

Dokumentation

Schriftliche Ausarbeitung zu » spotDrop «

16. März, 2021

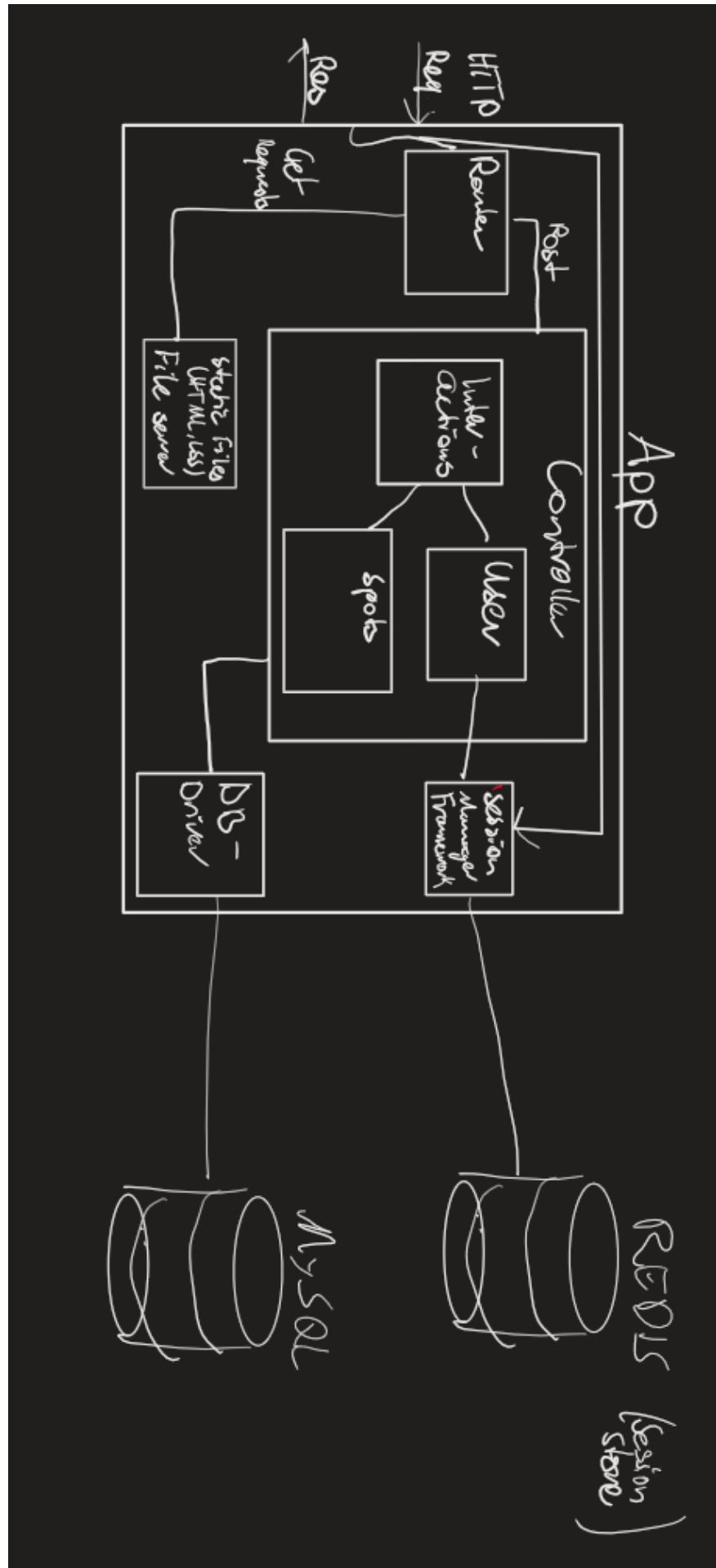
Zweck

spotDrop ist eine Plattform, die es ermöglicht, sogenannte „Spots“, also ganz bestimmte Orte, mit anderen Nutzer:innen zu teilen. Hochgeladene Spots können nach Titel oder Kategorie gefiltert gesucht und angezeigt werden. Kategorien könnten zum Beispiel inoffizielle Skate-Spots, Parkour-Spots, Spots für romantische Dates et cetera sein. Nutzer:innen sollen in der Lage sein, auf verschiedene Arten mit den Spots zu interagieren. Derzeit können Spots von nicht-registrierten Nutzer:innen angeschaut werden, registrierte User:innen können Spots hinzufügen, wieder löschen und kommentieren beziehungsweise bewerten.

Die Spots mit den meisten „Klicks“ werden auf der Startseite angezeigt. Zudem kann man die Profile anderer Nutzer:innen aufrufen, auf denen alle Spots aufgelistet werden, die ein User bisher hochgeladen hat. Geplant waren außerdem Favoritenlisten beziehungsweise Boards, auf denen beliebige Spots individuell, zum Beispiel nach Aktivitäten, gesammelt und für später gespeichert werden können. Diese Funktion wurde allerdings in der aktuellen Version, wie einige andere Überlegungen auch, noch nicht implementiert.

