GROUP 6

SDEV 265

3/22/2023

Software Requirements Document

1. Introduction

This document outlines the software requirements for a web application that combines a React frontend with a Django backend. The purpose of this project is to create a versatile and scalable web application, leveraging the benefits of the Django framework for backend development and React for a responsive and interactive frontend user interface. The target audience includes developers, businesses, and individuals seeking a powerful and flexible web application solution.

2. User Requirements Definition

2.1 Functional Requirements

- User authentication and authorization

- RESTful API for communication between frontend and backend

- Blog post creation, editing, and deletion

- Blog post listing and pagination

- Commenting on blog posts

2.2 Performance Requirements

- Fast response times for user interactions

- Efficient database queries for data retrieval and storage

2.3 Usability Requirements

- Intuitive and easy-to-use user interface

- Responsive design for various screen sizes

3. System Architecture

The software will be built with a modular architecture, consisting of:

- Django backend for handling database operations and API endpoints

- React frontend for rendering the user interface and handling user interactions

4. System Evolution

As the web application grows and evolves, additional features may be added, such as user profiles, advanced search capabilities, and integration with third-party services. The modular architecture allows for easy expansion and integration of new functionalities.

5. Appendices

5.1 Hardware Components

- Web server for hosting the application

5.2 Software Components

- Django framework

- React library

5.3 Selected (Proposed) Software Language

- Python (for Django backend)

- JavaScript (for React frontend)

6. Systems Development Life Cycle

6.1 Planning

- Identify the goals and objectives of the project

- Define the scope and requirements

6.2 Analysis

- Analyze user needs and system requirements

- Create use case diagrams and data flow diagrams

6.3 Design

- Design the overall system architecture

- Create wireframes and mockups for the user interface

6.4 Development

- Implement the frontend using React

- Implement the backend using Django

- Develop the RESTful API for communication between frontend and backend

6.5 Testing

- Perform unit testing on individual components

- Conduct integration testing to ensure seamless interaction between frontend and backend

6.6 Deployment

- Deploy the application to a web server

- Monitor performance and address any issues

6.7 Maintenance

- Provide updates and bug fixes

- Add new features and enhancements as needed