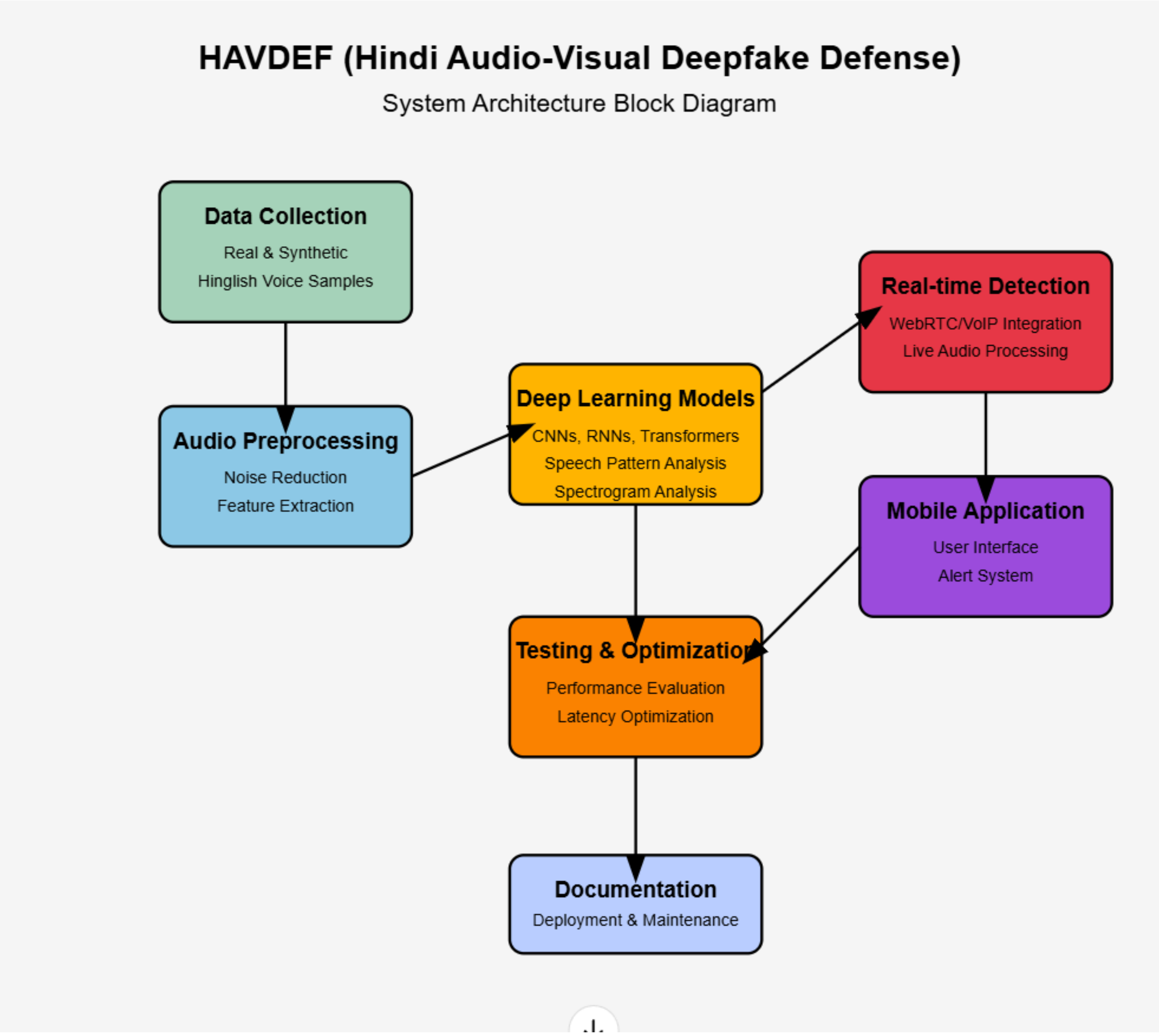
CAPSTONE PROJECT

HAVDEF (Hindi Audio-Visual Deepfake Defense)

THAPAR INSTITUT
OF ENGINEERING & TECHNOLOG
(Deemed to be University)

