



Idee

Unsere Projektidee war es, eine Web App mithilfe der Spotify API zu entwickeln, die einem Gastgeber ermöglicht, seine Gäste bei der Musikauswahl mitbestimmen zu lassen.

Unser Hauptaugenmerk lag auf der Benutzung unserer App mit einem kostenlosen Spotify Account.

Der Gastgeber legt dazu eine Party auf Basis seiner Lieblings-Spotify-Playlist mit einem Namen und einem Passwort an. Für den Anfang kann er einige Lieder festlegen, falls die Gäste zu Beginn noch nicht abstimmen. Gäste melden sich über Einscannen des Party-QR-Codes oder durch Eingabe der Party ID und des Passwortes bei der Party an. Sie können dann jedem Lied der Playlist einen „Daumen hoch“ oder einen „Daumen runter“ geben und es so in der Liste der zunächst gespielten Top Tracks auf- bzw. absteigen zu lassen. Das Lied mit der höchsten Wertung wird jeweils automatisiert als Übernächstes festgelegt.

Vorgehensweise/ Arbeitsmethoden

- Gitlab, Slack
 - Code Versionierung
 - Projektmanagement (Definition Meilensteine)
 - Interne Kommunikation
- Regelmäßige Meetings (wöchentlich)
 - Konzeption
 - Rücksprache und Austausch
 - Problembehandlung
- Einarbeitung
 - Internetrecherche (→ Technologien)
 - Tutorials (Python, Django)
 - Wissensaustausch

Technologien

Front-End

- HTML5, jQuery
- Bootstrap
- SASS
- Django Templates

Back-End

- Python
- Django (Webframework)
- SQLite3

Frontend

Da die Gäste der Party sich über mobile Endgeräte anmelden, stand die Responsiveness an oberster Stelle.

Bootstrap bietet ein sehr nützliches Grid-System und zudem viele Vorlagen für Oberflächengestaltungselemente und nützliche Plugins.

Es wird in der Basis HTML Seite, die als Template aller anderen Views dient, über einen Link auf das Content Delivery Network MaxCDN eingebunden, sodass es in jeder HTML-Seite verfügbar ist.

Bei der Gestaltung haben wir uns für Sass entschieden, da es die Verwendung von verschachtelten Regeln und Inline Importen ermöglicht und somit gut lesbar und komfortabel ist. Der CSS Präprozessor kompiliert den Sass Code anschließend zu reinem CSS in einer separaten Datei.

Javascript und jQuery kamen vor allem bei onClick-Events, beim dynamischen Erstellen von HTML-Elementen sowie bei asynchronen AJAX Requests (z.B. für Aktualisierung der Top Tracks) zum Einsatz.

Um sich Wertungen eines einzelnen Benutzers merken zu können, wird auf dem Client ein Cookie mit einer Liste seiner bewerteten Tracks angelegt. Auf diesen wird im Frontend nur lesend zugegriffen, beispielsweise nach jedem Page Reload des Benutzer Views.

