<Real-Time Deepfake Detection for Video Streaming Platforms>

Software Requirements Specification

Version 1.0

Group Id: <F24PROJECT3AEBE>

Supervisor Name :<Sonia Salman>

Revision History

Date (dd/mm/yyyy)	Version	Description	Author
05-Dec-2024	1.0	The project aims to develop a real-time deep-fake detection system that analyzes video streams to detect manipulated content, addressing growing concerns regarding media authenticity, misinformation and identity theft. With the emergence of deep-fake technology, there is a dire need for systems that can distinguish authentic videos from altered ones, particularly in live streaming and video conferencing contexts.	BC220422952

Table of Contents

SRS Document

Scope of Project:

The scope of this project involves the development of a sophisticated , real-time deep-fake detection system that analyze video streams and identify potential manipulation within the context of live streaming, video conferencing platforms, and user-uploaded videos. This cutting-edge system will leverage advanced Deep Learning techniques to analyze video frames towards inconsistencies.
In this web-based application, the institutions, organizations, and even individual users will be able to proactively detect the manipulated content during live sessions or by uploading videos for analysis. The system will provide reliable, fast and efficient detection, ensuring accurate results in real-time. This, in turn, will significantly contribute to enhancing media authenticity and fostering greater public trust.
Moreover, the system will play a crucial role in mitigating risks associated with impersonation and identity theft, which can have far-reaching consequences for individuals and society as a whole.

Functional and Non Functional Requirements:

Functional Requirements:

1. User Authentication and Authorization:

- Users will be able to create accounts by providing necessary information.
- Authenticated users will be able to log in to the system using their credentials.
- Users can recover passwords via password recovery mechanism
- The system will implement role-based access control to restrict access to specific functionalities.
- The system will enforce strong password policies.
- Optional 2FA can be implemented for enhanced security.

2. *Video Input and Processing:*

- The system will support integration with live streaming and video conferencing platforms.
- The users can upload recorded videos in supported formats for analysis.
- The system will validate video file formats and sizes to prevent malicious uploads.
- The system will extract individual frames from input videos for analysis.

3. Deepfake detection:

- The system will utilize advanced deep learning models to process frames in real-time.
- The system will extract relevant features from video frames, such as facial landmarks, eye-movements, lip syncing or motion patterns.
- The system will classify the input video as either fake or real based on the extracted features.

4. *Visualizations and Alerts:*

- The system will also display the detected frames along with a confidence score for each classification.
- The system, in real-time, will notify the users and administrators via alerts when a deepfake content is detected.
- For uploaded videos, the system will provide a detailed report with visualizations of detected deepfake segments.

5. User Feedback:

- Users can give feedback on the accuracy of the system's detection.
- User feedback will be used to improve the performance of the system.

6. Error Handling:

- The system will gracefully handle errors, such as invalid video input, model loading failures, processing delays and network issues.
- The error messages will be informative and user-friendly.

7. User Privacy and Data Handling:

- The system will process video data locally.
- The system will ensure end-to-end encryption to maintain user privacy.
- The system will obtain explicit user consent before processing and storing any video data.

Non-Functional Requirements:

1. Performance:

- The system will have optimized models to achieve real-time detection, especially for live streams.

2. Security:

- The system will implement robust data protection measures.
- The system will have guidelines and policies to ensure ethical use of technology and prevent misuse.

3. Usability:

- The system will have a intuitive and user friendly interface, considering the needs of both technical and non-technical users.
- The system will provide clear and informative feedback to users, including visual indicators and confidence scores.

4. Accessibility:

- The system will be accessible to users as a Web App.

5. Maintainability:

- The system will be designed in modular approach to facilitate updates, maintenance, and future enhancements.

