Music Website Project

Software Requirement Specification

Version 1.0

Prepared By Shashank Deep

Table of Contents

1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyse and give an in-depth insight of the complete web-based **Spotify Lead Generation System** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the web-based **Spotify Lead Generation System** are provided in this document.

1.1 Purpose

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2 Scope

Primarily, the scope pertains to the user dashboard and backend processing features for making Spotify Lead Generation System project live.

This SRS is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection of in-house and commercial software products. The standard can be used to create software requirements specifications directly or can be used as a model for defining a organisation or project specific standard. It does not identify any specific method, nomenclature or tool for preparing an SRS.

1.3 Definitions, Acronyms and Abbreviations

1.3 Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements and constraints and assumptions made while designing the Spotify Lead Generation System. It also gives the user viewpoint of product. Section 3 also gives the specific requirements of the product. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements. Section 4 is for supporting information.

2. Overall Description

The project entails the development of a Python-based web application aimed at providing users with a streamlined solution for generating leads related to the music industry by retrieving and filtering playlist data from Spotify. Leveraging Django as the backend framework and Celery for asynchronous task processing, the application facilitates the input of one or multiple usernames along with customisable filter options, such as playlist likes, genres, keywords and last updated date. Upon submission, the backend orchestrates the scraping of relevant playlist data from Spotify's API and stores it in a PostgreSQL database. Additionally, the application employs separate workers to accumulate and refine the retrieved data based on user preferences before presenting the final output to the user.

The following SRS contains the detail product perspective from different stakeholders. It provides the detail product functions of Spotify Lead Generation System with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

3. Specific Requirements

3.1 Functionality

3.1.1 User Account Creation

The system shall give users ability to sign up and login to the dashboard using email and password.

3.1.2 Payment Methods

The system shall give users the ability to add or remove payment methods (such as Credit Card, PayPal, etc) to facilitate purchase of paid elements.

3.1.3 Token Purchase

The system shall provide an interface to purchase credit tokens which can be used to access paid features of the website.

3.1.4 Manage Accounts

The system shall allow the users to manage basic details pertaining to their account via the dashboard.

3.1.5 Create Lead Generation Request

The system shall allow the users to create lead generation requests by spending their purchased credits. The cost for each request will vary and be calculated by the system prior to processing the request. The request will have several input parameters such as one or more Spotify profile URL(s) or username(s) as well as filters for playlists including keywords, minimum number of likes, last updated date, etc.

3.1.6 Email Notifications

The system will mail and notify the users of the completion of their queued requests.

3.1.7 View Generated Leads

The system shall allow users to view all generated lead requests and their respective data via the dashboard.

3.1.8 Export Leads To CSV

The system shall provide users a convenient method to export and download their generated leads to a CSV format.

3.1.9 Create Tickets

The system shall provide users a way to create support request tickets internally or via an external mechanism.

3.1.10 Admin Login

The system shall provide administrator level users a separate panel for management of various aspects of the tool.

3.1.11 Admin Dashboard - Manage Users

The system shall provide administrators to manage existing users including restricting or deleting any accounts that violate the product's terms of service.

3.1.12 Admin Dashboard - Manage Lead Generation Requests

The system shall provide administrators to view and manage all queued lead generation requests from users.

3.1.13 Admin Dashboard - Manage Generated Leads

The system shall provide administrators a panel for viewing all generated leads and collected data from Spotify.

3.1.14 Admin Dashboard - Refunds

The system shall provide administrators a convenient method process refunds for users in case of failed requests or any other reasons.

3.1.15 Admin Dashboard - Connect Spotify Account

The system will provide administrators an option to connect a Spotify account to the web scraping bot in order to access the official Spotify API.

3.2 Usability

3.2.1 Graphical User Interface

The system shall provide a modern and uniform look and feel to all the web pages.

The design of the forms and inputs shall take into account best user experience practices.

The system shall provide use of appropriate icons and toolbars.

3.2.2 Accessibility

The system shall provide handicap access.

The system shall provide multi language support.

3.3 Reliability & Availability

3.3.1 Cloud Hosting

The system shall be hosted on high performance cloud servers with 99.99% uptime guarantee.

The system shall ensure periodic backups in safe locations.

3.4 Performance

The product shall be based on web and be able to run from a web server.

The web servers need to be scalable and handle concurrent user requests.

The system should adopt appropriate methods to process tasks asynchronously and allow users to smoothly navigate the front-end interface.

3.5 Security

3.5.1 Data Transfer

The system shall use secure sockets in all transactions that include any confidential customer information.

The system shall confirm all transactions with the customer's web browser.

The system shall not leave any cookies on the customer's computer containing the user's password.

The system shall not leave any cookies on the customer's computer containing any of the user's confidential information.

3.5.1 Data Storage

The customer's web browser shall never display a customer's credit card number after retrieving from the database. It shall always be shown with just the last 4 digits of the credit card number.

The system's back-end servers shall never display a customer's password. The customer's password may be reset but never shown.

The system’s back-end servers shall only be accessible to authenticated administrators.

The system’s back-end databases shall be encrypted.

3.6 Third-party Constraints

The data collection mechanism of the system is highly dependent of available of data from the Spotify website and API.

3.7 Interfaces

There are many types of interfaces as such supported by the Spotify Lead Generation System namely; User Interface, Software Interface and Hardware Interface.

3.7.1 User Interfaces

The user interface for the system shall be compatible to any browser such as Internet Explorer, Mozilla or Netscape Navigator by which user can access to the system.

The user interface shall be implemented using HTML5, TailwindCSS, JavaScript (including additional libraries) and written using the jinja2 templating language.

3.7.2 Hardware Interfaces

The system can be hosted on a cloud provider such as Amazon AWS using any appropriate CI/CD pipeline.

3.7.3 Software Interfaces

1. The system shall communicate with the Django backend and API to load data on the webpages.

2. The system shall communicate with a payment gateway such as RazorPay to facilitate transactions of users.

3. The Django backend shall communicate internally using Redis to offload lead generation and other time-intensive requests to Celery workers.

4. The system shall communicate with a CRM system to handle support requests.

3.6 Licensing Requirements

Not Applicable

3.7 Supporting Information

Please refer the following documents:

1. Data Flow Diagram

2. Project Plan