ABSTRACT

Faculty of Computers & Artificial Intelligence

Benha University

Catch The AI

A senior project submitted in partial fulfilment of the requirements for the degree of Bachelor of Computers and Artificial Intelligence.

Artificial Intelligence Department,

Project Team

1- Romani Nasrat Shawqi
2- Ahmed Mohamed Ali
3- Zeyad Elsayed Abdel-Azim
4- Sara Reda Moatamed
5- Reham Moustafa Ali
6- Rawan Abdel-Aziz Ahmed
7- Abdallah Mohammed Abdel-monnem
8- Mohannad Ayman salah
9- Mohammed Abdallah Abdel-salam

Under Supervision of

Prof. Eman Abdel-Latef

Eng. Sahar Mohamed

Artificial Intelligence,

JUNE 2024

ABSTRACT

The Growing Challenge of AI-Generated Media: A Multimodal Detection System for Enhanced Trust

The rapid evolution of deep learning, particularly Generative Adversarial Networks (GANs) and Variational Auto-encoders (VAEs), has yielded remarkably realistic AI-generated media content. While advantageous for creative fields like film and advertising, this technology presents a growing challenge for Multimedia Information Process and Retrieval (MIPR) systems, facial recognition, and speech recognition. The potential for manipulating these systems with AI-generated content raises concerns about the spread of misinformation and the erosion of trust in online interactions.

This project tackles this challenge by proposing a comprehensive model for detecting the origin of multimedia content, encompassing text, images, and audio. Utilizing advanced algorithms, the system identifies subtle manipulations indicative of AI generation. Three primary detection models form the core of this system:

- **AI Image Generation Detection:** Accurately identifies images created by artificial intelligence.
- **AI Text Generation Detection:** Recognizes text authored by AI algorithms and distinguishes it from human-generated content.
- **AI-Generated Audio Detection:** Identifies and differentiates between audio content generated by artificial intelligence and authentic human-recorded audio.

By prioritizing content authenticity across all these modalities, this project aims to safeguard against the potential pitfalls of AI-generated media. It promotes a digital environment characterized by transparency, trust, and ethical AI use, ultimately fostering a more reliable and trustworthy online ecosystem.

Table of Contents

Table of Contents

List of Figures

Chapter One

1. Introduction

1.1 Project Overview

The realm of artificial intelligence (AI) is experiencing rapid growth, with tech giants vying for dominance in groundbreaking technologies like transformers, Generative Adversarial Networks (GANs), and Large Language Models (LLMs). However, alongside these advancements arises a critical challenge: the increasing difficulty in differentiating AI-generated content from authentic human-created material. This challenge extends to images, text, and even audio, raising concerns about the spread of misinformation and the erosion of trust in online interactions. Our project proposes a comprehensive online platform equipped with specialized detection models to address this issue. By leveraging machine learning techniques, we aim to empower users to discern between AI-generated and human-created content, fostering transparency and trust in the digital landscape.

1.2 Problem Statement

The rapid evolution of AI, particularly with advanced techniques like GANs and transformers, has enabled the creation of remarkably lifelike and convincing AI-generated images, text, and audio. This progress presents a significant challenge: the increasing difficulty in distinguishing AI-generated content from real content. This situation opens doors for malicious actors to create and disseminate deceptive information, like the challenges posed by "fake news." Large Language Models (LLMs) exacerbate this problem by generating human-quality text, posing unique challenges in educational settings. As open-source AI technologies advance, the line between machine-generated and human-created media blurs, paving the way for potential misuse. Robust systems to identify content origin become paramount to protect trust and transparency in online interactions.

The potential consequences of undetected AI-generated media are vast, including the spread of misinformation, erosion of trust, academic dishonesty, and cyberbullying.

1.3 Project Objective

The objective of this project is to develop a comprehensive detection system that identifies and differentiates between human-generated and AI-generated multimedia content. By leveraging advanced algorithms and deep learning techniques, this system aims to safeguard against the proliferation of deceptive and manipulated media. The project focuses on three primary detection models:

- **AI Image Generation Detection**: Accurately identify images generated by artificial intelligence and distinguish them from human-created images.
- **AI Text Generation Detection**: Recognize text generated by AI and determine its authorship.
- **AI-Generated Audio Detection**: Identify and differentiate between audio content generated by artificial intelligence and authentic human-recorded audio.

By addressing these challenges, the project seeks to enhance the reliability of multimedia information, protect facial and speech recognition systems, and promote a trustworthy digital ecosystem through transparency and ethical AI use.

1.4 Scope and Requirements

1.4.1 key Features:

- **AI-Generated Image Detection:** Clearly identify and differentiate between AI-created and human-created images, safeguarding against the misuse of visuals.
- **AI-Generated Text Detection:** Recognize text generated by AI and distinguish it from human-written content, ensuring the authenticity of information.
- **AI-Generated Audio Detection:** Identify and differentiate between audio content generated by AI and authentic human-recorded audio.

1.4.2 Core Requirements

- **Machine Learning Model Development:** Building robust machine learning models for accurate detection across all modalities.
- **Data Collection and Preprocessing:** A substantial and well-categorized dataset for each media type (images, text, and audio) is crucial for training effective models.

1.4.3 Scope Exclusions

- **Privacy and Security Measures:** While important, these are not the focus of this project but should be considered during development.
- **Technical Challenges:** Acknowledging the exclusion of addressing all technical challenges, especially advanced, hidden AI techniques.

1.5 Target Audience

The target audience for this project includes:

- **Academics and Researchers:** Individuals conducting research in AI media detection.
- **Software Developers:** Professionals interested in implementing AI-driven solutions.
- **Students:** Those studying computer science, AI, and related fields.
- **Media Analysts:** Experts needing tools to identify AI-generated content.
- **Law Enforcement Agencies:** Utilizing AI-generated audio detection for investigating deepfakes.

1.6 Methodology

The project follows a structured methodology:

1. **Requirement Analysis**: Identify and document requirements, engage with stakeholders.
2. **System Design**: Design architecture, create UML diagrams.
3. **Data Collection and Preprocessing**: Gather and preprocess datasets.
4. **Model Development**: Utilize pretrained models for training and fine-tuning.
5. **Implementation**: Develop frontend with React, backend with Django.
6. **Integration and Testing**: Perform thorough testing.
7. **Deployment**: Deploy using Docker, host on cloud platforms.

8. **Documentation**: Document development process, prepare user manuals.

1.7 Document Structure

The document is structured as follows:

1. **Title Page**: Project title, author, institution, date.
2. **Abstract**: Summary of objectives, methods, outcomes.
3. **Table of Contents**: Sections and subsections with page numbers.
4. **Introduction**: Background, motivation, problem statement, objectives, scope.
5. **Literature Review**: Existing research and techniques.
6. **Project Methodology**: Project plan, technologies, system design, data handling, model development.
7. **Implementation**: Frontend and backend development, integration, deployment strategies.
8. **Testing and Results**: Testing methodologies, performance analysis, results.
9. **User Manual**: Installation instructions, system requirements, usage guide.
10. **Conclusion and Future Work**: Findings, limitations, future enhancements.
11. **References**: Academic papers, books, other sources.
12. **Appendices**: Additional materials such as code snippets, data samples, detailed diagrams.

Chapter Two

2. Literature Review

2.1 Existing Techniques

In recent years, the proliferation of AI-generated media has necessitated the development of robust detection techniques. Various methods have been proposed and implemented across different modalities—text, audio, and images. Each modality presents unique challenges and has prompted the development of specialized techniques.

2.1.1 Text

AI-generated text detection has evolved significantly with the advent of sophisticated language models like GPT-3 and beyond. Early approaches relied on statistical methods and linguistic features such as word frequency and sentence structure. However, these methods were often inadequate against advanced AI-generated text. Current state-of-the-art techniques leverage deep learning models, particularly those based on transformers. For example:

- **RoBERTa** (Robustly optimized BERT approach) has been fine-tuned for detecting AI-generated text by training on large datasets containing both human and AI-generated text.
- **DeBERTa** (Decoding-enhanced BERT with disentangled attention) introduces new mechanisms to improve the detection of subtle nuances in AI-generated text, which are often missed by simpler models.