(102203191) Shivane Kapoor
(102203194) Kaustubh Singh
(102203205) Japneet Singh
(102203499) Arpit Jain
(102253002) Diwakar Narayan Sood

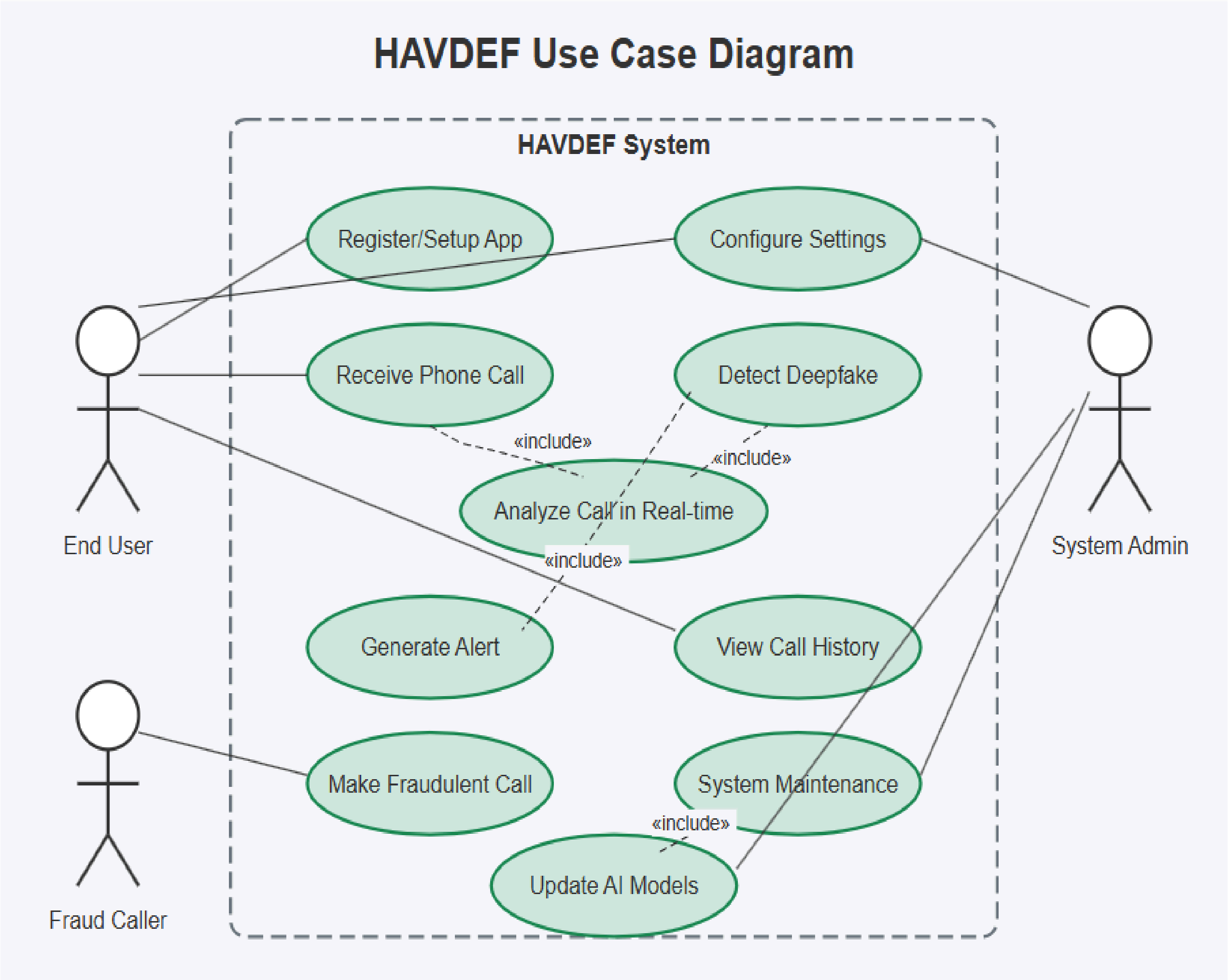
Block Diagram of Work Flow



HAVDEF (Hindi Audio-Visual Deepfake Defense)