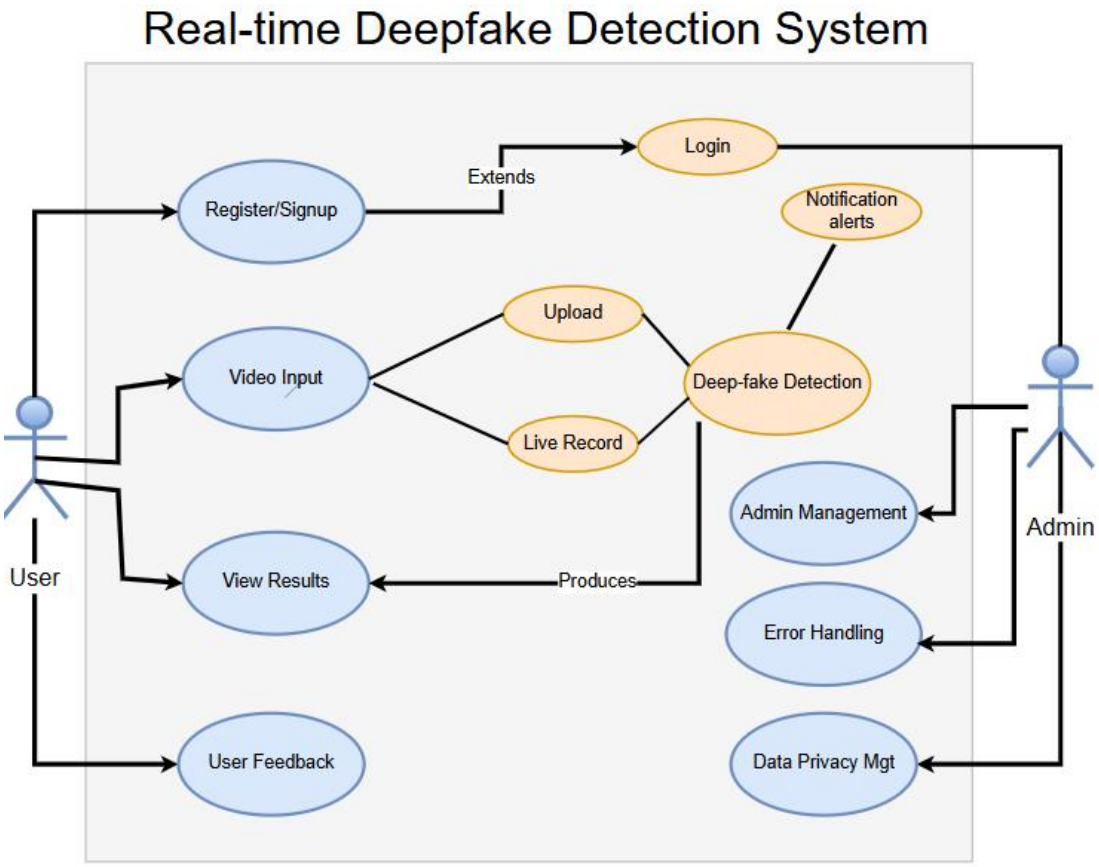
6. Flexibility:

- The system will be flexible to adapt to different video formats and resolutions.

7. Reliability:

- The system will strive to maintain high reliability by employing robust error handling mechanisms.
- The system will be designed to detect and recover from potential failures, ensuring continuous operation and minimizing downtime.

Use Case Diagram(s):



Usage Scenarios:

Use Case Title:	Register/Sign-up
Use Case ID:	UC-01
Actor(s):	User
Actions:	1. User accesses the registration form. 2. User fills the required details. 3. System validates input and sends email verification link. 4. User verifies the email and registration is completed.
Description:	This use case allow users to create account to access the deepfake detection system.
Pre-conditions:	1. User has access to internet. 2. Have a valid email address 3. Web App should be running. 4. Create new account page is displaying properly.
Post-conditions:	A new user account is created.
Exceptions:	1. Email already exists in the system. 2. Failure to send verification email.
Alternative Path:	System detects invalid email, weak password or validation failure and prompts user to correct it before proceeding.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Login
Use Case ID:	UC-02
Actor(s):	User, Admin
Actions:	1. User accesses the login page 2. User enters valid credentials. 3. System authenticates and user is redirected to dashboard.
Description:	This use case allow registered users to log into the platform securely.
Pre-conditions:	1. User has valid credentials. 2. Authentication service is active.
Post-conditions:	User or Admin is successfully logged in.
Exceptions:	Invalid credentials are provided.
Alternative Path:	If user forgets the password, they can use “Forgot Password” option.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Video Input
Use Case ID:	UC-03
Actor(s):	User
Actions:	1. User selects live streaming or uploads a pre-recorded video. 2. For live streaming, user grants access to camera and microphone. 3. For uploads, user selects a file in the valid format and size. 4. System begins to process the video.
Description:	This use case allow users to provide video input for analysis.
Pre-conditions:	1. User is logged in. 2. The device supports live streaming or file upload.
Post-conditions:	Video is submitted successfully for processing.
Exceptions:	Network Error occurs during video input.
Alternative Path:	If camera access is denied or video format is unsupported, the system prompts the required message.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Deepfake detection
Use Case ID:	UC-04
Actor(s):	System
Actions:	1. System processes the video frame by frame and generates detection results with confidence scores. 2. Real-time alerts are triggered for live streams when deepfake content is detected.
Description:	This use case uses advanced deep learning models to detect manipulated content.
Pre-conditions:	1. Video input is done successfully 2. The deepfake detection model is operational.
Post-conditions:	Manipulated content is detected and flagged.
Exceptions:	Network connectivity issues impacts the processing.
Alternative Path:	Processing fails or detection system crashes due to some errors.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	View Results
Use Case ID:	UC-05
Actor(s):	User
Actions:	1. The system generates report highlighting manipulated content. 2. User views results with confidence scores.
Description:	This use case allow users to view analysis results
Pre-conditions:	Detection process has been completed.
Post-conditions:	User can review flagged frames and confidence scores.
Exceptions:	Fail to view results due to system error or unstable network connection.
Alternative Path:	The system recommends re-analysis if results are inconclusive.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	User Feedback
Use Case ID:	UC-06
Actor(s):	User
Actions:	1. User accesses the feedback from after viewing detection results. 2. System stores the feedback for future improvement.
Description:	This use case allow users to provide feedback on the accuracy of detection system.
Pre-conditions:	User had reviewed detection results.
Post-conditions:	Feedback is saved.
Exceptions:	Server issues prevent feedback submission.
Alternative Path:	If system fails, the user is notified and asked to try again later.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Error Handling
Use Case ID:	UC-07
Actor(s):	Admin
Actions:	1. System identifies and logs an error during video processing, detection or result generation. 2. User is informed with a meaningful error message or recommended action.
Description:	This use case manages and responds to errors that occur in the system during operations.
Pre-conditions:	1. The user has initiated a process. 2. Error-handing mechanism is in place.
Post-conditions:	Error is communicated to the user.
Exceptions:	System fails to detect error.
Alternative Path:	System provides corrective guidance if an error occurs.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Admin Management
Use Case ID:	UC-08
Actor(s):	Admin
Actions:	1. Admin logs in using privileged credentials. 2. Admin views and manages user accounts. 3. Admin monitors system logs, performance metrics and user feedbacks.
Description:	This use case allow administrators to manage user accounts, system configurations and flagged content.
Pre-conditions:	1. Admin account is active and properly configured. 2. Admin has access to an authenticated device.
Post-conditions:	User accounts and system data are succeefully managed.
Exceptions:	Admin priviledges are compromised.
Alternative Path:	If an unauthorized user attempts admin access, the system logs the event and generates security alerts.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Data Privacy Management
Use Case ID:	UC-09
Actor(s):	Admin
Actions:	1. System processes video data locally, ensuring end-to-end encryption. 2. User data is saved after consent and deleted upon request.
Description:	This use caseensures privacy of users' data through secure data handling and user consent management.
Pre-conditions:	System is configured to comply with privacy regulations.
Post-conditions:	User's data is processed and managed in accordance with privacy regulations.
Exceptions:	Privacy settings are not applied correctly due to system failure.
Alternative Path:	If there is a breach in privacy settings, the system triggers an immediate alert to both users and admins.
Author:	BC220422952 (F24PROJECT3AEBE)

Adopted Methodology

VU Process Model

Background:

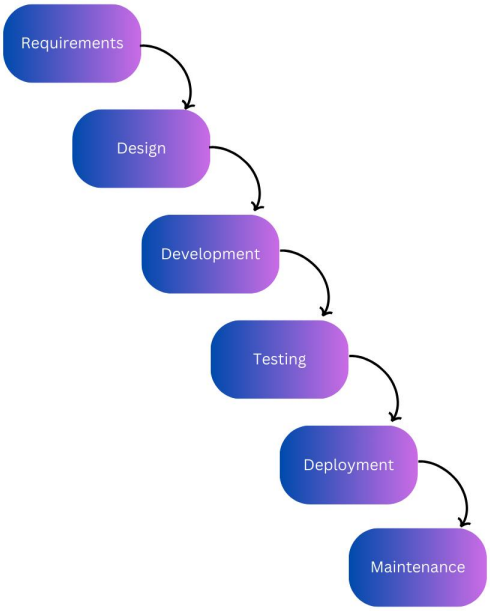
To effectively manage the complexity and dynamic nature of this project, I propose to adopt the "**VU Process Model**", a hybrid approach that combines the structured nature of the Waterfall Model with the iterative and risk-focused principles of the Spiral Model.

Waterfall Model:

The Waterfall model is a linear, sequential design approach where each phase must be completed before the next begins.

Phases of Waterfall Model:

1. Requirements: Define and document system requirements.
2. Design: Create a detailed design of the system architecture.
3. Development: Develop the software components.
4. Testing: Verify and Validate the software.
5. Deployment: Deploy the software to the production environment.
6. Maintenance: Provide ongoing support and updates.

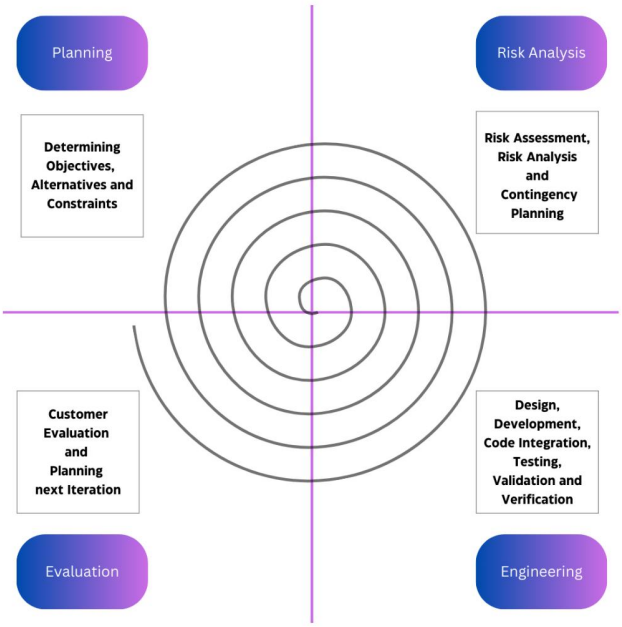


Spiral Model:

The Waterfall model is an iterative development model that emphasizes risk management and customer involvement.

Phases of Waterfall Model:

1. Planning: Define objectives, alternatives and constraints.
2. Risk Analysis: Identify and assess potential risks.
3. Engineering: Develop and test the product.
4. Evaluation: Evaluate the product and plan the next phase.