2.1.2 Audio

Detection of AI-generated audio, such as synthesized speech or deepfake voices, has also seen substantial advancements. Traditional methods focused on spectral analysis and acoustic features. With the rise of powerful speech synthesis models like Wav2Vec2, more sophisticated techniques are required. Modern approaches include:

- **Wav2Vec2**: This model, originally designed for automatic speech recognition, can be adapted for detecting anomalies in speech patterns indicative of synthetic audio.
- **Mel-frequency cepstral coefficients (MFCCs)**: These are still used in conjunction with neural networks to detect inconsistencies in the spectral properties of AI-generated audio.

2.1.3 Images

The detection of AI-generated images, such as those created by GANs (Generative Adversarial Networks), involves analysing various visual features. Traditional image analysis techniques are often insufficient against high-quality GAN-generated images. Modern approaches include:

- **EfficientNet**: This model architecture, known for its efficiency and high performance on image classification tasks, is utilized to discern subtle differences between real and AI-generated images.
- **Image Forensics Techniques**: Techniques such as analysing photo response non-uniformity (PRNU) and checking for inconsistencies in lighting, shadows, and reflections are used in conjunction with deep learning models.

2.2 Strengths and Limitations

2.2.1 Strengths

- **Accuracy**: Modern deep learning models, particularly those based on transformers, have significantly improved the accuracy of AI-generated media detection.
- **Scalability**: These models can be scaled to handle large datasets, making them suitable for real-world applications.
- **Versatility**: Techniques developed for one modality (e.g., text) can often be adapted for another (e.g., audio), leveraging common underlying principles.

2.2.2 Limitations

- **Evolving Nature of AI**: As AI models improve, detection techniques must continually adapt. This ongoing "arms race" presents a significant challenge.
- **Resource Intensive**: Training and deploying sophisticated models require substantial computational resources.
- **False Positives/Negatives**: No method is perfect; there are always trade-offs between sensitivity and specificity, leading to potential misclassification.

2.3 Research Gap

While substantial progress has been made, several research gaps remain:

- **Multimodal Detection**: Most existing techniques focus on a single modality. There is a need for integrated approaches that can simultaneously handle text, audio, and images.
- **Real-time Detection**: Developing models that can operate in real-time, especially for audio and video streams, remains a significant challenge.
- **Generalizability**: Ensuring that models generalize well across different datasets and contexts is critical. Current models often perform well on specific datasets but struggle with out-of-domain samples.
- **Explainability**: Providing clear, understandable reasons for why a piece of media is classified as AI-generated or human-created is essential for trust and adoption.

By addressing these gaps, this project aims to advance the field of AI-generated media detection, providing more robust, scalable, and explainable solutions across multiple modalities

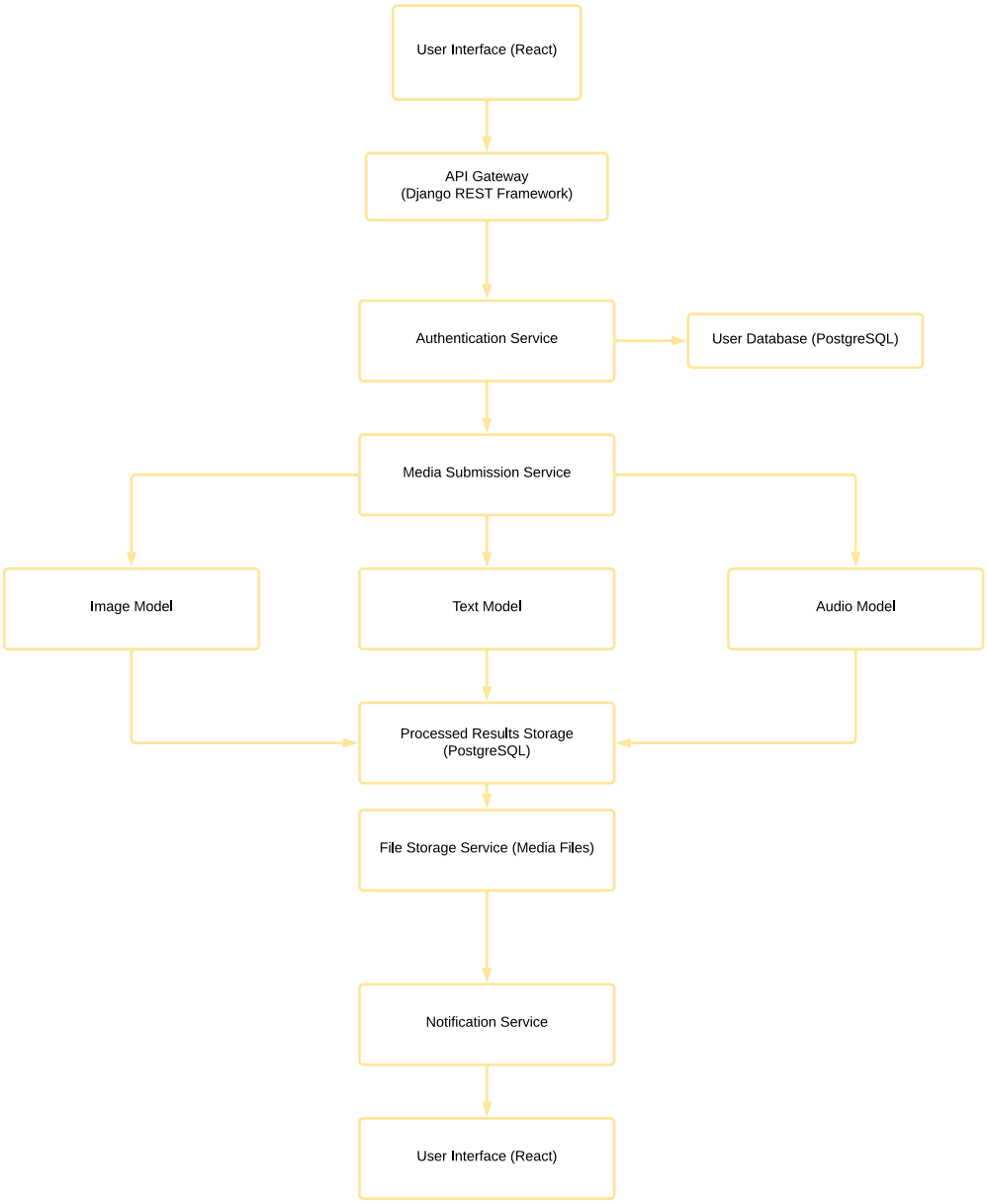
Chapter Three

3. System Design and Analysis

3.1 System Architecture

The system architecture of our AI-generated content detection platform consists of several interconnected components designed to provide a seamless and efficient user experience. The architecture is divided into three main layers: the presentation layer, the application layer, and the data layer.

3.1.1 High-Level System Diagram:



1. System Architecture

- **Presentation Layer:** This layer is responsible for user interactions and consists of the frontend application built using React. It includes user interfaces for authentication, media submission, and result visualization.
- **Application Layer:** The core functionality of the system resides in this layer, implemented using Django. It handles business logic, user authentication, media processing, and interaction with the machine learning models.
- **Data Layer:** This layer manages data storage and retrieval, using PostgreSQL for structured data (user information, media metadata) and a file storage service for media files. It also includes connections to pre-trained machine learning models for AI content detection.

Interaction Between Components:

1. **User Interaction:** Users interact with the system through the frontend, submitting media for analysis and viewing results.
2. **API Requests:** The frontend communicates with the backend via RESTful APIs to perform actions like user authentication, media submission, and fetching detection results.
3. **Media Processing:** Submitted media is processed by the backend, which invokes the appropriate machine learning models to analyse the content.
4. **Results Storage:** Analysis results are stored in the database and made available to users through the frontend interface.

