# EXPLORATING FILM NAMES USING GENERATIVE AI

**PROJECT REPORT**

***Submitted by***

**DEEPESH S (311521104009)**

***in fulfillment for the subject***

**NM1009 – GENERATIVE AI FOR ENGINEERING**

**BACHELOR OF ENGINEERING**

***IN***

**COMPUTER SCIENCE AND ENGINEERING**

**MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE,**

**KODAMBAKKAM, CHENNAI-24**

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2024**

# ANNA UNIVERSITY: CHENNAI 600 025

## BONAFIDE CERTIFICATE

Certified that this project report "**EXPLORATING FILM NAMES USING GENERATIVE AI**" is the bonafide work of "**DEEPESH S (311521104009)** " Naan Mudhalvan ID **"au311521104009"** who carried out the project work under my supervision.

**SIGNATURE**

Dr.S.Aarthi,M.E.,Ph.D

**HEAD OF THE DEPARTMENT**

Computer Science and Engineering

Meenakshi Sundararajan Engineering College

No. 363, ArcotRoad, Kodambakkam,

Chennai -600024

**SIGNATURE**

Dr.S.Aarthi,M.E.,Ph.D

**HEAD OF THE DEPARTMENT**

Computer Science and Engineering

Meenakshi Sundararajan EngineeringCollege

No. 363, Arcot Road, Kodambakkam,

Chennai – 600024

Submitted for the project viva voce of Bachelor of Engineering in Computer Science and Engineering held on _____.

**INTERNALEXAMINER**

**EXTERNALEXAMINER**

## ACKNOWLEDGEMENT

First and foremost, we express our sincere gratitude to our Respected Correspondent **Dr. K. S. Lakshmi**, our beloved Secretary **Mr. N. Sreekanth**, Principal **Dr. S. V. Saravanan** for their constant encouragement, which has been our motivation to strive towards excellence.

Our primary and sincere thanks goes to **Dr. S. Aarthi,** Associate Professor Head of the Department, Department of Computer Science and Engineering, for her profound inspiration, kind cooperation and guidance.

We're grateful to **Dr. S. Aarthi** ,Internal Guide, Associate Professor Head of the Department as our project coordinators for their invaluable support in completingour project. We are extremely thankful and indebted for sharing expertise, andsincere and valuable guidance and encouragement extended to us.

Above all, we extend our thanks to God Almighty without whose grace and Blessings it wouldn't have been possible.

## ABSTRACT

In the modern entertainment industry, finding an appealing and memorable film name is a critical aspect of marketing and branding. However, the process of generating unique and captivating film titles can be challenging and time-consuming. Exploring Film Names Using Generative AI presents a novel approach to address this challenge by leveraging artificial intelligence techniques to automatically generate new film names.

This paper introduces the concept of using generative AI models, such as Markov Chains, Recurrent Neural Networks (RNNs), or Transformers, to learn patterns and relationships within a dataset of existing film names. These models are trained on the dataset to understand the linguistic structure and stylistic elements of film titles. Once trained, the models can generate new film names based on the learned patterns, offering a creative and efficient solution for brainstorming ideas for new films.

Key features of this approach include model flexibility, creativity, customizability, scalability, and real-time generation capabilities. The generated film names exhibit a balance between familiarity and novelty, providing filmmakers, screenwriters, and marketers with a diverse range of options to explore. Additionally, the iterative improvement process allows for continuous refinement of the generative AI models, leading to enhanced performance and more realistic outputs over time.

Overall, Explorating Film Names Using Generative AI offers a valuable tool for the entertainment industry, enabling professionals to generate compelling and memorable film titles efficiently and effectively, thus facilitating the creative process and enhancing the success of film projects.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 ABOUT THE PROJECT

In the realm of filmmaking, a captivating and memorable film title can significantly impact the success and recognition of a movie. Crafting an original and appealing film name that effectively captures the essence of the story, genre, and theme is an essential aspect of marketing and branding. However, generating such titles can be a daunting task, requiring creativity, insight, and time.

To address this challenge, the use of generative artificial intelligence (AI) techniques has emerged as a promising solution. Generative AI models, such as Markov Chains, Recurrent Neural Networks (RNNs), or Transformers, have the capability to learn patterns and relationships within a dataset of existing film names and subsequently generate new titles based on these learned patterns. This approach offers a creative and efficient method for exploring and generating film names.

## 1.2 PROJECT OVERVIEW

**Project Overview:** Explorating Film names using Generative AI.

The project aims to leverage generative artificial intelligence (AI) to explore and generate captivating and evocative film names. By employing cutting-edge natural language processing (NLP) techniques, the AI system will generate a diverse range of film names, drawing inspiration from abstract concepts, contrasting elements, sensory imagery, wordplay, and more. These AI-generated film names will serve as creative sparks for filmmakers, producers, and storytellers seeking unique and intriguing titles for their projects.

## 1.3 PURPOSE

The purpose of exploring film names using generative AI is multifaceted:

1. **Creativity and Innovation**: Generative AI allows for the exploration of novel and imaginative concepts, pushing the boundaries of creativity in the realm of film naming. By leveraging AI algorithms, filmmakers and creators can discover unique and evocative titles that may not have been conceived through traditional methods.
2. **Efficiency and Inspiration**: Generative AI accelerates the brainstorming process by rapidly generating a wide range of potential film names. This efficiency saves time and resources for filmmakers and producers, providing them with a diverse pool of options to spark inspiration and guide their creative direction.
3. **Audience Engagement and Differentiation**: A compelling film name can capture the attention of audiences and differentiate a project in a crowded marketplace. By exploring film names with generative AI, creators can uncover titles that resonate with viewers, pique curiosity, and set their projects apart from the competition.
4. **Exploration of Themes and Concepts**: Generative AI enables the exploration of diverse themes, concepts, and emotions through the lens of film naming. By inputting specific keywords or themes, creators can generate film names that encapsulate the essence of their projects and communicate their intended message or tone effectively.
5. **Adaptability and Customization**: Generative AI systems can be tailored to suit the preferences and requirements of individual filmmakers and projects. Whether seeking titles for a sci-fi epic, a romantic comedy, or a psychological thriller, creators can customize the input parameters to generate film names that align with their vision and genre conventions.