Backend

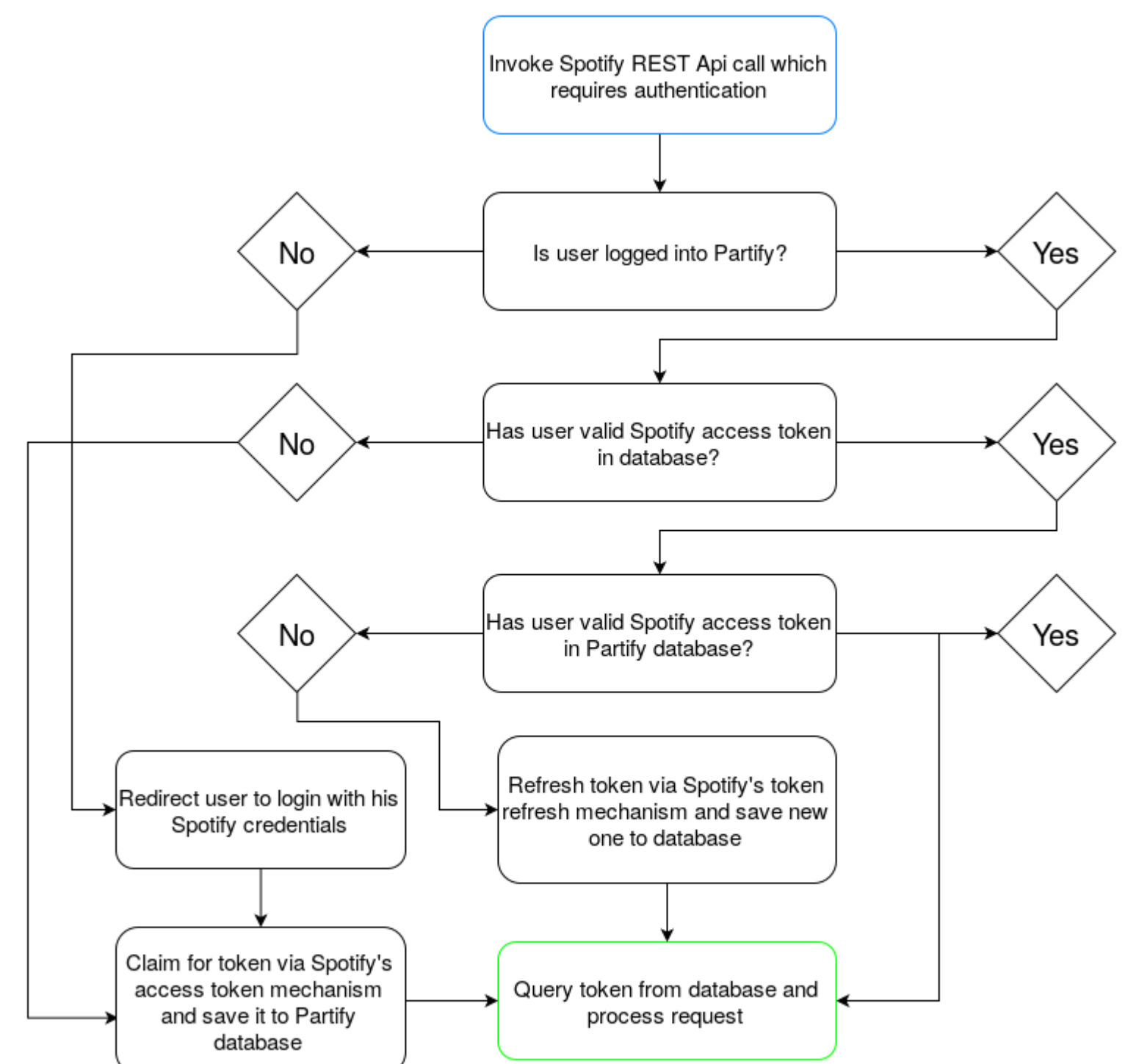
Die Umsetzung des Backends realisieren wir mithilfe des Python Webframeworks Django. Dieses bringt APIs für Datenbankzugriffe (SQL) sowie das Erstellen einer REST API mit. Weitere externe Bibliotheken haben wir verwendet für den Zugriff auf die Spotify API (spotipy), QR Code Generierung (qrcode) und für die Generierung von PDFs (reportlab).

Server

Unser Webserver läuft auf einer Linux Ubuntu Maschine im HdM Rechenzentrum, auf den wir per Key Authentication zugreifen. Darauf läuft ein Apache Webserver, welchen wir explizit für das Django Framework konfiguriert haben. Um die Seite verschlüsselt anbieten zu können haben wir ein Open SSL Zertifikat mit Hilfe des Webservices Let's Encrypt installiert.