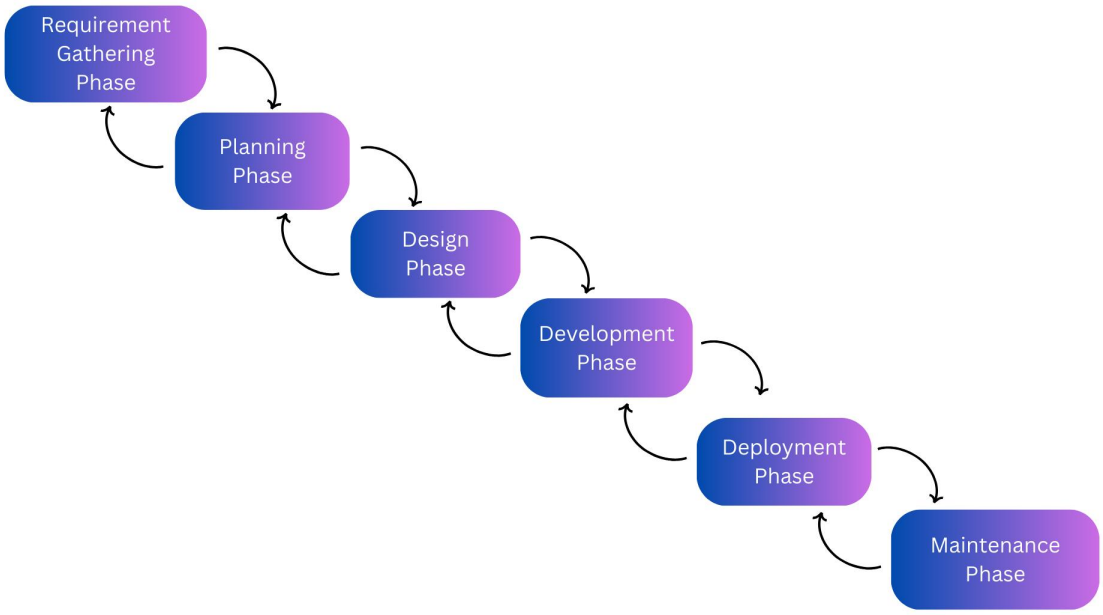


VU Process Model:

The VU Process Model, tailored to the specific needs of this project, combines the strengths of both Waterfall and Spiral models.

Phases of VU Process Model:

1. Requirement Gathering Phase: Detailed requirement elicitation, prioritization and documentation.
2. Planning Phase: Project Planning and Risk Assessment.
3. Design Phase: Detailed system design, Review and Feedback
4. Development Phase: Coding, testing, verification and validation.
5. Deployment Phase: Deployment planning and execution.
6. Maintenance Phase: Post-deployment support



Reasons for choosing VU Process Model:

1. Structured Approach: The waterfall-inspired phases provide a framework for systematically designing and implementing complex deep learning architectures.

2. Iterative Development: The spiral model component allows for rapid prototyping and testing real-time processing algorithms.

3. Continuous Improvement: By iteratively refining the model, we can optimize its performance for real-time application.

4. Risk Management: The spiral model will help identifying and mitigating risks associated with training and deploying large-scale models.

5. Feedback Process: The spiral model will enable us to gather user feedback and incorporate it into design process, resulting into user-friendly interface.

6. Privacy-Preserving Design: The waterfall model will ensure that data privacy and security are considered from the outset of the project.

By leveraging the strengths of both waterfall and spiral model, VU Process Model provides a robust framework for developing and deploying a reliable and efficient deepfake setection system.

Work Plan

WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION
1	**SRS Document**	**11/12/2024**	**12/8/2024**	26
1.1	Scope	11/12/2024	11/12/2024	0
1.2	Functional Requirements	11/13/2024	11/14/2024	1
1.3	Non Functional Requirements	11/15/2024	11/15/2024	0
1.4	Usecase Diagram	11/16/2024	11/18/2024	2
1.5	Usage Scenarios	11/19/2024	11/24/2024	5
1.6	Adopted Methodology	11/25/2024	11/30/2024	5
1.7	Work Plan	12/1/2024	12/3/2024	2
2	**Design Document**	**12/9/2024**	**3/3/2025**	84
2.1	Introduction	12/21/2024	12/24/2024	3
2.2	ERD	12/25/2024	12/31/2024	6
2.3	Sequence Diagram	1/1/2025	1/11/2025	10
2.4	Architecture Design Diagram	1/12/2025	1/22/2025	10
2.5	Class Diagram	1/23/2025	1/31/2025	8
2.6	Database Diagram	2/1/2025	2/6/2025	5
2.7	Interface Diagram	2/7/2025	2/19/2025	12
2.8	Test Cases	2/20/2025	2/28/2025	8
3	**Prototype**	**3/4/2025**	**3/17/2025**	13
3.1	Problem Statement	3/4/2025	3/5/2025	1
3.2	Design	3/6/2025	3/8/2025	2
3.3	Development	3/9/2025	3/15/2025	6
3.4	Final Prototype	3/16/2025	3/17/2025	1
4	**Final**	**3/18/2025**	**5/9/2025**	51
4.1	Model Design	3/18/2025	3/23/2025	5
4.2	Web App Layout	3/24/2025	3/27/2025	3
4.3	Web App Development	3/28/2025	3/31/2025	3
4.4	Coding	4/1/2025	4/15/2025	14
4.5	Testing	4/16/2025	4/23/2025	7
4.6	Final Development	4/24/2025	4/30/2025	6
4.7	Final Report	5/1/2025	5/9/2025	8