Project Design Document: AI-Powered Story Generation System

1. Introduction

This document outlines the design for a story generator system using a fine-tuned GPT-2 model. The system aims to generate creative stories based on user prompts, leveraging the power of advanced language models and machine learning techniques.

2. System Overview

The story generator system consists of four main components: 1. Data Preparation 2. Model Fine-tuning 3. Story Generation 4. Web Interface

3. Detailed Component Design

3.1 Data Preparation

- **Input**: Cleaned creative writing dataset (CSV format)
- **Process**:
  - Load dataset using Hugging Face's `datasets` library
  - Preprocess text data (tokenization, padding, truncation)
- **Output**: Tokenized and preprocessed dataset ready for model training

3.2 Model Fine-tuning

- **Input**: Preprocessed dataset, pre-trained GPT-2 model
- **Process**:
  - Load pre-trained GPT-2 model using Hugging Face's `transformers` library
  - Set up training arguments (learning rate, batch size, number of epochs, etc.)
  - Fine-tune the model using Hugging Face's `Trainer` class
  - Implement early stopping to prevent overfitting
- **Output**: Fine-tuned GPT-2 model

3.3 Story Generation

- **Input**: User prompt, fine-tuned GPT-2 model
- **Process**:
  - Tokenize user prompt
  - Generate text using the fine-tuned model
  - Implement temperature and top-k sampling for diverse outputs
- **Output**: Generated story text

3.4 Web Interface

- **Input**: User interaction (entering prompts)
- **Process**:
  - Create a Gradio interface for user input and story display
  - Set up ngrok for public access to the local web app
- **Output**: Web-based UI for story generation

4. Technology Stack

- **Programming Language**: Python
- **Machine Learning Framework**: PyTorch
- **NLP Library**: Hugging Face Transformers
- **Experiment Tracking**: MLflow
- **Web Interface**: Gradio
- **Public Access**: ngrok

5. Data Flow

1. User enters a prompt in the web interface
2. Prompt is sent to the story generation component
3. Story generation component uses the fine-tuned model to create a story
4. Generated story is displayed in the web interface

6. Scalability and Performance Considerations

- Implement caching mechanisms for frequently used prompts
- Consider deploying the model on cloud infrastructure for better scalability
- Optimize model size and inference time for faster story generation

7. Security Considerations

- Implement input sanitization to prevent malicious prompts
- Use HTTPS for secure communication between client and server
- Regularly update dependencies to address potential vulnerabilities

8. Future Enhancements

- Implement user accounts and story saving functionality
- Add options for different story genres or writing styles
- Integrate with other creative writing tools or platforms

9. Conclusion

This design document provides a comprehensive overview of the story generator system using a fine-tuned GPT-2 model. By following this design, we can create a robust and user-friendly application for generating creative stories based on user prompts.