Partify – Spotify Authentifizierungsprozess

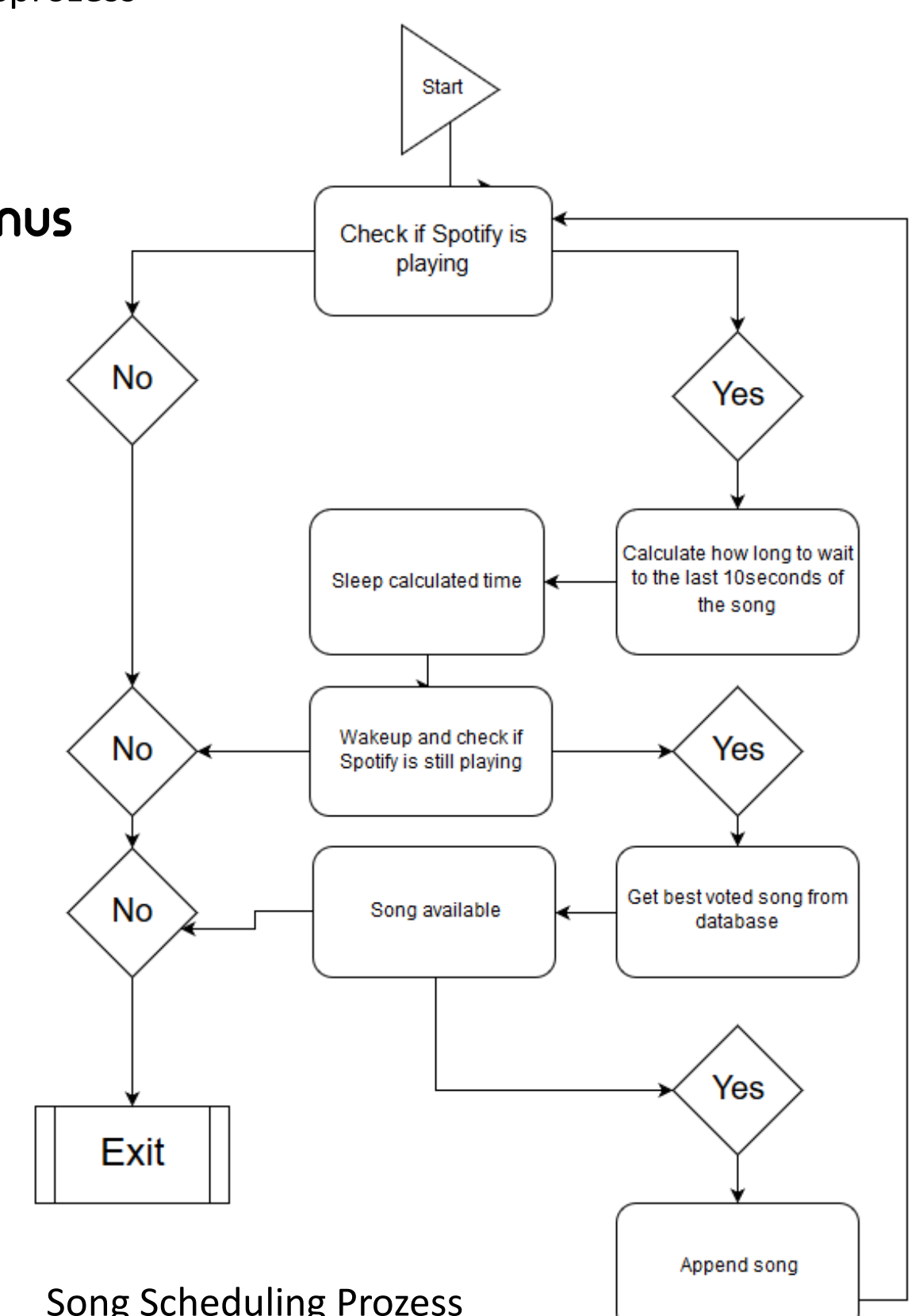
Dieser Workflow ist nötig um die Berechtigung von Spotify zu erlangen, User spezifische Daten ab zu fragen, sowie seine Playlistdaten zu manipulieren. Dabei erfordert das Nutzen verschiedener Endpoints verschiedene Berechtigungs Scopes. Dafür stellt die Spotify API einen Mechanismus bereit, nutzerbezogene Accesstoken über die REST Schnittstelle zu erlangen. Diese verlieren jedoch nach einer Stunde an Gültigkeit und müssen mit einem ebenfalls mitgesendeten Refresh token erneuert werden.



Spotify Authentifizierungsprozess

Song Scheduling Algorithmus

Die Aufgabe der Song Scheduler Klasse ist es, zum richtigen Zeitpunkt den am besten gewoteten Song zur laufenden Playlist der Party hinzuzufügen. Für jede Party werden eigene Scheduling Objekte erzeugt, die von einem statischen Song Manager verwaltet werden. Folgendes Diagramm beschreibt den Scheduling Prozess innerhalb eines Scheduling Objektes:



Song Scheduling Prozess