Umsetzung

Wir haben uns bewusst dagegen entschieden, unser Projekt mithilfe von frei verfügbaren Webservern wie Apache oder NGINX zu implementieren und haben unseren Webserver daher selbst programmiert. Einer der Beweggründe dafür war die dadurch gesteigerte Flexibilität und Erweiterbarkeit – noch viel wichtiger war uns aber, neue Erfahrungen zu sammeln und uns mit größeren Herausforderungen auseinanderzusetzen. Wir verbrachten also zusätzlich zu Planung, Entwicklung und Implementierung viel Zeit damit, neue Ansätze kennenzulernen und zu verstehen, wie man Projekte dieser Art am besten mit JavaScript umsetzt – denn darauf basiert unser Server. Wir nutzen JavaScript entsprechend multidirektional, um Informationen zwischen Webserver, einer mysql-Datenbank und dem Client auszutauschen. Grundlagen des Frontends sind also HTML5, CSS und JavaScript, häufig in Verbindung mit der Script-Bibliothek jQuery, die wir zur DOM-Navigation und -Manipulation verwenden.



Grober struktureller Aufbau des Servers, Backend-seitig

Arbeitsprozess

Da unsere Gruppe aus zwei Personen besteht, mussten wir nicht lange darüber nachdenken, wie wir die Arbeit aufteilen werden. Wir haben uns dazu entschieden, dass Leonard sich vorrangig um das Backend, also die Serverarchitektur, und Leonardo sich hauptsächlich um das Frontend kümmern wird. So konnten wir beide das machen, woran wir am meisten Spaß haben und wofür wir uns interessieren. Natürlich lief die Entwicklung trotz dessen Hand in Hand ab, sodass wir beide teils auch im Bereich des anderen tätig waren und so auch gegenseitig einiges voneinander lernen konnten.

Schwerwiegende informatische Probleme gab es überhaupt, an sich waren wir in der Lage, unsere Vorhaben Step by Step umzusetzen. Teils arbeiteten wir gemeinsam via Discord, die meiste Zeit programmierten wir aber bei Leonard zu Hause.

Wir nutzen git, um Versionsverläufe zu visualisieren und diese zu verwalten. Dank git konnten wir sehr gut individuell und unabhängig voneinander arbeiten und beiden Branches [Front-/Backend] regelmäßig mergen, um unsere Software so Stück für Stück zu vervollständigen.

Besonders geholfen hat uns Trello.com, wo wir verschiedene to-Do-Listen angefertigt haben, um einen Überblick darüber zu bewahren, was wir schon geschafft haben, was wir noch implementieren wollen und welche Funktionen zunächst optional bleiben können. So konnten wir unser ursprüngliches Pflichtenheft grafisch verwalten und Aufgaben einfach und übersichtlich zuteilen.

Ein Großteil des benötigten Wissens auf Grundlage unseres Vorwissens erlangten wir durch verschiedenste Web-Tutorials [z.B auf udemy.com], über YouTube, gezieltes googlen und natürlich via Trial-and-Error. Erstaunlicherweise hat vieles von dem, was wir uns vorgestellt haben, fast auf Anhieb funktioniert – der andere Teil war dafür aber besonders zeitintensiv, sodass wir auch die ein oder andere Nacht durchgearbeitet haben.

Für das nächste Mal wissen wir also auf jeden Fall, dass wir besonders Acht auf ein adäquates Zeitmanagement geben sollten. Insgesamt sind wir mit dem bisherigen Produkt aber ziemlich zufrieden und gelernt haben wir umso mehr.

Zukunftspläne

Da uns das Projekt ehrlich gesagt mehr Spaß bereitet hat als wir ursprünglich dachten und die Idee im Allgemeinen recht nützlich sein könnte, wollen wir unser Softwareprojekt in naher Zukunft größer skalieren und vielleicht eine umfangreichere Internetseite oder eine Mobile-App veröffentlichen. Diese sollen dann all das können, was wir uns die ganze Zeit vorgestellt haben, wozu uns aber schlichtweg die Zeit und das nötige Wissen fehlte.

Ursprünglich bestand die Idee darin, Spots teilen zu können, die keine feste Adresse haben und somit ggf. schwerer zu finden oder generell unbekannt sind. Daher besteht eines unserer vielen Vorhaben darin, das System so zu erweitern, dass ein User, wenn er sich direkt am Spot, den er teilen möchte, befindet, seine Location freigeben kann und wir somit die genauen Koordinaten des jeweiligen Ortes bestimmen können. Des Weiteren soll es eine große Karte geben, die den Aufenthaltsort des Users loggen und entsprechend die meist besuchten, neuesten oder anders gefilterte Spots in der Nähe [Umkreis] des Nutzers dynamisch anzeigen kann.

Auch die bereits angesprochenen Boards, eine Nutzerverwaltung, E-Mail-Verifikationen und die Implementierung der Secure-Variante des http-Protokolls stehen weit oben auf der Agenda.

Glücklicherweise gibt es noch viele weitere Dinge, die wir implementieren könnten. Aufgrund dieses Potentials der Skalierbarkeit sind wir besonders froh darüber, unsere Projektidee so gewählt zu haben, da wir so einen riesigen Spielraum haben, in der Ferne noch so einiges lernen und unser Projekt weiterentwickeln zu können.