaubten Anfragen sein.

Somit wäre die Menge an Informationen nicht effizient in einer SQL-Datenbank verwaltbar.

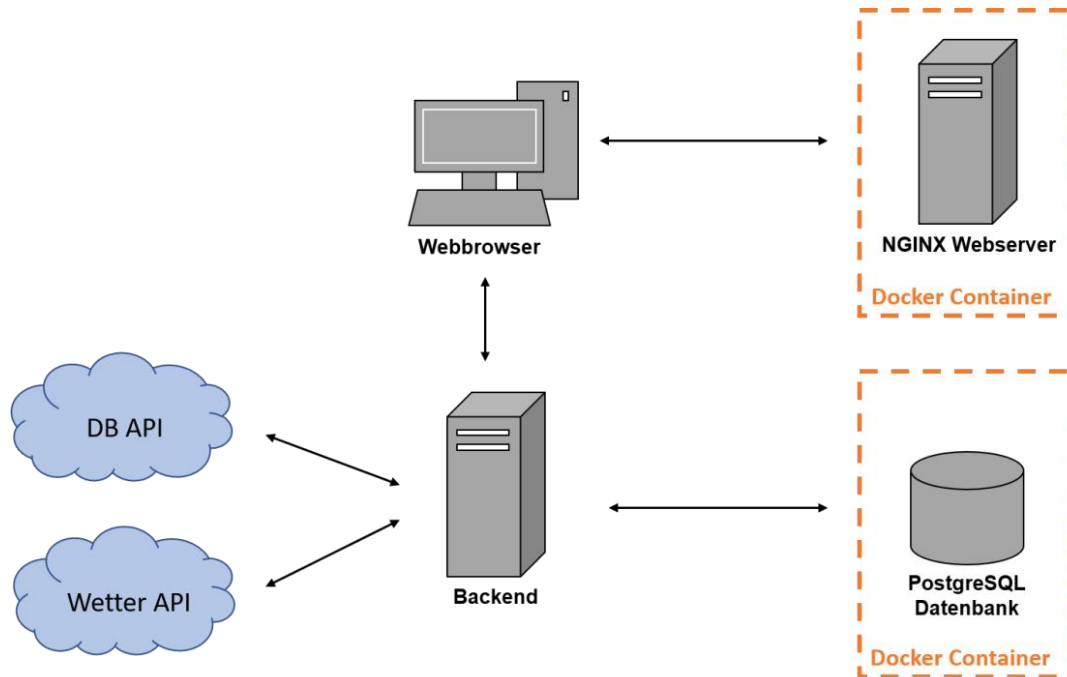
Bei der Wahl des Standardverkehrsmittel sprechen verschiedene Gründe für die Bahn. FlixBus beispielsweise fordert zur Nutzung der API eine Bewerbung per E-Mail mit Namen der Firma und Anzahl der Websitebesuche pro Monat. Für das Projekt SunnyTravel ist keine Firma gegeben, weshalb hier eventuell der Zugang zur API verwehrt sein könnte. Im Vergleich muss bei der Bahn nur ein Account angelegt werden.

Weiterhin fährt FlixBus nur von größeren Städten zu anderen größeren Städten. Nutzer müssten somit eine Möglichkeit finden, zu den Busstationen zu gelangen. Da die Anwendung die Planung der gesamten Reisestrecke übernehmen soll, würde hier die Komplexität der Anwendung drastisch steigen. Letztendlich wurde sich aus diesen Gründen gegen die Nutzung von FlixBus entschieden.

Auch wurden Flugrouten nicht in die Software miteinbezogen. Im Gegensatz zum Schienenverkehr besitzt keine Firma ein Monopol im Luftverkehr. Somit gibt es hier eine Anzahl verschiedener APIs, die theoretisch genutzt werden können. Der Aufwand, diese APIs zu bewerten und letztendlich eine Entscheidung zu finden, ist hier äußerst groß.

Durch Flugreisen würde der Umfang des Projekts drastisch steigen. Es müssten entweder mehr Urlaubsziele definiert werden, oder sich nur auf eine Region beschränkt werden. Flüge innerhalb Deutschlands sind zwar preislich günstig, aber dafür nur zwischen wenigen Städten möglich und sollten aufgrund der hohen Umweltbelastung nicht als primäre Reiseart angepriesen werden.

Letztendlich wurde aus diesen Gründen entschieden, sich lediglich auf Reiseoptionen mit der Bahn festzulegen.