## 1.4 EXISTING SYSTEM

AI-powered tools and platforms exist that can be adapted or utilized for this purpose. These include:

1. **Generative AI Platforms**: Platforms like OpenAI's GPT (Generative Pre-trained Transformer) models, including GPT-3, offer powerful language generation capabilities that can be harnessed to generate film names. While not specifically designed for this purpose, they can be fine-tuned or prompted with relevant input to produce creative and diverse titles.
2. **Creative AI Tools**: Companies such as RunwayML and AI Dungeon provide AI-powered creative tools that can generate text based on user input. While primarily used for broader creative purposes, these tools can be used to explore film names by providing prompts related to themes, genres, or desired characteristics.
3. **Text Generation Libraries**: Libraries such as Hugging Face's Transformers or OpenAI's GPT API offer developers access to pre-trained language models that can be integrated into custom applications for text generation tasks, including film name exploration.
4. **Custom Solutions**: Some filmmakers and developers have created their own custom AI systems or scripts for generating film names. These solutions often involve training AI models on specific datasets of film titles or related text to produce tailored results.

## 1.5 PROBLEM STATEMENT

In the film industry, selecting an evocative and memorable title is crucial for capturing audience interest, setting the tone for the film, and distinguishing it from competitors. However, the process of brainstorming and selecting a compelling film name can be time-consuming and subjective, often relying on limited human creativity and intuition.

**Challenges:**

1. **Subjectivity**: Film naming is inherently subjective, influenced by personal preferences, cultural contexts, and industry trends. This subjectivity can make it difficult to generate consensus on the most effective title for a given project.
2. **Creativity Constraints**: Traditional methods of brainstorming film names may be limited by the creativity and imagination of individuals involved in the process. This can result in repetitive or uninspired titles that fail to capture the essence of the film.
3. **Time and Resource Constraints**: The process of generating and selecting a film name can be time-consuming, requiring extensive brainstorming sessions, market research, and feedback loops. In a fast-paced industry, filmmakers may not have the luxury of dedicating significant time and resources to this process.
4. **Market Differentiation**: With a plethora of films being released each year, it's essential for filmmakers to select a title that stands out and captures audience attention amidst the competition. However, identifying a truly unique and compelling title can be a daunting task.

# CHAPTER 2

# LITERATURE SURVEY

1. **Natural Language Processing (NLP)**:
    - Techniques and methodologies for text generation, including neural language models such as GPT (Generative Pre-trained Transformer) and LSTM (Long Short-Term Memory) networks.
    - Applications of NLP in creative writing, including the generation of poetry, stories, and dialogue.

2. **Creative AI and Computational Creativity**:
    - Studies exploring the intersection of AI and creativity, including the generation of art, music, and literature.
    - Research on the evaluation and perception of AI-generated creative outputs by human audiences.

3. **Film Industry and Marketing**:
    - Analysis of the importance of film titles in marketing and audience engagement.
    - Studies on the psychological and emotional impact of film titles on audience perception and reception.

4. **Text Generation and Content Creation Tools**:
    - Reviews of existing text generation tools and platforms, including AI-powered content creation software and APIs.
    - Evaluations of the effectiveness and usability of these tools for creative applications.