3.2 System Requirements

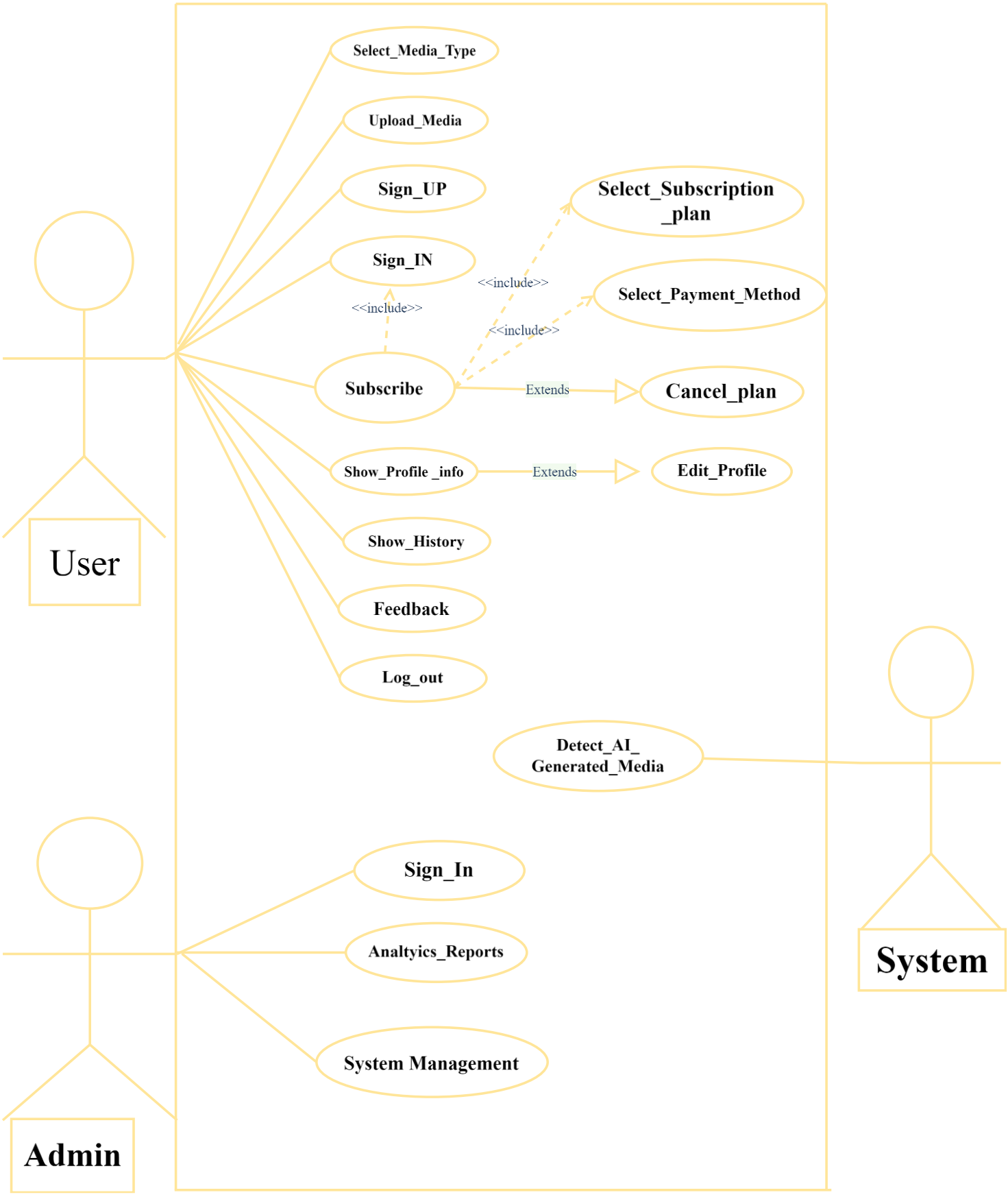
3.2.1 Functional Requirements

The system offers several key functionalities from a user's perspective:

1. **User Authentication:** Secure login and registration functionalities to manage user access.
2. **Media Submission:** Users can submit images, text, and audio files for AI-generated content detection.
3. **Detection Results:** The system provides detailed reports on the authenticity of the submitted media, indicating whether it is AI-generated or human-created.
4. **Subscription Plans:** Users can subscribe to different plans offering various levels of access and functionality (e.g., number of submissions, advanced analysis features).
5. **Admin Management:** Administrators can manage users, review system usage, and update detection models.

3. System Design and Analysis

3.2.1.1**Use Case Diagram:**



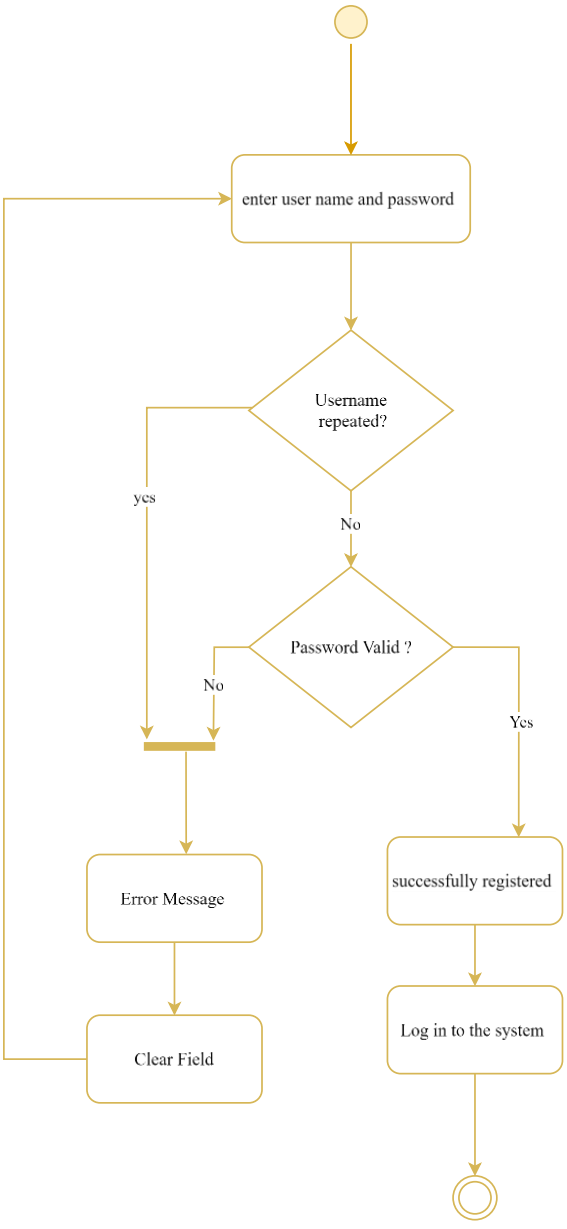
2. Use Case Diagram

Use Case Definitions:

- **User Authentication:** Users can create accounts, log in, and manage their profiles.
- **Media Submission:** Users can upload images, text, and audio for analysis.
- **View Detection Results:** Users can view detailed reports on the submitted media.
- **Subscription Management:** Users can subscribe to different service plans.
- **Admin Management:** Administrators can manage users and system settings.

3.2.1.2 **Activity Diagram:**

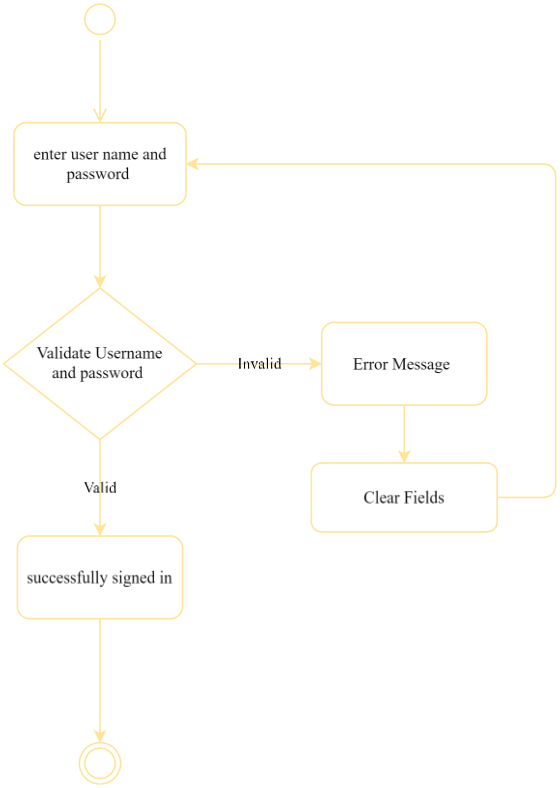
1. Sign up activity:



3. Sign up activity

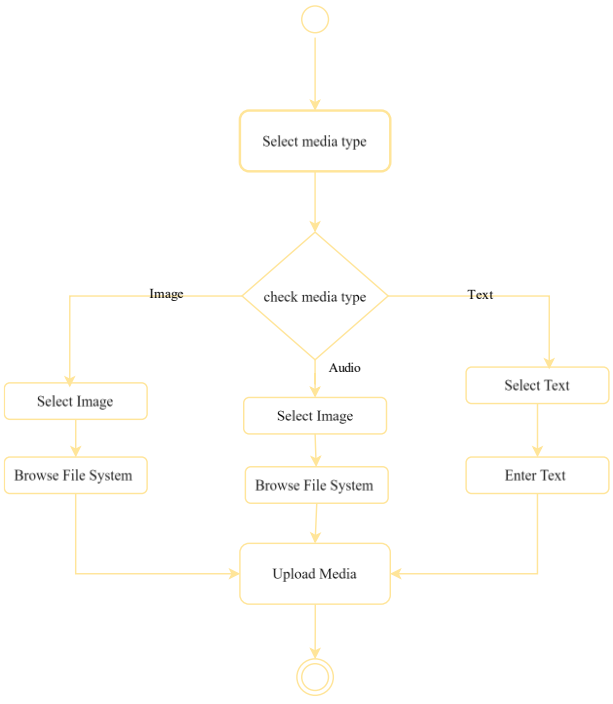
3. System Design and Analysis

2. Sign in activity:



4. Sign in activity

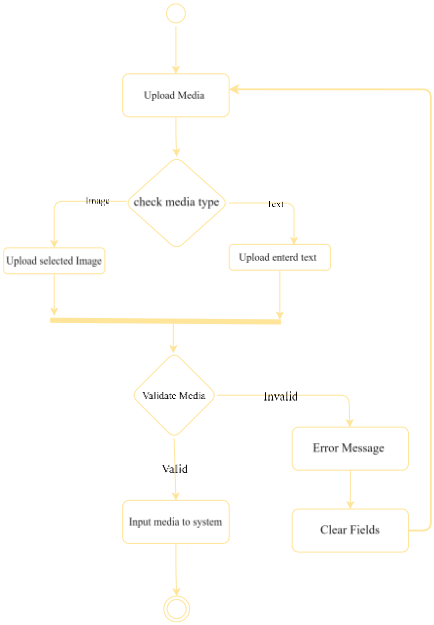
3. Select media type activity:



5. Select media type activity

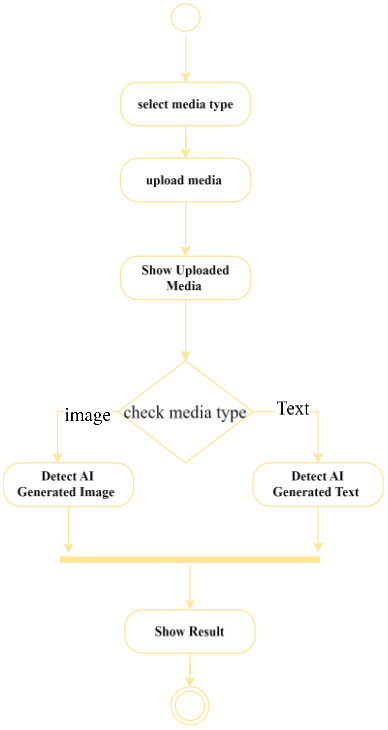
3. System Design and Analysis

4. Upload media activity



6. Upload media activity

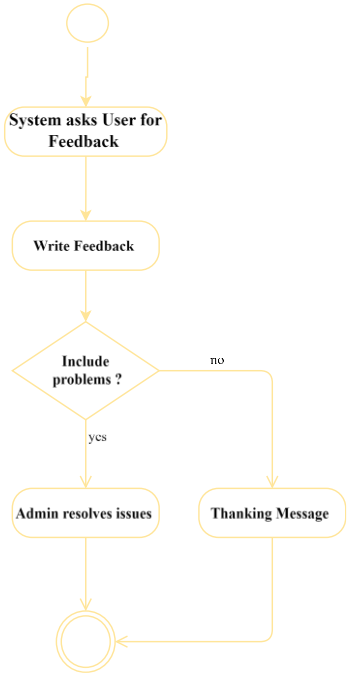
5. Detect AI-generated Media activity:



7. Detect AI-generated Media activity

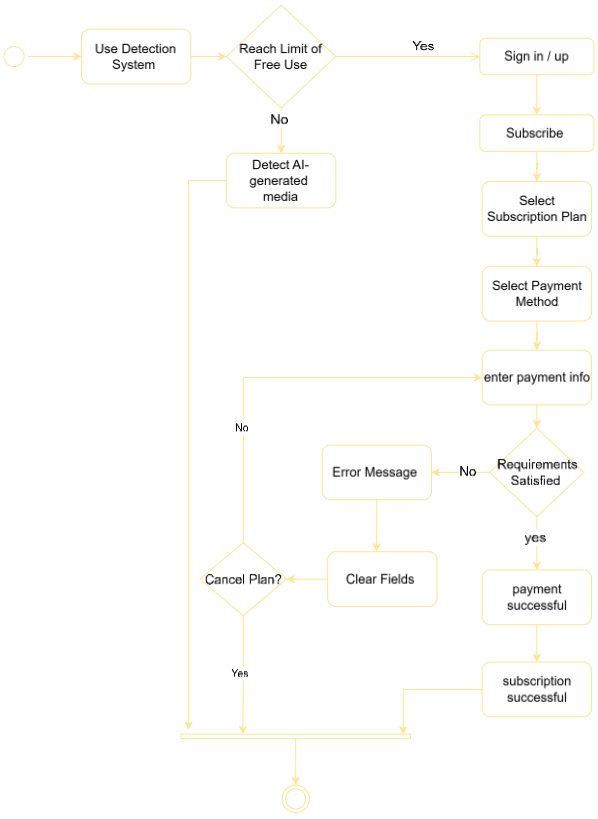
3. System Design and Analysis

6. Feedback activity:



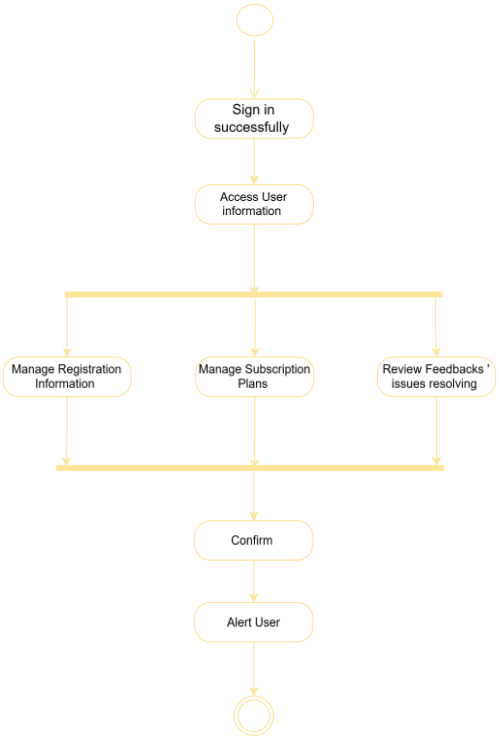
8. Feedback activity

7. Subscription activity:



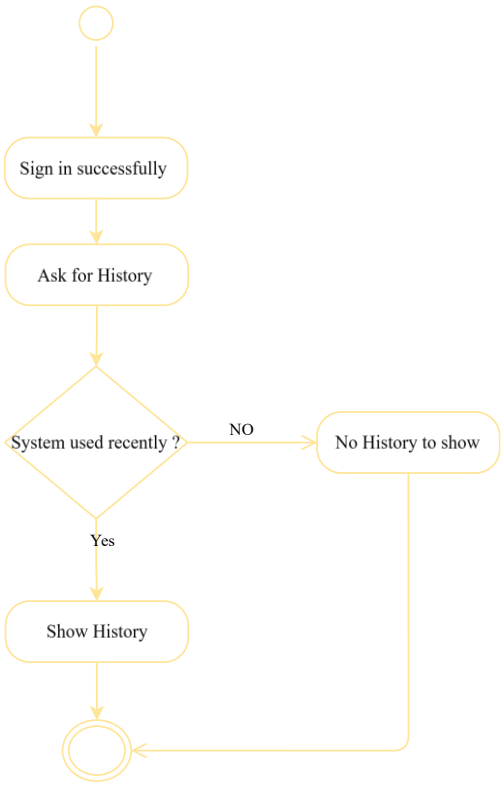
9. Subscription activity

8. Admin Management activity:



10. Admin Management activity

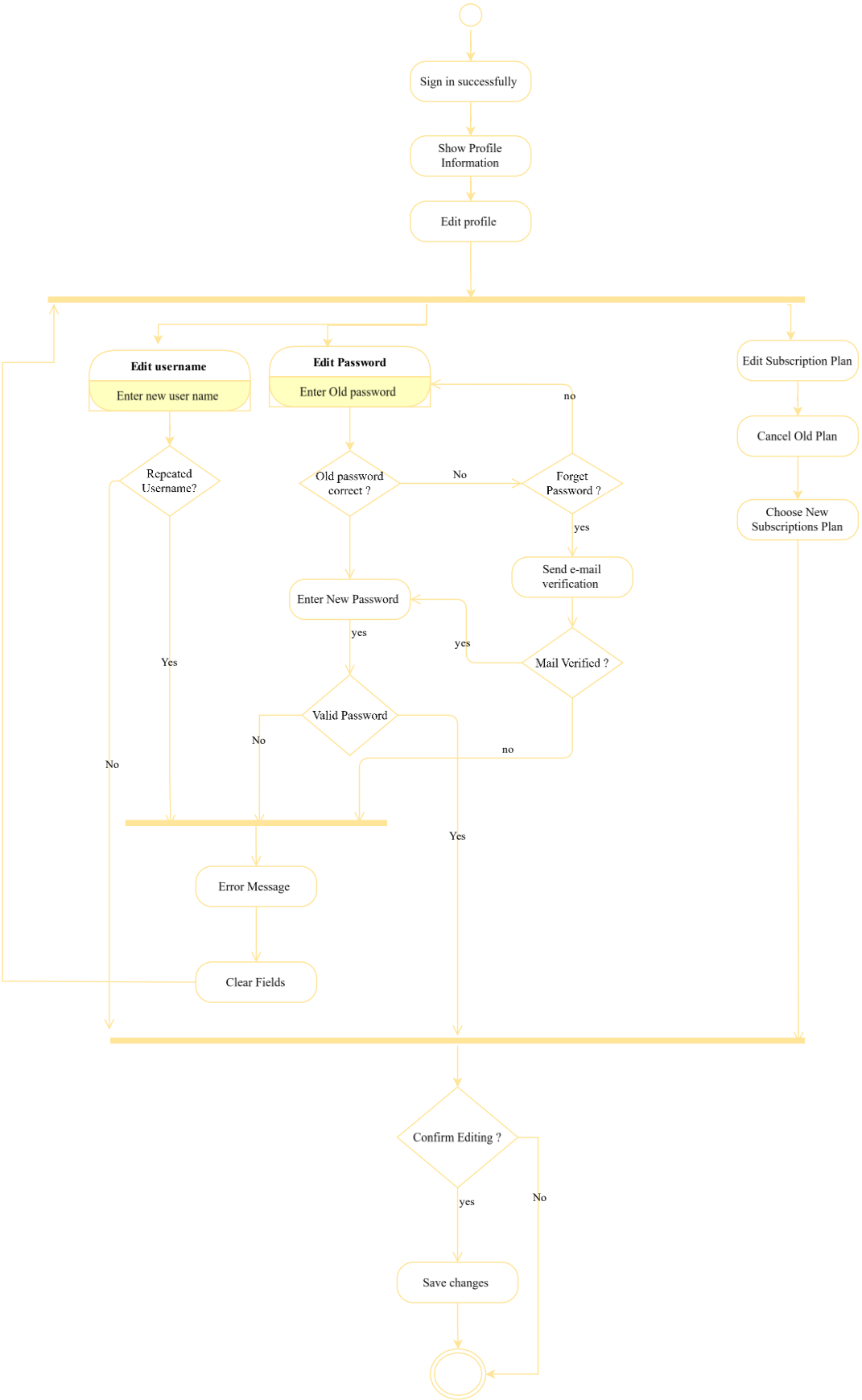
9. Show History activity:



11. Show History activity

3. System Design and Analysis

10. Edit Profile activity:



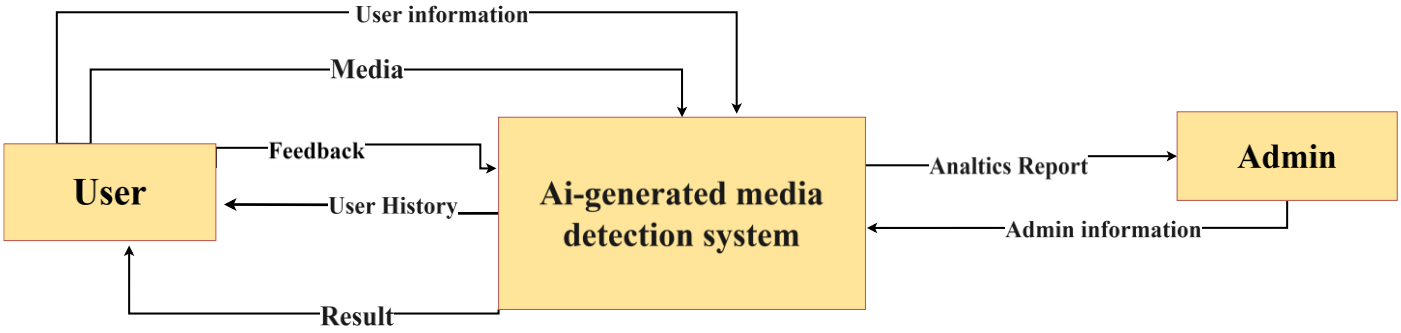
12. Edit Profile activity

3.2.2 Non-Functional Requirements

During the development of the system, several non-functional aspects were considered to ensure a robust and user-friendly application:

1. **Security:** Ensuring secure user authentication, protecting user data, and preventing unauthorized access to the system.
2. **Usability:** Designing an intuitive and user-friendly interface to facilitate easy interaction with the system.
3. **Performance:** Optimizing the system to handle multiple concurrent users and process large media files efficiently.
4. **Scalability:** Designing the system to scale horizontally, allowing for the addition of more servers to handle increased load.
5. **Reliability:** Ensuring the system is reliable and available with minimal downtime, using strategies like load balancing and regular backups.