Das Frontend der Software wurde unter Verwendung von HTML, CSS und JavaScript erstellt. Gleichzeitig wurden Bootstrap, JQuery und die Google Maps API verwendet.

Bootstrap wurde gewählt, um ein ansprechendes Design der Website sicherzustellen und gleichzeitig die Skalierung der Website automatisch an das Endgerät anzupassen.

Die gesamte Website wurde ohne spezielles Framework erstellt, da es sich bei der aktuellen Website nur um einen Proof of Concept handelt und das Nutzen eines Frameworks mehr Zeit in Anspruch genommen hätte. Bei der Endversion der Software ist es empfehlenswert, ein Framework zu verwenden, welches die Sicherheit der Website gewährleistet.

## Initialisierung und Programmablauf

Um das Backend zu starten, müssen zunächst Datenbank und NodeJS-Backend gestartet werden. Im Anschluss sammelt die App automatisch die Wetterdaten der OpenWeather API und speichert diese in der Datenbank.

Wenn der Webserver bereit ist, kann dieser Anfragen, bestehend aus Datum und Abfahrtsort annehmen. Dazu muss lediglich index.html im Browser geöffnet werden. Bei eingehenden Anfragen wird das Wetter an allen Zielorten am Abreisetag abgefragt. Das Wetter wird nun durch eine eigens konstruierte Funktion bewertet und mit einem Wert zwischen 1 und 100 versehen. Dabei wird auf Luftfeuchtigkeit, Temperatur, Windstärke und Bewölkung geachtet. Die Gewichtung dieser vier Faktoren wird durch Variablen im Backend repräsentiert. Es ist also möglich, diese in Zukunft zu ändern und ggf. vom Nutzer anpassen zu lassen.

Gleichzeitig sendet das Backend eine Anfrage an die API der Bahn, um eine Route mit Preis vom Startbahnhof zum Zielbahnhof für jede Zielstadt zu erhalten.

Die Informationen der Wetterbewertung und des Preises werden nun zusammen dem Frontend übergeben und in Form eines Rankings dem Nutzer ausgegeben. Gleichzeitig werden alle Zielorte auf einer Google-Map angezeigt.

Zuletzt kann der Nutzer sich die Wetterinformationen und Reisekosten der Zielorte ansehen und ein Ziel auswählen. Zur Buchung soll er in Zukunft auf die Seite der Deutschen Bahn weitergeleitet werden.

## Ausblick

Insgesamt wurde durch die momentane Version der Website bewiesen, dass die Idee umsetzbar und auch praktikabel ist. Jedoch ist die Website in einigen Bereichen, allen voran in Sicherheitsaspekten und Personalisierbarkeit, noch nicht fertig ausgereift.

Bei der Website sind noch keine Sicherheitsvorkehrungen getroffen worden, wodurch beispielsweise Code- oder SQL-Injection möglich sind. Vor einem eventuellen Release muss dies noch dringend durch Eingabevalidierung oder ein entsprechendes Framework behoben werden.

Es existiert ebenfalls noch keine Fehlerbehandlung. Sollte der Nutzer einen ungültigen Startbahnhof eingeben, wird diese Eingabe trotzdem an die Bahn API gesendet.

Auch ist eine Weiterleitung zum Ticketangebot der Deutschen Bahn in der momentanen Version noch nicht vorhanden. Weiterhin müssten für eine öffentlich zugängliche Version rechtliche Gegebenheiten bei der Nutzung der APIs beachtet werden.

Die Website könnte des Weiteren durch eine Login-Funktion erweitert werden. Dazu würde die Seite von einer zustandslosen Anwendung in eine zustandsvolle umgewandelt werden.

Weiterhin wäre es denkbar, den Nutzer über relevante Sehenswürdigkeiten an den Urlaubszielen zu informieren. Auch wäre es von Interesse, den Nutzern Hotelvorschläge für ihren Aufenthalt anzubieten. Die Website nimmt im Moment an, dass der Nutzer alleine in den Urlaub reist. Hier wäre eine Option zur Angabe der Anzahl der Reisenden sowie deren Bahncards noch gut denkbar.

Zuletzt wäre es noch möglich, den Nutzer eigene Präferenzen beim Wetter festlegen zu lassen oder verschiedene vordefinierte Gewichtungen anzubieten, beispielsweise für Stadtreisen, Strandurlaub oder Wanderurlaub.