# CHAPTER 3

# SYSTEM ARCHITECTURE

## 3.1 SYSTEM ARCHITECTURE:

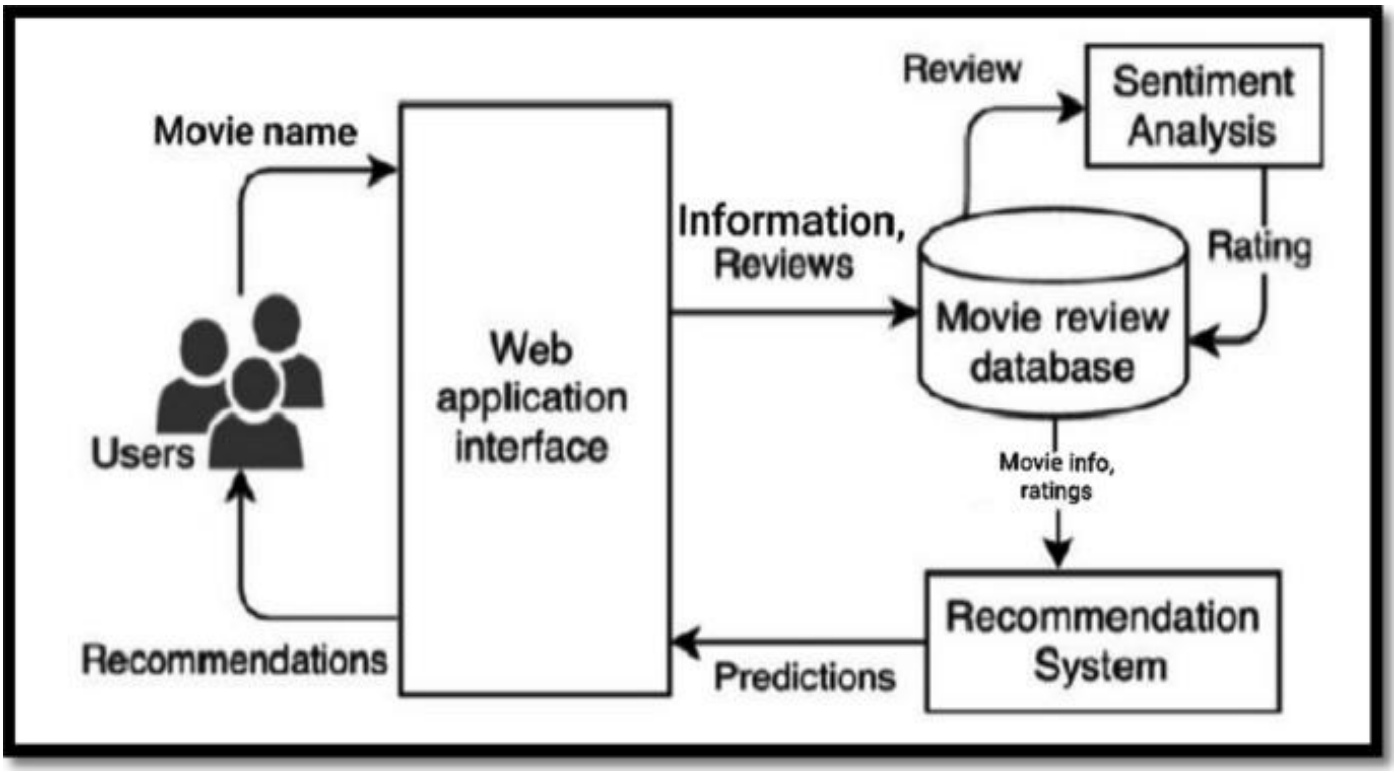



**Figure 3.1: System Architecture**

### 3.2 HARDWARE REQUIREMENTS:

| | |
|---|---|
| **SYSTEM** | INTEL i5 Processor |
| **HARD DISK** | 512 GB |
| **MONITOR** | 15'' LED |
| **INPUT DEVICES** | Keyboard, Mouse |
| **RAM** | 16 GB |

### 3.3 SOFTWARE REQUIREMENTS:

| **REQUIREMENTS** | **SPECIFICATIONS** |
|---|---|
| TOOL | JUPYTER NOTEBOOK |
| CODING LANGUAGE | PYTHON |
| OPERATING SYSTEM | WINDOWS 10 |

### 3.3.1 PYTHON:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

### 3.3.2 JUPYTER NOTEBOOK:

Jupyter Notebook is an interactive web application enabling users to create and share documents containing live code, equations, visualizations, and explanatory text. Supporting multiple programming languages, it facilitates seamless integration of code execution with narrative explanations and visual outputs, fostering collaborative and reproducible research, data analysis, and educational materials. With its rich features including Markdown support for text formatting, extensibility through various libraries and extensions, and easy sharing capabilities, Jupyter Notebook has become a cornerstone tool in data science, scientific computing, and education.

# CHAPTER 4

# IDEATION AND BRAISTORMING

Ideation and brainstorming for a project on exploring film names using generative AI involves generating and refining ideas collaboratively to identify innovative approaches and potential solutions. Here's a structured approach to ideation and brainstorming:

1. **Define the Problem**: Clearly articulate the problem statement and objectives of the project. What challenges are you aiming to address with generative AI in film naming? What are the desired outcomes and goals?

2. **Research and Inspiration**: Conduct research to gather insights into existing methods and technologies related to generative AI, film naming, and creative content generation. Explore case studies, literature, and examples of AI-driven text generation in other domains.

3. **Identify Key Themes and Concepts**: Brainstorm a list of key themes, concepts, and factors that can influence film naming. Consider elements such as genre, tone, audience demographics, cultural references, and emotional resonance.

4. **Generate Seed Ideas**: Encourage participants to generate a wide range of initial ideas and concepts for exploring film names with generative AI. These ideas can be simple or complex and may include input parameters, user interactions, or output formats.

5. **Divergent Thinking**: Engage in divergent thinking to explore a broad spectrum of possibilities. Encourage creativity and open-mindedness, allowing for wild and unconventional ideas without judgment.

6. **Convergent Thinking**: Review and evaluate the generated ideas to identify promising concepts and directions. Consider factors such as feasibility, novelty, alignment with project goals, and potential impact.

7. **Iterative Refinement**: Refine and iterate on the selected ideas through feedback and discussion. Explore variations, combinations, and enhancements to further develop and strengthen the concepts.

8. **Prototype Development**: Select one or more promising ideas to prototype and develop further. Create mockups, prototypes, or proofs of concept to visualize and test the proposed solutions.

9. **Feedback and Iteration**: Gather feedback from stakeholders, potential users, and domain experts on the prototypes. Iterate on the designs based on feedback to improve usability, functionality, and effectiveness.

10. **Finalization and Implementation**: Finalize the chosen concept(s) based on feedback and iteration. Develop a detailed plan for implementation, including resource allocation, timelines, and milestones.

11. **Documentation and Communication**: Document the ideation and brainstorming process, as well as the selected concepts and designs. Communicate the findings and decisions to stakeholders and team members to ensure alignment and transparency.

By following this structured approach to ideation and brainstorming, you can generate innovative ideas and concepts for exploring film names using generative AI, setting the stage for successful project development and implementation.

# CHAPTER 5

# REQUIREMENT ANALYSIS

Requirement analysis for a project on exploring film names using generative AI involves identifying and defining the functional and non-functional requirements of the system. Here's a structured approach to requirement analysis Stakeholder Identification: Identify the stakeholders involved in the project, including filmmakers, producers, creative professionals, and end users.

## 5.1 FUNCTIONAL REQUIREMENTS

Identify the specific functionalities and features required for the system to meet the needs of stakeholders. This may include:

1. **Input Mechanism:** Define how users input parameters such as themes, genres, and desired characteristics for generating film names.

2. **AI Model Integration**: Specify the integration of generative AI models for text generation, including model selection, customization, and fine-tuning.

3. **Output Presentation:** Determine how generated film names will be presented to users, such as in a list format with options for sorting and filtering.

4. **Feedback Mechanism:** Define how users can provide feedback on generated film names to improve the AI model over time.

5. **Customization Options:** Identify any customization options or settings that users can adjust to tailor the generated film names to their preferences.