System Architecture Block Diagram

Use Case Diagram



Use - Case Template

Field	Details
Use Case Name	Detect Deepfake
Actors	Primary: End User Secondary: System Admin, Fraud Caller
Description	This use case allows the End User to register and receive phone calls. The system detects deepfakes during the call and analyse the conversation in real-time to ensure security and detect fraudulent activity. It also generates alerts if necessary. The System Admin is responsible for updating AI models and maintaining the system.
Preconditions	- End User must have the app registered and set up. - System Admin must have proper credentials to access system maintenance functionalities. - The user must be logged into the system. - The system must have access to the internet for real-time analysis and detection.
Basic Flow (Main Scenario)	1. The End User registers and sets up the app. 2. The End User receives a phone call. 3. The system analyses the call in real-time for deepfake detection.

	4. If a deepfake is detected, the system alerts the End User and generates an alert for the Fraud Caller. 5. The system continues analysing the call and detects any fraudulent activities or threats. 6. System Admin can review and maintain the system, including updates and configuration.
Alternative Flows (Exceptions & Variations)	- If the system detects a deepfake during the call, it will flag the call for further action and notify the End User. - If the End User tries to make a fraudulent call, the system will block it and generate an alert. - If a technical error occurs, the system will display an error message and suggest troubleshooting steps.
Includes & Extends	- This use case includes "Analyse Call in Real-time" - This use case extends "Generate Alert". - This use case includes "View Call History". - This use case includes "Update AI Models".
Triggers	- End User registers and sets up the app. - Fraud Caller attempts to make a fraudulent call. - System Admin starts system maintenance.

Business Rules	- The system must detect fraudulent calls and deepfake activities in real-time. - Alerts must be generated in case of fraudulent activities or deepfake detection. - The system must update AI models periodically. - Call history must be stored and available for review.
Post-conditions	- Deepfake detection results are stored in the system database. - Alerts are generated if fraudulent activity is detected. - System Admin performs necessary system maintenance and updates.
Non-functional Requirements	- The system must handle multiple concurrent users without performance degradation. - The system must detect deepfakes and fraudulent calls in under 5 seconds in real-time. - The system must be highly available with minimal downtime for maintenance.
Notes	- This use case is critical for preventing fraudulent activity and maintaining the integrity of phone calls. - Continuous improvement of the AI models is essential to accurately detect deepfakes and fraudulent behaviour.

Set of Tasks:

1.System Setup & Authentication

- User registration and login
- Role-based access and permission management
- Profile management and user-specific settings

2. Audio-Visual Deepfake Detection

- Collect and preprocess multilingual and accent-rich datasets
- Develop and integrate AI models for deepfake detection (speech and video)
- Implement real-time audio-visual analysis and fraud detection
- Test and optimize models for different phonetic and linguistic challenges

3. Fraud Detection & Reporting

- Assess severity and calculate fraud detection accuracy
- Generate alerts in case of detected deepfakes or fraudulent calls
- Export reports on detection results, including timestamps and analysis data

4. User Management & Administration

- Manage users, roles, and permissions (admin and end users)
- System log tracking and user activity management
- Admin tools for overseeing system performance and handling detected fraud

