MediaSound Paper Delivery

1st Henry Jhonmarcos Osorio Sánchez, 2nd Juan Pablo Sotelo Rativa
Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia
hjosorios@udistrital.edu.co , jpsotelor@udistrital.edu.co

***Abstract*—**
This research suggests a new podcast software application that includes elements of recommendations, trends, and categories to improve the podcast searching experience. With the help of good use of language, structure and design patterns in the software, we get related podcasts according to each person's choice. To provide an agile and effective system we propose using Python as a programming language, in turn with Django we will develop a user interface for this application to guarantee its proper functioning. Among other things, the potential of software in podcast content marketing is also discussed, along with the implications of podcast technology to enable safe, free-speech and transparent content distribution. This study presents a solution that improves podcast technology, benefiting both listeners and podcast creators.

***Index Terms*—Podcast, Media, User, Platform**

I. Introduction

This project consists of the creation of a platform that allows uploading Podcasts of free themes, and being able to make playlists with them. However, to differentiate ourselves from other similar content platforms, MediaSound's main objective is to choose freedom of expression over cancel culture. Leaving aside technicalities, the main problem that we can encounter when creating this platform is scalability, since, in a hypothetical case of exponential growth within the platform, it could present problems in its operation. Also, applying design patterns to a base model is challenging, especially when the software to be created is constantly improving within optimal development. For this, we propose this document as a basis and record of the problems that arise throughout the Podcasting App programming process.

II. Methods and Materials

The software development was primaly conducted using python programming language, given its versatility and extensive support in web application development. Various python frameworks and libraries were employed for different aspects of software development and testing, among which the following stand out:

1. Pytest: The Pytest testing framework was utilized for creating and executing unit tests and integration tests in the application's backend.

2.Postman: The Postman platform was used for API testing and integration testing in the application's backend. Postman facilitates sending and receiving HTTP requests, as well as response validation and test automation, which is useful for ensuring the proper functionality of APIs and system interoperability.

3. Faker: Python's Faker library was employed to generate realistic and randomized test data. Faker enables the creation of simulated data that mimics real-life application usage scenarios, aiding in the creation of test cases and the evaluation of system performance and reliability.

4. MySQL: The MySQL database management system was chosen for storing and managing the application's information. MySQL is a robust and widely used option that provides reliable performance and scalability for medium to large-scale web applications.

5. Django: It is a high-level web framework that promotes rapid and clean development. It provides a wealth of ready-to-use functionalities, such as user authentication, database administration, and session management.

The software development followed a collaborative approach between the two students, who worker together in all stages of the project. Regular meetings were established to discuss progress, review code, and resolve technical issues.

A shared repository on a version control platform such as GitHub was used to facilitate collaboration and track the work done by each student. Additionally, key steps of the development process were documented, and decisions made during the project were recorded

A. Maintaining the Integrity of the Specifications

subsection

III. Experiments and Results

In this section, we outline our approach to conducting backend testing for our podcast management application, as well as the tools and strategies we plan to employ. The primary objective of backend testing is to validate the system's behavior and functionality, detect potential errors, and ensure data integrity. Additionally, backend testing will allow us to evaluate the application's performance, scalability, and compatibility with various platforms and devices.

Our Backend testing approach will be based on a combination of unit testing, integration testing, and API testing. We will utilize the Pytest framework for writing and executing unit and integration tests, as well as the Postman platform for API testing, additionally, we use Faker library to generate realistic test data.

Our testing plan will encompass a series of test cases covering various aspects of the backend functionality, including

Identify applicable funding agency here. If none, delete this.

user authentication, podcast management, content playback, podcast search, and user interaction with the platform.

Our testing strategy will involve executing tests regularly throughout the project development lifecycle, with the aim of identifying and addressing errors in a timely manner

REFERENCES

[1] "Builder Method - Python Design Patterns - GeeksforGeeks". GeeksforGeeks. Accedido el 6 de abril de 2024. [En línea]. Disponible: https://www.geeksforgeeks.org/builder-method-python-design-patterns/

[2] A. Shvets. "Patrones de diseño en Python". Refactoring and Design Patterns. Accedido el 29 de marzo de 2024. [En línea]. Disponible: https://refactoring.guru/es/design-patterns/python

[3] "Python 3.11 Documentation". 3.12.2 Documentation. Accedido el 18 de abril de 2024. [En línea]. Disponible: https://docs.python.org/3/

[4] "How To Create A Podcast App In 2024". Dev Technosys. Accedido el 1 de abril de 2024. [En línea]. Disponible: https://devtechnosys.com/insights/create-a-podcast-app/

[5] "Singleton Pattern in Python - A Complete Guide - GeeksforGeeks". GeeksforGeeks. Accedido el 6 de abril de 2024. [En línea]. Disponible: https://www.geeksforgeeks.org/singleton-pattern-in-python-a-complete-guide/