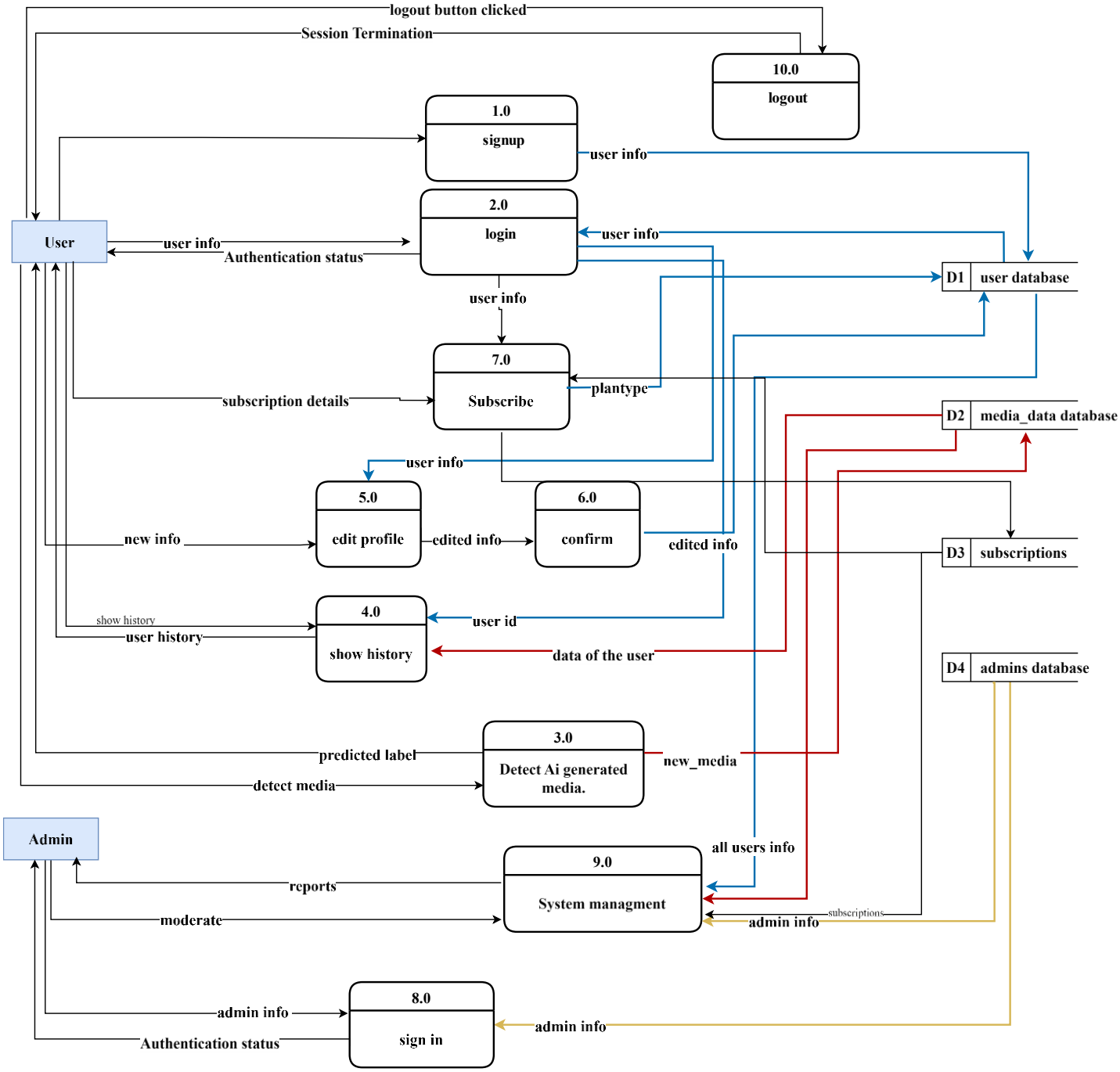
Context Diagram:



13. Context Diagram

3. System Design and Analysis

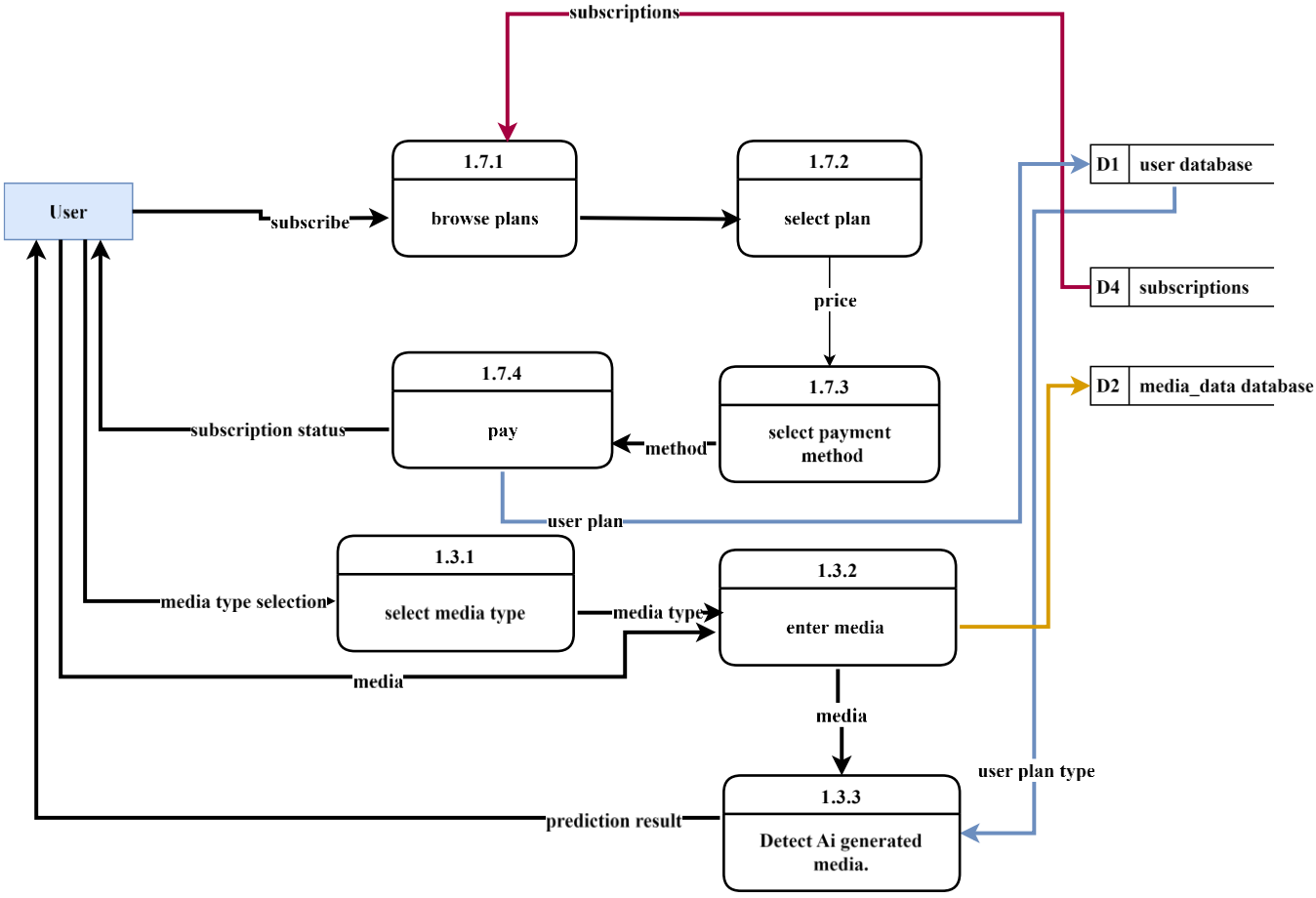
Data Flow Diagram Level 0:



14. Data Flow Diagram L0

3. System Design and Analysis

Data Flow Diagram Level 1:



15. Data Flow Diagram Level 1

3.2.3 Design Patterns

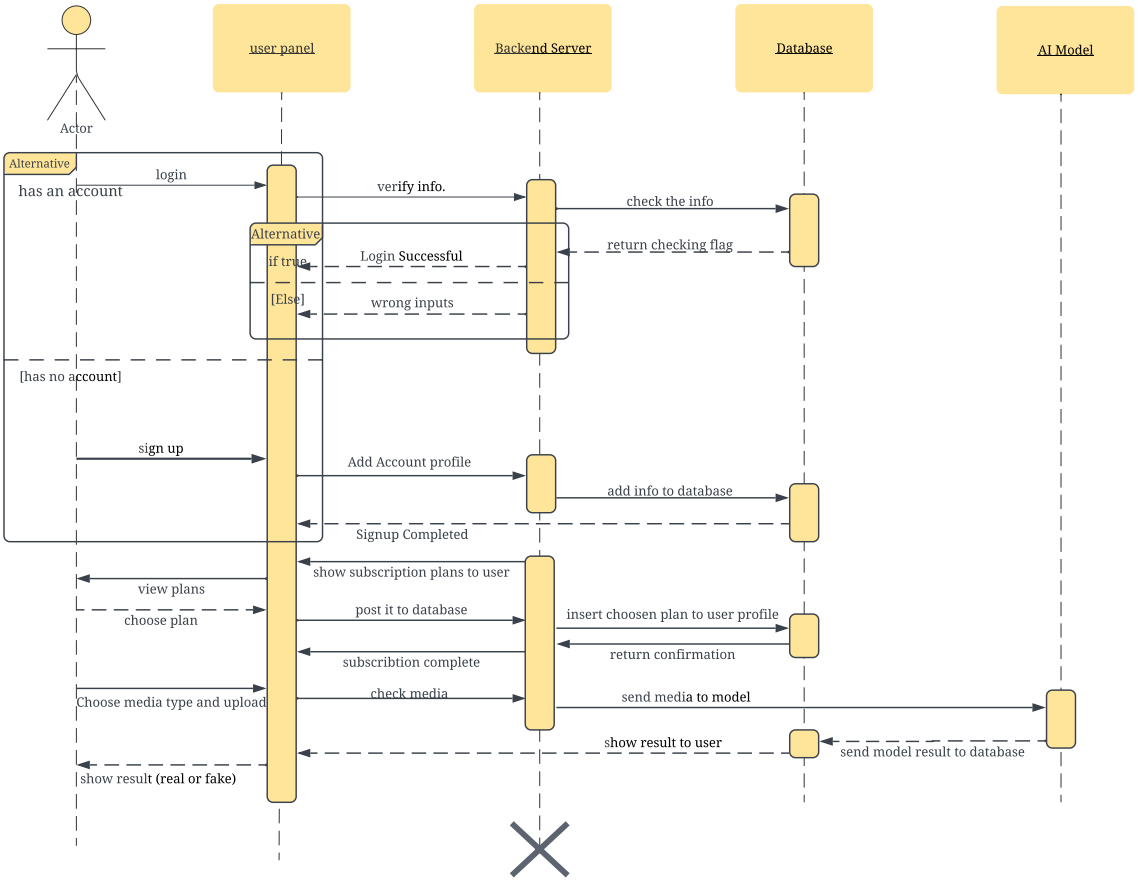
To enhance the system's design and maintainability, several design patterns were implemented:

1. **Model-View-Controller (MVC):** The MVC pattern was used for the overall structure of the system, separating concerns into three main components: Models (data), Views (UI), and Controllers (business logic). This separation improves maintainability and allows for independent development and testing of each component.
    - *Benefit:* Facilitates a clear separation of concerns, making the system easier to manage and extend.
2. **Multimodal Architecture:** This pattern was used to integrate different modalities (image, text, audio) into the system, allowing it to process and analyse various types of media through a unified interface.
    - *Benefit:* Provides flexibility to handle different types of input data using a cohesive architecture, improving extensibility.
3. **Decorator Pattern:** Implemented for flexible processing pipelines, allowing dynamic addition of processing steps for media analysis without modifying the core logic.
    - *Benefit:* Enhances the flexibility and reusability of the media processing pipeline, enabling easy modification and extension of processing steps.
4. **Observer Pattern:** Used for event handling, particularly for real-time notifications and updates, such as informing users about the status of their media analysis.
    - *Benefit:* Decouples event handling from core logic, allowing for scalable and maintainable event-driven architecture.

3. System Design and Analysis

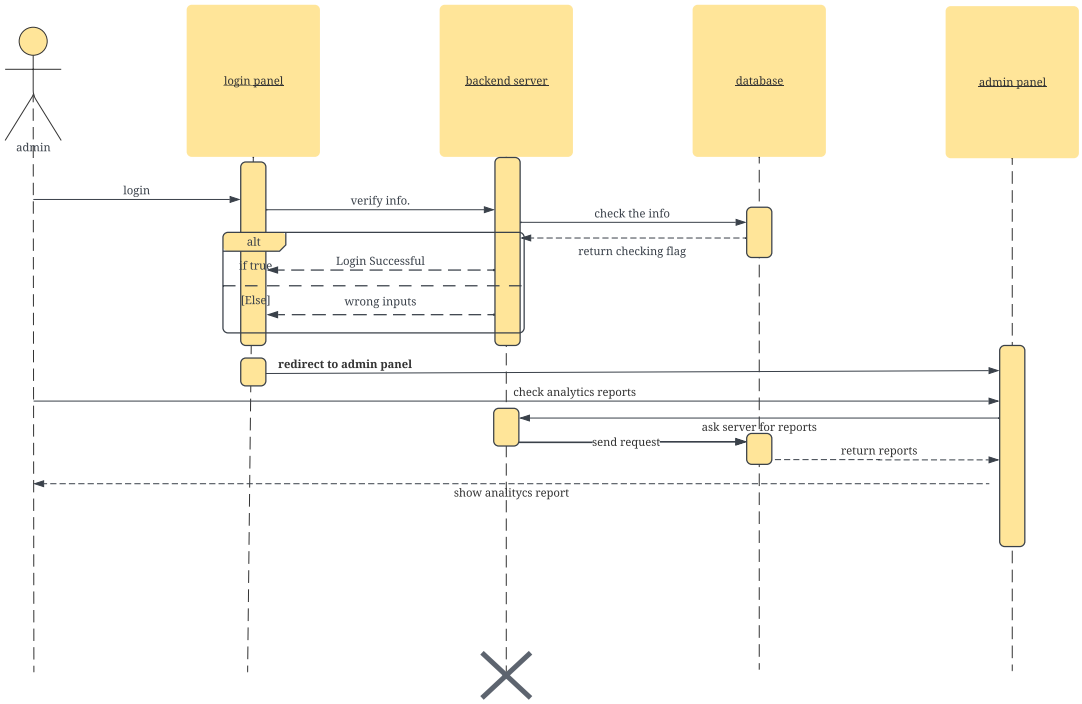
3.2.3.1Sequence Diagram:

1. User



16. User Sequence Diagram

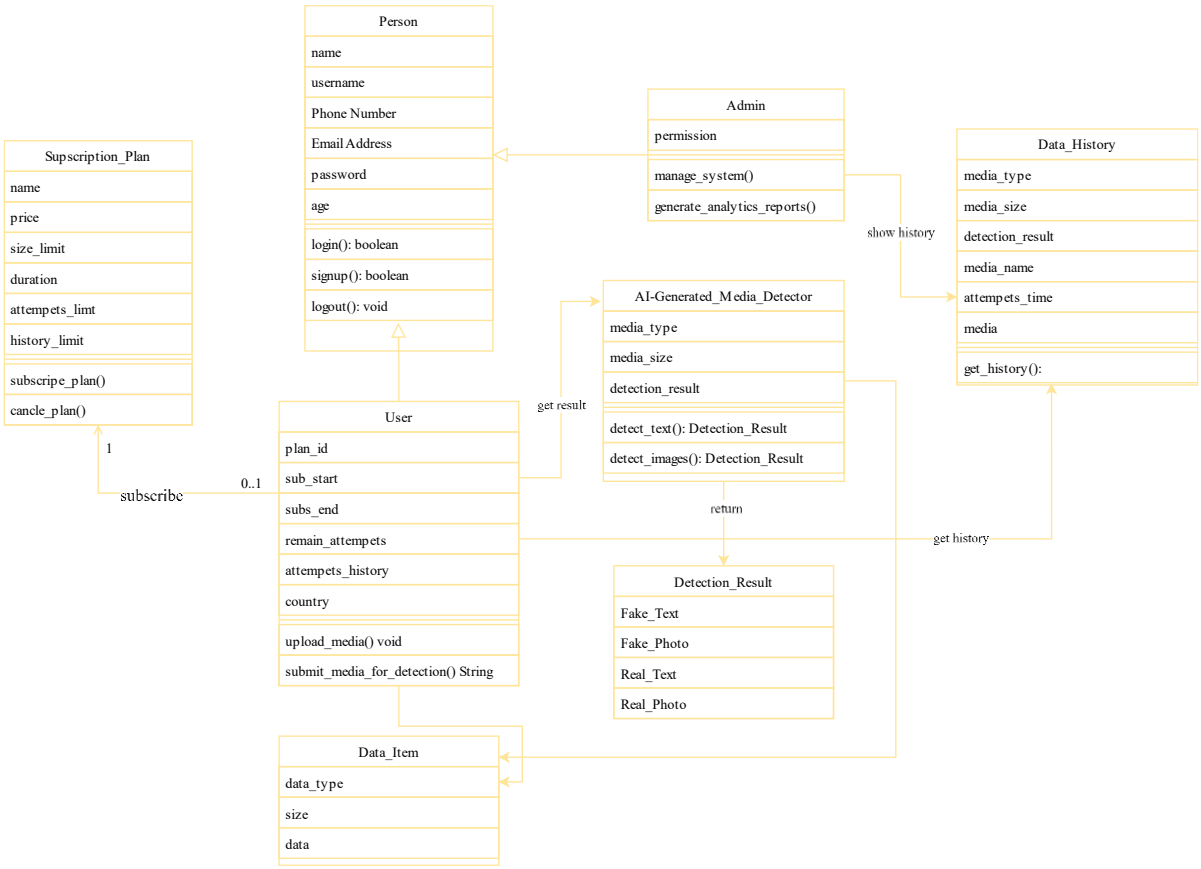
2. Administrator



17. Administrator Sequence Diagram

3. System Design and Analysis

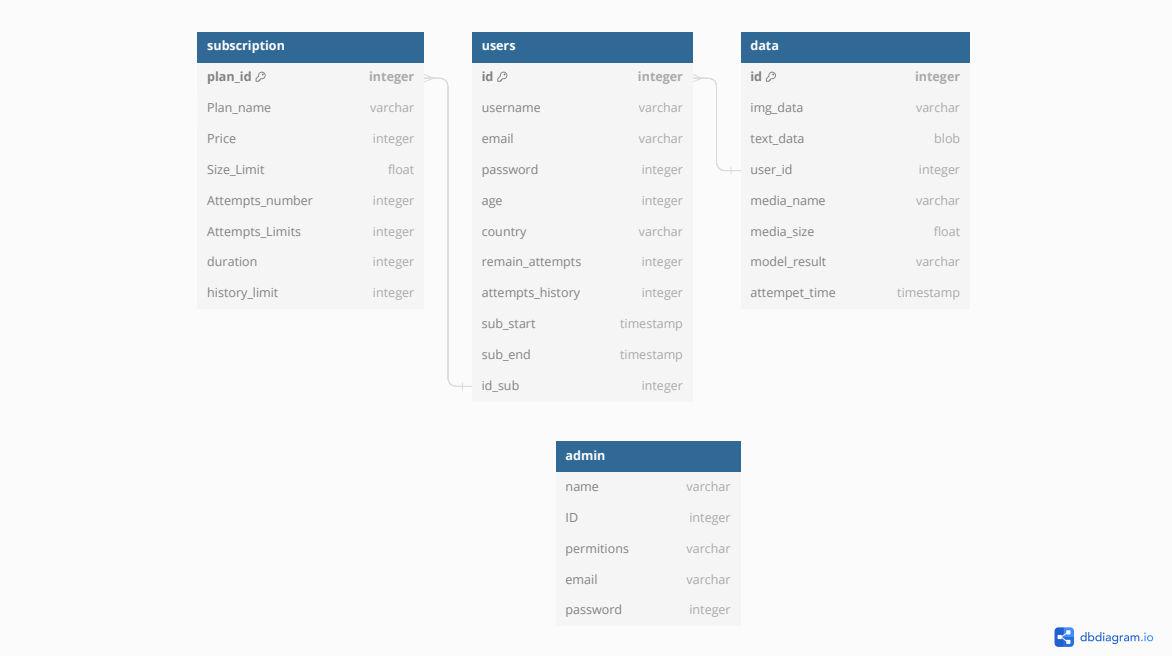
Class Diagram:



18. Class Diagram

3.2.4 Database Design

Database Schema:

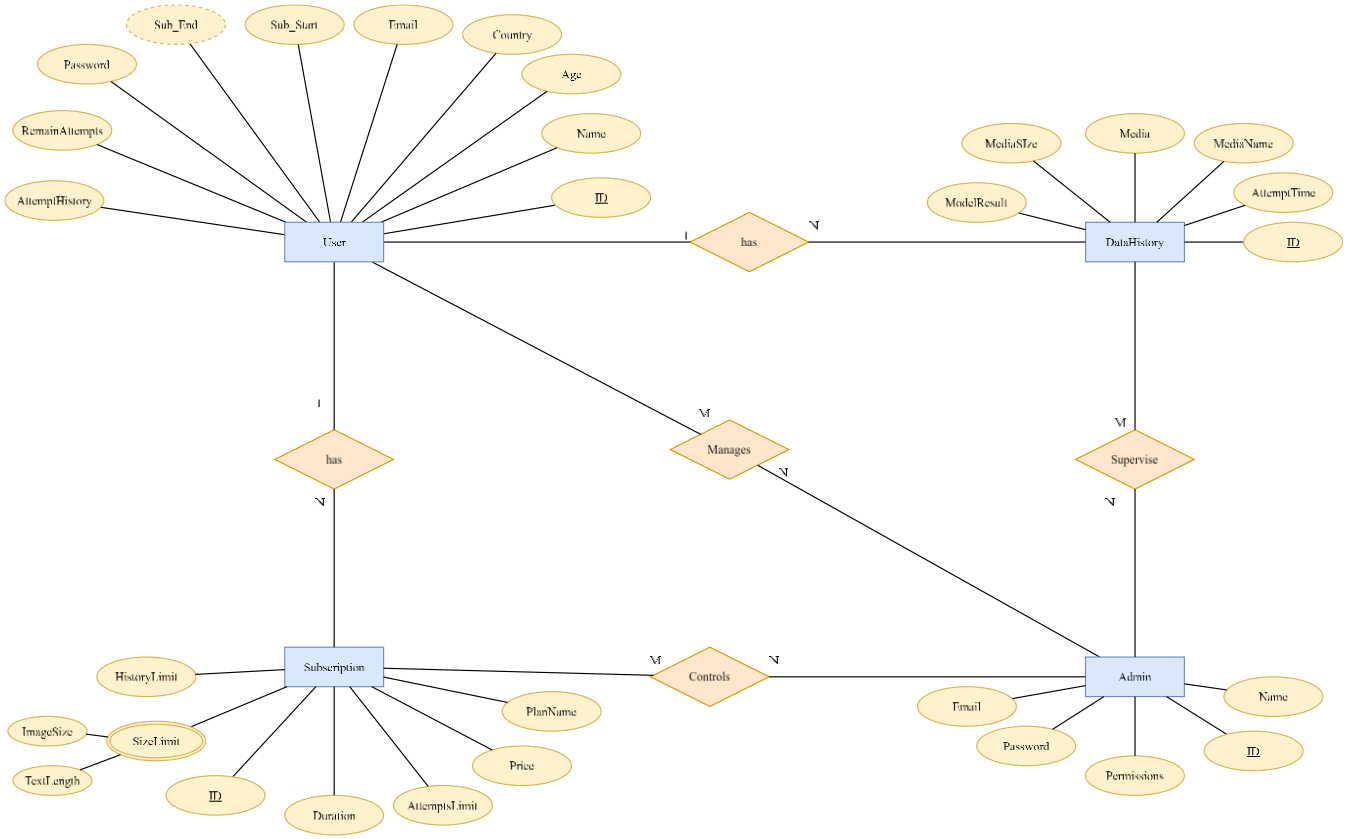


19. Database Schema

3. System Design and Analysis

- **Users Table:** Stores user information, including authentication details.
- **Media Table:** Stores metadata and file paths for submitted media.
- **Subscription Plans Table:** Stores subscriptions plan specifications.
- **Admins Table:** Stores admins credentials and permissions.

ERD Diagram:



20. ERD Diagram

Used Technologies and Tools

- **Frontend:** React, Bootstrap
- **Backend:** Django, Django REST Framework
- **Database:** PostgreSQL
- **Machine Learning:** Hugging Face (RoBERTa, DeBERTa, Wav2Vec2, EfficientNet)

Summary

This chapter outlined the comprehensive system design and analysis for the AI-generated content detection platform. We discussed the overall system architecture, functional and non-functional requirements, and detailed the design patterns used to ensure a robust and maintainable system. Additionally, we included various diagrams to illustrate the system's structure, interactions, and data flow, providing a clear overview of the project's technical foundation.

Chapter Four

4. Methodology