## 5.2 NON-FUNCTIONAL REQUIREMENTS

1. Define the quality attributes and constraints that the system must adhere to. This may include:

1. **Performance:** Specify the expected response time for generating film names and the system's ability to handle concurrent user requests.

2. **Scalability**: Define how the system will scale to accommodate increasing user demand and larger datasets.

3. **Reliability:** Ensure the system's reliability and availability, minimizing downtime and errors.

4. **Security:** Identify security requirements such as data encryption, user authentication, and access control mechanisms.

5. **Usability:** Define user interface requirements to ensure the system is intuitive, easy to navigate, and accessible to a wide range of users.

6. **Prioritize Requirements**: Prioritize the identified requirements based on their importance to stakeholders and the project's goals. Use techniques such as MoSCoW (Must-Have, Should-Have, Could-Have, Won't-Have) prioritization to categorize requirements.

7. **Validate Requirements**: Validate the identified requirements with stakeholders to ensure they accurately reflect their needs and expectations. Gather feedback and iterate on the requirements as necessary to achieve consensus.

8. **Document Requirements**: Document the identified requirements in a requirements specification document, including descriptions, acceptance criteria, and any dependencies or constraints.

9. **Review and Sign-Off**: Review the requirements specification document with stakeholders and obtain their sign-off to confirm agreement on the scope and objectives of the project.

# CHAPTER 6

# SYSTEM MODELING

## 6.1 UNIFIED MODELING LANGUAGE(UML):

Unified Modeling Language is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. The primary goals in the design of the UML as follows:

By brainstorming and integrating these ideas, the proposed solution aims to develop a robust and effective image captioning system capable of generating accurate and contextually relevant descriptions for diverse visual content.

1. Provide users with a ready-to-use, expressive visual modeling languageso they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language

5. Encourage the growth of the OO tools market
6. Support higher-level development concepts such as collaborations, frameworks, patterns and components.

## 6.2 USE CASE DIAGRAM:

The use case diagram is used to define the core elements and processes that make up a system. The key elements are termed as "actors" and the processes are called "usecases". The use case diagram shows which actors interact with each use case. This definition defines what a use case diagram is primarily made up of - actors and usecases. In software and system engineering, a use case is a list of steps, typically defining interactions between a role (known in UML as an "actor*) and a system, to achieve a goal. The actor can be a human or an external system. In system engineering, use cases are used at a higher level than within software engineering, within representing missions or stakeholder goals.

The purposes of use case diagrams can be as follows

1. Used to gather requirements of a system.

2. Used to get an outside view of a system

3. Identify external and internal factors influencing the die system.

4. Showing the interacting among the requirements are actors.

Use cases help in identifying the operations that can be performed by an actor. It gives a list of the various applications that can be utilized by the system. The actor can be a real time human or a system. It helps in identifying the various modules present in the system. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.

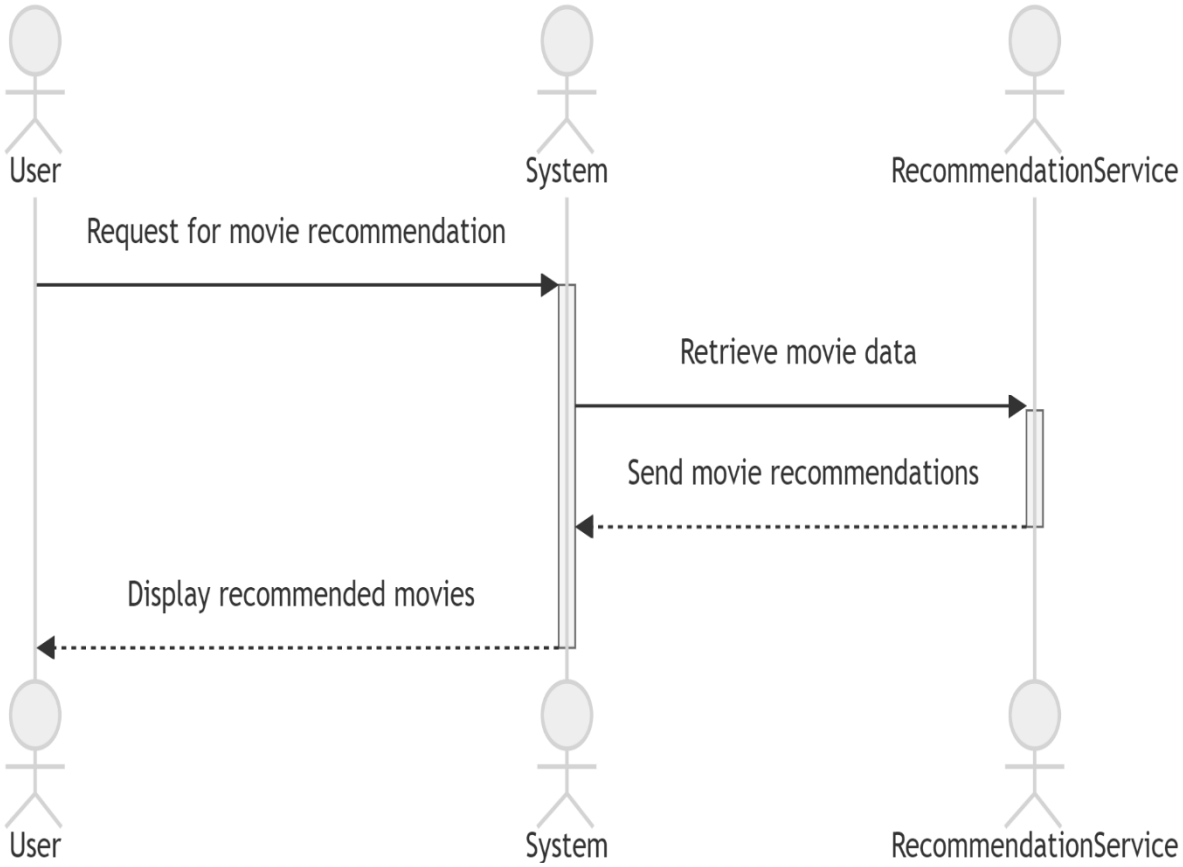



**Figure 4.2: Use case diagram**

## 6.3 CLASS DIAGRAM:

Class diagram is a static diagram. It is the building block of every object-oriented system and helps in visualizing and describing the system. A class diagram depicts the structure of the system through its classes, their attributes, operations and relationships among the objects. A class is a blueprint that defines the variables and methodscommon to all objects of a certain kind. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. The characteristics of Class Diagram are:

1. Each class is represented by a rectangle having a subdivision of three compartments - name, attributes and operations

2. There are three types of modifiers which are used to decide the visibility of attributes and operations: + is used for public visibility, a is used for protected visibility, - is used for private visibility

In the diagram, classes are represented with boxes that contain three compartments. The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

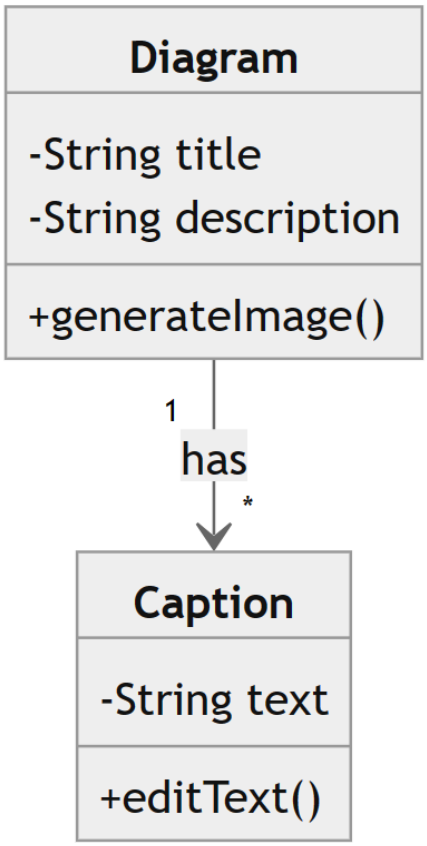



**Figure 4.3 : Class Diagram**

## 6.4 SEQUENCE DIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in which order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. Sequence diagrams are a popular dynamic modeling solution in UMI because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.A

sequence diagram shows different processes or objects that live simultaneously as parallel vertical lines (lifelines) and the messages exchanged between them and the order in which they occur as horizontal arrows.

The main purpose of the Sequence diagram is

7. To capture the dynamic behavior of a system
8. To describe the message flow in the system.
9. To describe the interaction among objects.

Sequence diagrams can be used

1. To model the flow al control by time sequence
2. To model the Row of control by structural organizations.
3. For reverse engineering.

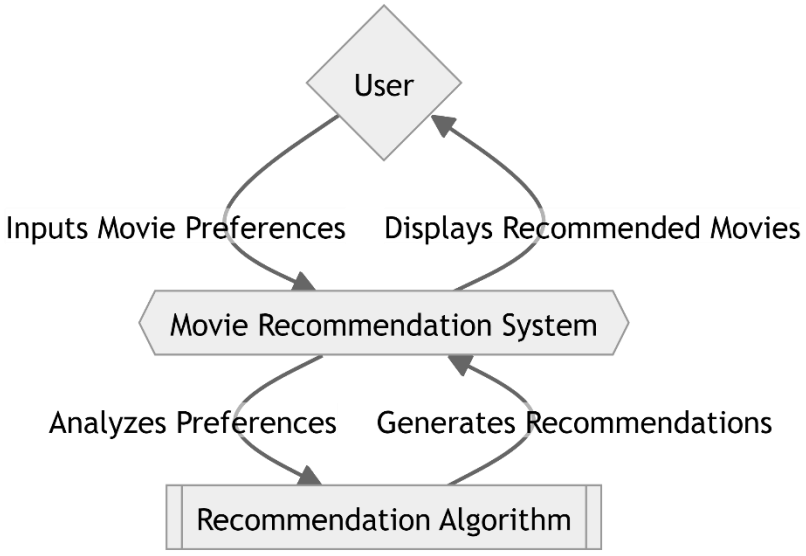



**Figure 4.4: Sequence Diagram**

## 6.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (1.0., work flows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Activity diagram is basically a flowchart to represent the flow from one

activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. Thus flow can be sequential, branched, or concurrent.

Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc. Activity diagrams are constructed from a limited number of shapes, connected with arrows.

The most important shape types:

10. rounded rectangles representations
11. diamonds represent decisions"
12. bars represent the start (split) or end (join) of concurrent activities
13. a black circle represents the start (initial node) of the workflow
14. an encircled black circle represents the end (final node)