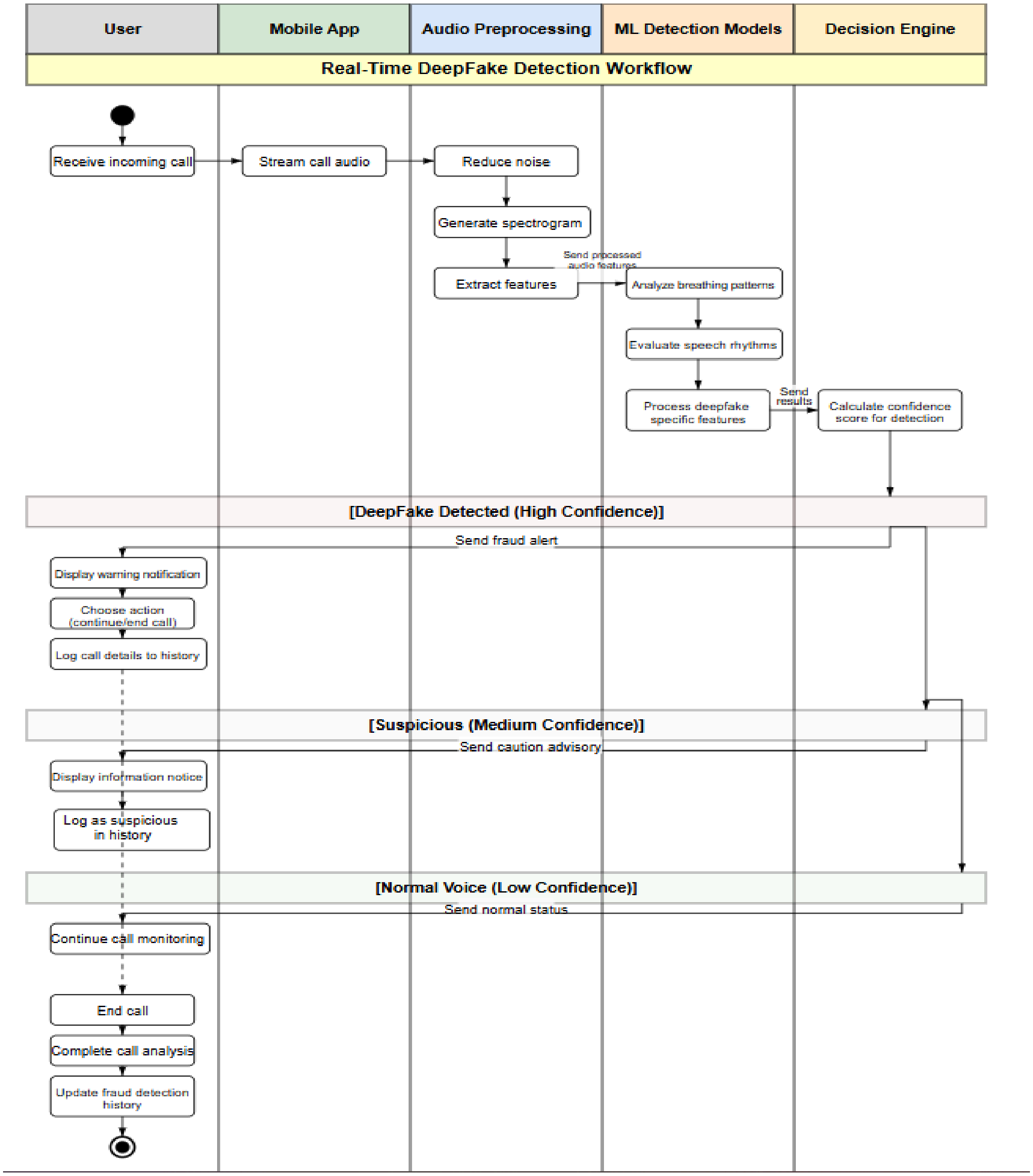
5. Data Analytics & Predictions

- Analyze historical data to predict trends in fraud calls
- Predict potential threats based on call behavior and patterns
- Provide a dashboard with analytics and visual insights for both end users and system admins