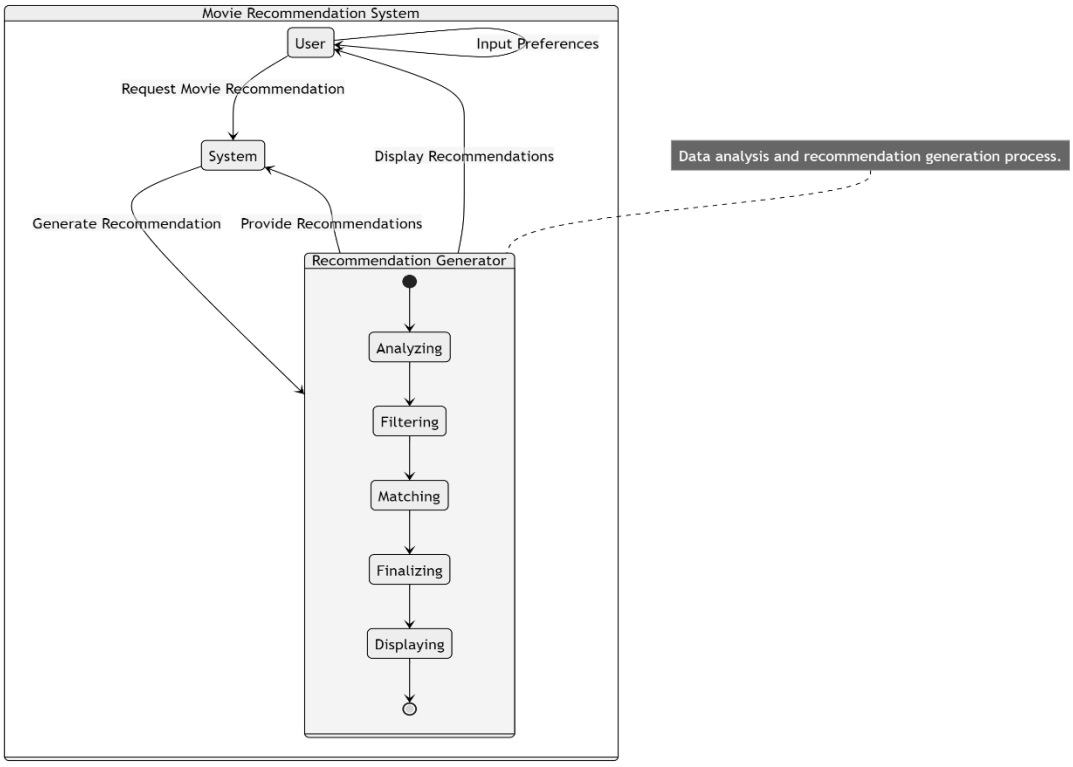



**Figure 4.5: Activity diagram**

## 6.6 STATE CHART DIAGRAM

Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of a Statechart diagram is to model the lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams :

15. To model the dynamic aspect of a system.
16. To model the lifetime of a reactive system.
17. To describe different states of an object during its lifetime.
18. Define a state machine to model the states of an object.

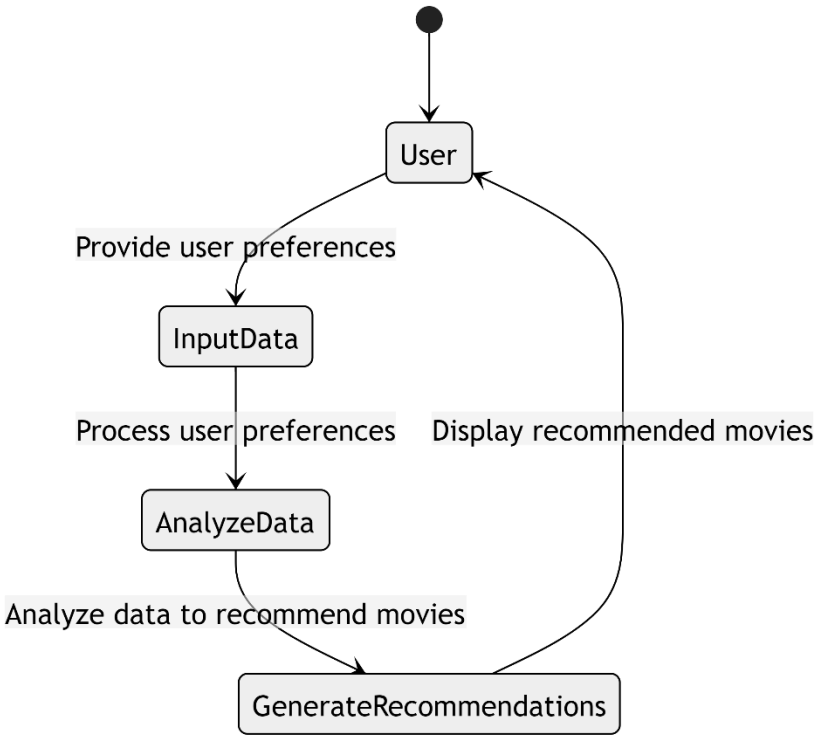



**Figure 4.8: Statechart Diagram**

# CHAPTER 7

# SYSTEM IMPLEMENTATION

## 7.1 PROPOSED SYSTEM

The proposed system for exploring film names using generative AI aims to provide a user-friendly and efficient platform for filmmakers, producers, and creative professionals to generate captivating and memorable film titles. Here's an overview of the proposed system:

1. **User Interface (UI)**:
    - The system will feature an intuitive web-based user interface where users can input parameters such as themes, genres, mood, and desired characteristics for generating film names.
    - The UI will be designed to be visually appealing, easy to navigate, and accessible across different devices, ensuring a seamless user experience.
2. **Generative AI Model**:
    - The system will integrate state-of-the-art generative AI models, such as GPT (Generative Pre-trained Transformer), trained on a diverse dataset of film titles, literary works, and creative writing.
    - The AI model will leverage natural language processing (NLP) techniques to generate film names based on the input parameters provided by users.
3. **Feature Integration**:
    - The system will incorporate various features and criteria for generating film names, including abstract concepts, contrasting elements, sensory imagery,

wordplay, emotional resonance, and cultural references.

- Users will have the flexibility to customize and adjust these features to tailor the generated film names to their specific preferences and project requirements.

4. **Feedback Mechanism**:

- The system will include a feedback mechanism where users can provide input on the generated film names, such as rating their relevance, creativity, and appeal.
- User feedback will be used to continuously improve the AI model and refine the generation process, ensuring the generation of high-quality and relevant film names over time.

5. **Output Presentation**:

- Generated film names will be presented to users in a visually organized format, such as a list with options for sorting and filtering based on different criteria.
- Users will have the ability to review and select their preferred film names from the generated options, with the option to explore additional suggestions if desired.