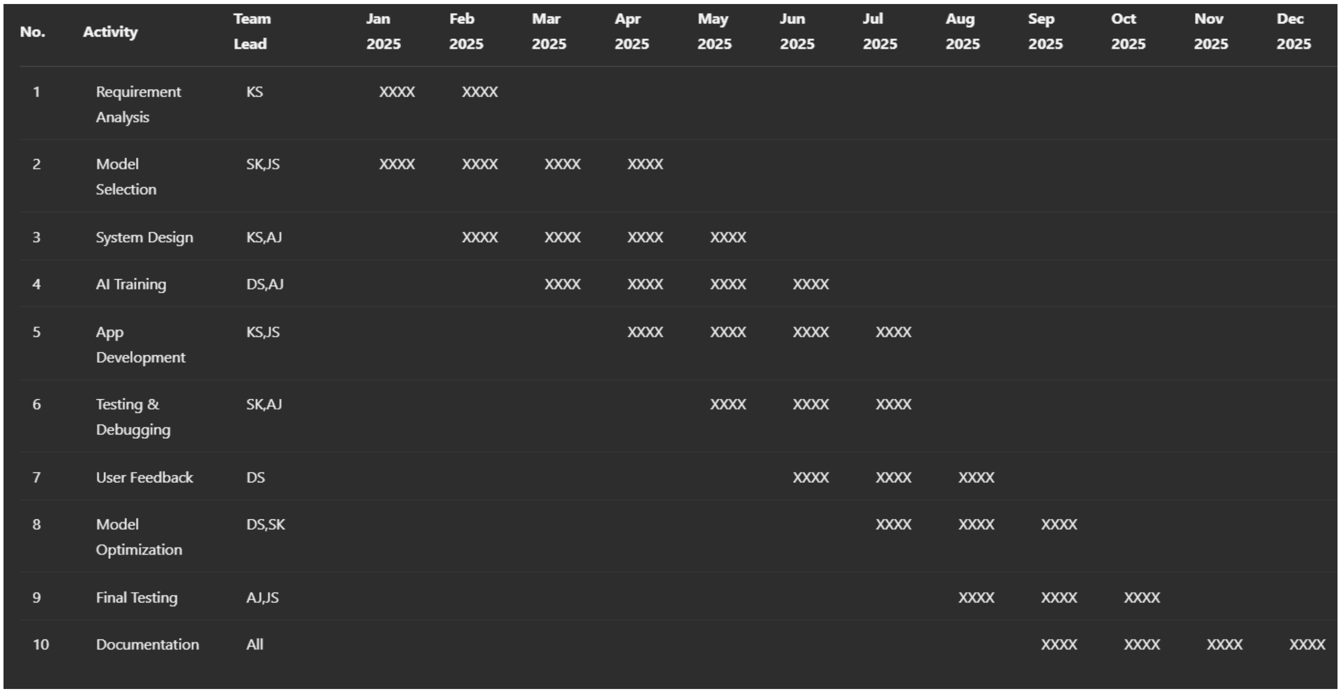
6. Deployment & Integration

- Cloud setup for system deployment and scalability
- Integrate APIs for external systems (payment, notifications, etc.)
- Seamless deployment and system monitoring tools for performance tracking

Activity/ Swimlane Diagram


User
Mobile App
Audio Preprocessing
ML Detection Models
Decision Engine
Real-Time DeepFake Detection Workflow
Receive incoming call
Stream call audio
Reduce noise
Generate spectrogram
Extract features
Send processed audio features
Analyze breathing patterns
Evaluate speech rhythms
Process deepfake specific features
Send results
Calculate confidence score for detection
[DeepFake Detected (High Confidence)]
Send fraud alert
Display warning notification
Choose action (continue/end call)
Log call details to history
[Suspicious (Medium Confidence)]
Send caution advisory
Display information notice
Log as suspicious in history
[Normal Voice (Low Confidence)]
Send normal status
Continue call monitoring
End call
Complete call analysis
Update fraud detection history