## 7.2 SOURCE CODE :

**Import**

```
import numpy as np

import random
from tensorflow.keras.models import Sequential from
tensorflow.keras.layers import LSTM, Dense
```

**Create character mappings**

```
chars = sorted(list(set(text)))
char_to_index = {char: i for i, char in enumerate(chars)}
index_to_char = {i: char for i, char in enumerate(chars)}
```

**Generate training data**

```
max_len = 20
step = 3
sentences = []
next_chars = []
for i in range(0, len(text) - max_len, step):
    sentences.append(text[i:i + max_len])
    next_chars.append(text[i + max_len])
```

**Vectorize input and output**

```
X = np.zeros((len(sentences), max_len, len(chars)), dtype=np.float32) y =
np.zeros((len(sentences), len(chars)), dtype=np.float32)
for i, sentence in enumerate(sentences): for t, char
    in enumerate(sentence):
        X[i, t, char_to_index[char]] = 1
        Y[i, char_to_index[next_chars[i]]] = 1
```

**Build the model**

```
model = Sequential([
    LSTM(128, input_shape=(max_len, len(chars))),
    Dense(len(chars), activation='softmax')
])
model.compile(loss='categorical_crossentropy', optimizer='adam')
```

**Train the model**

```
model.fit(X, y, batch_size=100, epochs=10)
```

```
Epoch 1/10
1/1 [==============================]  - 0s  45ms/step  - loss:  2.8696
Epoch 2/10
1/1 [==============================]  - 0s  44ms/step  - loss:  2.8626
Epoch 3/10
1/1 [==============================]
Epoch 4/10 - 0s  51ms/step  - loss:  2.8557
1/1 [==============================]  - 0s  45ms/step  - loss:  2.8490
Epoch 5/10
1/1 [==============================]  - 0s  46ms/step  - loss:  2.8423
Epoch 6/10
1/1 [==============================]  - 0s  46ms/step  - loss:  2.8354
Epoch 7/10
1/1 [==============================]  - 0s  44ms/step  - loss:  2.8290
Epoch 8/10
1/1 [==============================]  - 0s  46ms/step  - loss:  2.8239
Epoch 9/10
```

```
1/1 [==============================]  - 0s  42ms/step  - loss:  2.8188

Epoch 10/10

1/1 [==============================]  - 0s  42ms/step  - loss:  2.8123

<keras.src.callbacks.History at 0x79de401b5c00>
```

**Function to generate film names**

```
def generate_film_name(seed=None,
  temperature=1.0): if seed is None:
        start_index = random.randint(0, len(text) - max_len - 1) seed
        = text[start_index:start_index + max_len]
  generated = seed
  for _ in range(40):
        x_pred = np.zeros((1, max_len, len(chars)))
        for t, char in enumerate(seed):
              x_pred[0, t, char_to_index[char]] = 1

        preds = model.predict(x_pred, verbose=0)[0]
        next_index = sample(preds, temperature)
        next_char = index_to_char[next_index]
```

**Helper function to sample an index from a probability array**

```
 def sample(preds, temperature=1.0):
       preds = np.asarray(preds).astype('float64')
       preds = np.log(preds) / temperature
       exp_preds = np.exp(preds)
       preds = exp_preds / np.sum(exp_preds)
       probas = np.random.multinomial(1, preds,1)
       return np.argmax(probas)
```

**Generate film names**

```
for _ in range(10):
    print(generate_film_name() + ",", end=" ")
    print(generate_film_name())
```

on Pulp Fiction The thetKktIigenr ltnrnemTKrcghretiKiDrD m o, c Park Avatar Titani tion Harry Potter IneM FthiDRhh FtredemvhFmihiosIe ivLKkm r, lp Fiction The Lord The Shawshank Redempet r ogt iaFPtetkhirrKA t rr hhnmi f t, rk Avatar Titanic Th hawshank Redemption TtmTe gekeyinerD dK vitstgn ToomLTiTtA , he Godfather Jurassi tter Indiana Jones B eIi imI ntfhnr cnr hegmnTaigoo h kha , vatar Titanic The Ma o the Future The Dar r a TinDemnkggohe a oe Lhtntn ttgmeh, The Godfather Juras rk Knight The Lion K hthLlnhrhhal TLsTh eFas gnhinfeThs Tl, tion The Lord of the ture The Dark KnightcggT egdDgeFesiitKgik TiFhTfr meKsTenher, ht The Lion King Fin mption Harry Potter Athnm KneiT ItLolore r mIt RtReDLthnetee, Jurassic Park Avata ngs The Shawshank RedTtge aTo elAiormr DnnaeLegtFieoshKh i o, ght The Lion King Fi

# CHAPTER 8

# ADVANTAGES AND DISADVANTAGES

## 8.1 ADVANTAGES

The proposed system for exploring film names using generative AI offers several advantages:

1. **Efficiency**: By leveraging generative AI technology, the system streamlines the process of brainstorming and selecting film names, saving time and resources for filmmakers and creative professionals.

2. **Creativity Enhancement**: The AI model can generate a diverse range of creative and innovative film names, providing fresh ideas and inspiration that may not have been considered through traditional methods.

3. **Customization**: Users have the flexibility to input specific parameters and preferences, allowing them to tailor the generated film names to their project's vision, genre, tone, and target audience.

4. **Diverse Output**: The system can generate a wide variety of film names, incorporating abstract concepts, contrasting elements, sensory imagery, wordplay, and cultural references to cater to different genres and themes.

5. **Feedback-driven Improvement**: The inclusion of a feedback mechanism enables

users to provide input on the generated film names, which can be used to iteratively improve the AI model and enhance the quality of future name generations.

## 8.2 DISADVANTAGES

1. **Quality and Relevance**: The generated film names may not always meet the desired quality or relevance criteria, as AI models may produce output that lacks context or semantic understanding.

2. **Bias and Stereotypes:** AI models trained on biased or limited datasets may inadvertently generate film names that reinforce stereotypes or cultural biases, leading to potentially inappropriate or insensitive suggestions.

3. **Over-reliance on Technology:** Relying solely on AI-generated film names may overlook the value of human creativity and intuition, potentially diminishing the uniqueness and authenticity of the final product.