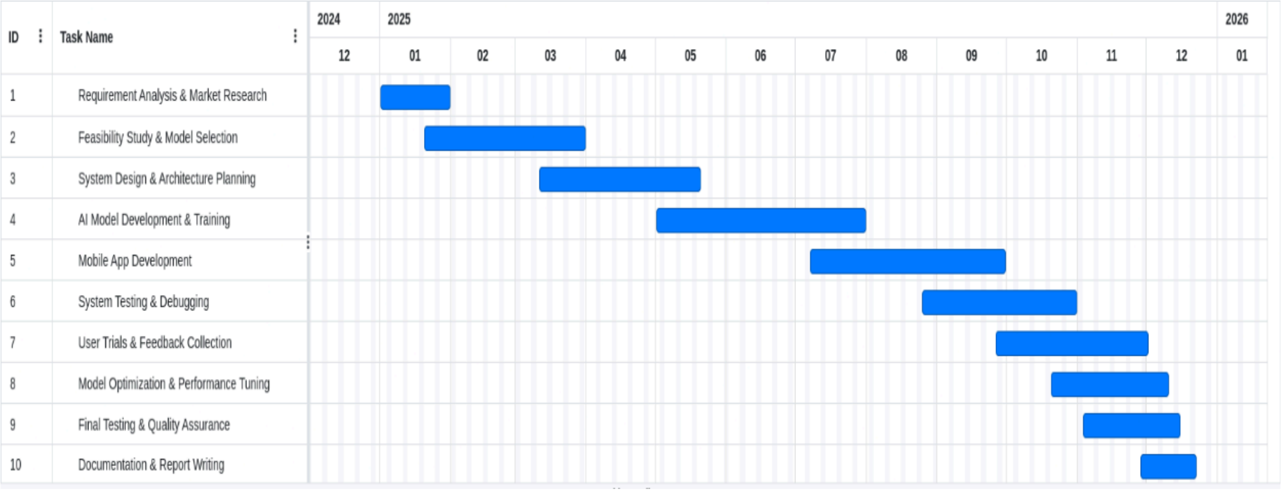
Work Breakdown Structure



No.	Activity	Team Lead	Jan 2025	Feb 2025	Mar 2025	Apr 2025	May 2025	Jun 2025	Jul 2025	Aug 2025	Sep 2025	Oct 2025	Nov 2025	Dec 2025
1	Requirement Analysis	KS	XXXX	XXXX										
2	Model Selection	SK,JS	XXXX	XXXX	XXXX	XXXX								
3	System Design	KS,AJ		XXXX	XXXX	XXXX	XXXX							
4	AI Training	DS,AJ			XXXX	XXXX	XXXX	XXXX						
5	App Development	KS,JS				XXXX	XXXX	XXXX	XXXX					
6	Testing & Debugging	SK,AJ					XXXX	XXXX	XXXX					
7	User Feedback	DS						XXXX	XXXX	XXXX				
8	Model Optimization	DS,SK							XXXX	XXXX	XXXX			
9	Final Testing	AJ,JS								XXXX	XXXX	XXXX		
10	Documentation	All									XXXX	XXXX	XXXX	XXXX

Legend:
- KS: Kaustubh Singh
- SK: Shivane Kapoor
- JS: Japneet Singh
- AJ: Arpit Jain
- DS: Diwakar Sood

Gantt Chart



ID	Task Name
1	Requirement Analysis & Market Research
2	Feasibility Study & Model Selection
3	System Design & Architecture Planning
4	AI Model Development & Training
5	Mobile App Development
6	System Testing & Debugging
7	User Trials & Feedback Collection
8	Model Optimization & Performance Tuning
9	Final Testing & Quality Assurance
10	Documentation & Report Writing

Functional Requirements (What the system must do)

1. **User Authentication & Access Control**
    - Secure login for users.
    - Role-based access for different user levels.

2. **Real-Time Call Monitoring & Deepfake Detection**
    - Analyze live phone calls for deepfake voices.
    - Detect AI-generated fraud in Hinglish conversations.

3. **Audio-Visual Analysis**
    - Process voice data to identify synthetic speech patterns.
    - Analyze video for deepfake detection in video calls.

4. **Alert Mechanism**
    - Notify users instantly if deepfake fraud is detected.
    - Provide confidence scores for detection accuracy.

5. **Model Training & Updates**
    - Improve detection accuracy using real-world Hinglish datasets.
    - Enable continuous learning and model retraining.

6. **User Feedback Collection**
    - Allow users to report false positives or negatives.
    - Store flagged calls for further model refinement.

7. **Report Generation & History Logs**
    - Generate reports summarizing detected deepfake cases.
    - Maintain logs of analyzed calls for reference.

Non-Functional Requirements (Quality attributes of the system)

1. **Performance**
    - Analyze calls and provide results within 2-5 seconds.

2. **Scalability**
    - Support multiple concurrent users without lag.

3. **Usability**
    - Intuitive UI for seamless real-time fraud detection.

4. **Security & Privacy**
    - Encrypt call data to protect user privacy.
    - Secure cloud storage for detected fraud cases.

5. **Reliability**
    - Ensure >90% accuracy in detecting AI-generated voices.
    - Handle diverse accents and noisy environments.

6. **Maintainability**
    - Modular codebase for easy debugging and updates.

7. **Compatibility**
    - Accessible via mobile and web apps across various devices.