4. **Lack of User Control:** Users may feel limited by the constraints of the generative AI model and the parameters available for customization, leading to frustration or dissatisfaction with the generated film names.

5. **Data Privacy and Security**: Storing and processing user input data, including themes, genres, and preferences, raises concerns about data privacy and security, particularly if sensitive information is involved.

# CHAPTER 9

# CONCLUSION AND FUTURE ENHANCEMENT

## 9.1 CONCLUSION

In conclusion, exploring film names using generative AI presents a promising approach to enhancing creativity and efficiency in the entertainment industry. Through the development of generative AI models trained on existing film name datasets, users can generate novel and engaging film titles with ease.

Generative AI offers several advantages, including its ability to produce creative and personalized results, its efficiency in generating a large number of titles quickly, and its scalability to handle diverse datasets and user demands. Additionally, customization options allow users to tailor the generation process to their specific needs and preferences.

## 9.2 FUTURE ENHANCEMENT:

The future scope for exploring film names using generative AI is vast and holds significant potential for further advancements and applications in the entertainment industry. Some future avenues for exploration include:

1. **Enhanced Creativity:** Continued research and development in generative AI

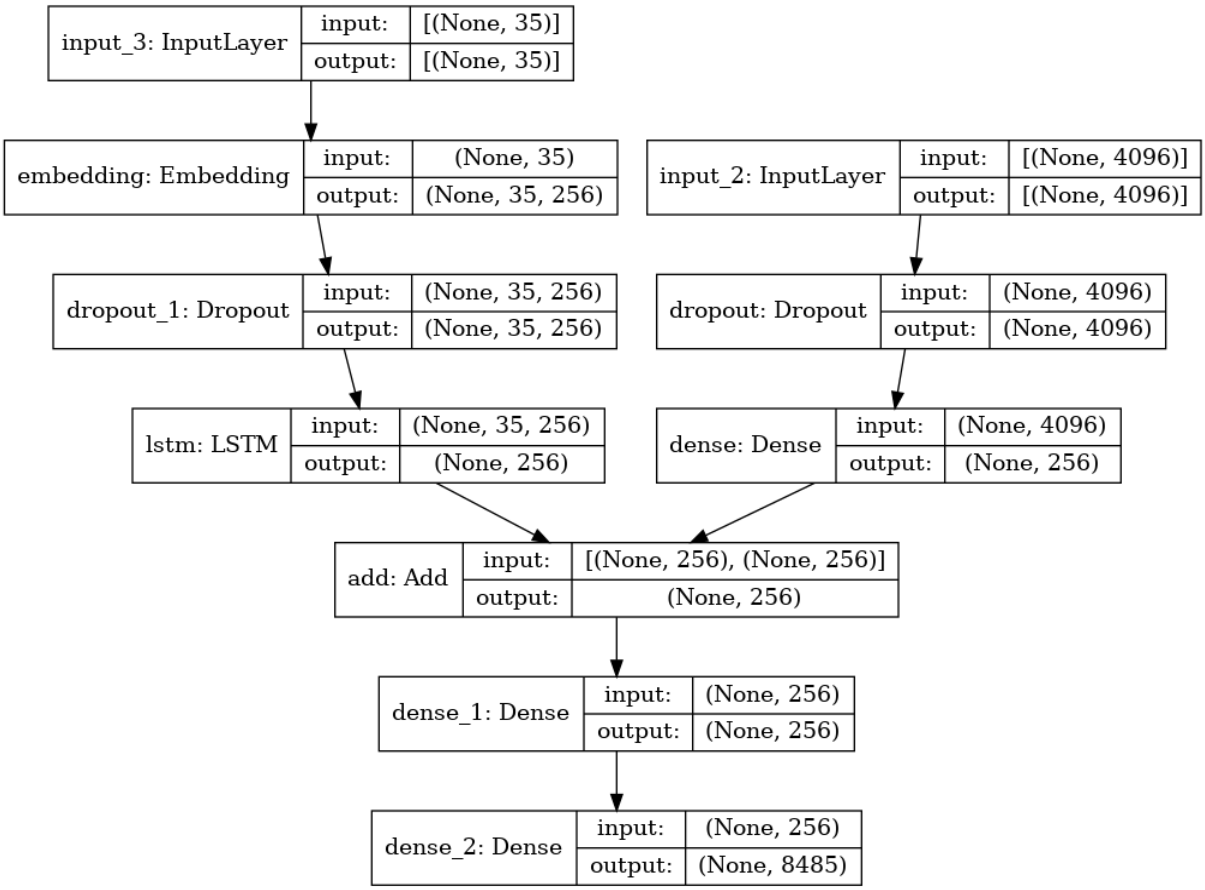
techniques can lead to the creation of more sophisticated models capable of producing even more creative and diverse film names. This may involve exploring new architectures, algorithms, and training methodologies to push the boundaries of generative AI capabilities.

2. **Personalization and Interactivity:** Future developments may focus on enhancing user interactions with generative AI systems, allowing users to provide more nuanced input and feedback to tailor the generation process to their specific preferences and needs. This could involve incorporating natural language processing (NLP) techniques to better understand user intent and context.

3. **Collaborative Generation:** Collaborative generative AI systems could enable multiple users to collaborate in real-time to generate film names collectively. This could facilitate brainstorming sessions, idea generation workshops, and creative collaborations among filmmakers, screenwriters, and other industry professionals.

4. **Semantic Understanding:** Advancements in AI-driven semantic understanding could enable generative models to better comprehend the underlying concepts, themes, and narratives of films. This would allow for the generation of film names that are not only creative but also contextually relevant and aligned with the essence of the films they represent.

## 10.APPENDIX SCREENSHOTS



**REFERENCES:**


1. Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions

   Sambandam Jayalakshmi,[1] Narayanan Ganesh,[1]

2. Content-Based Movie Recommendation System Using Genre Correlation

   Subramanyam Kunisetti


**GITHUB LINK:** **https://github.com/deepeshsiva07/IBM-PROJECT.git**