AegisSecure: AI-Powered Threat Detection App

Course Project: Mid-Evaluation Submission

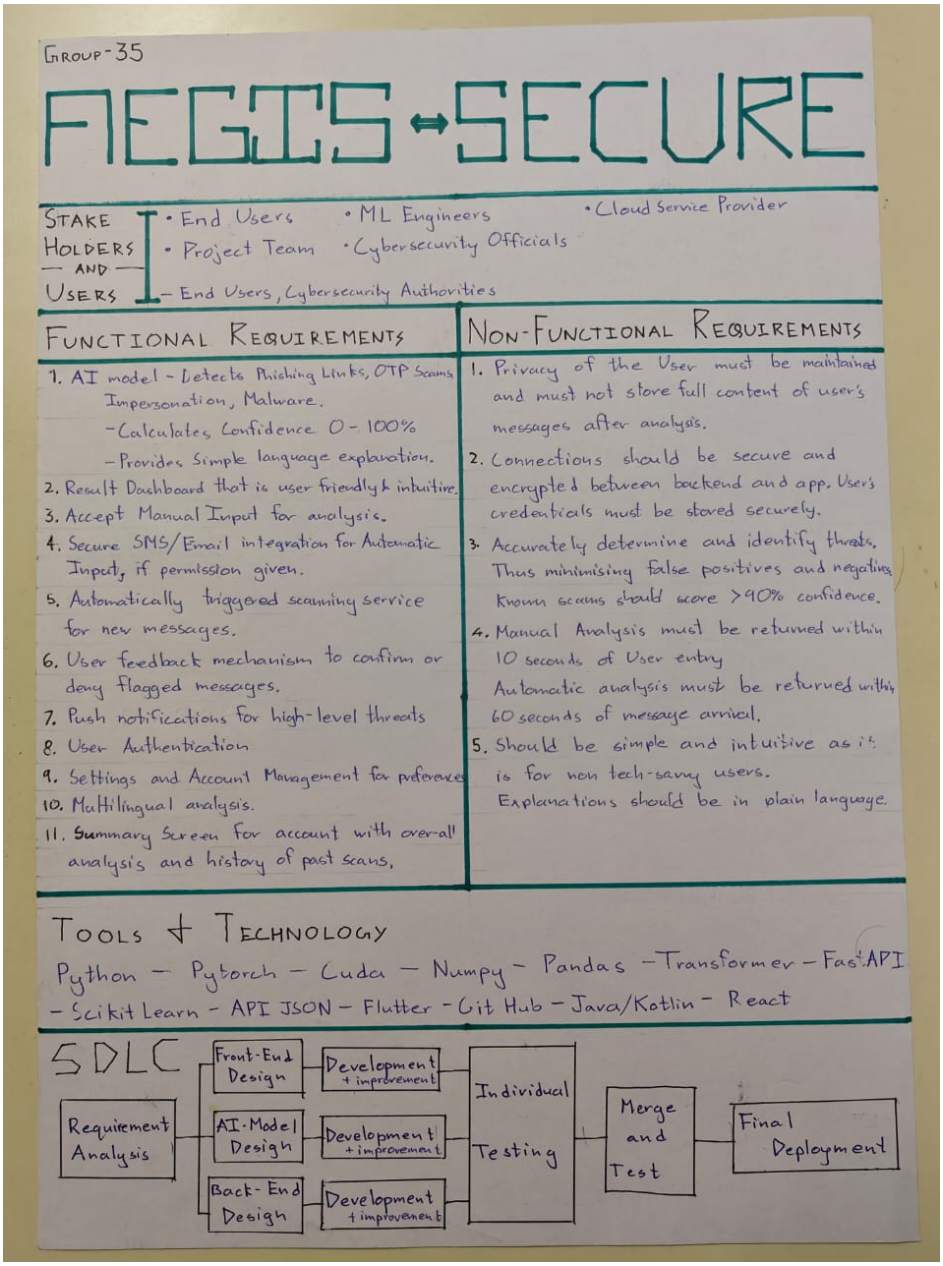
Group Number: 35
Meevada Soham Meghalkumar [202301484]
Gohil Suryadeepsinh Hardevsinh [202301463]
Rana Neelabh Vijaykumar [202301476]
Hrithik B Patel [202301441]
Vadsmiya Pransu Pradipkumar [202301445]
Dhruv Jigneshkumar Patel [202301095]
Bhagiya Jenish Rameshbhai [202301480]
Akshat Bhatt [202301460]
Vrajkumar Makwana [202301436]
Chavda Mihirsinh Labhubhai [202301479]

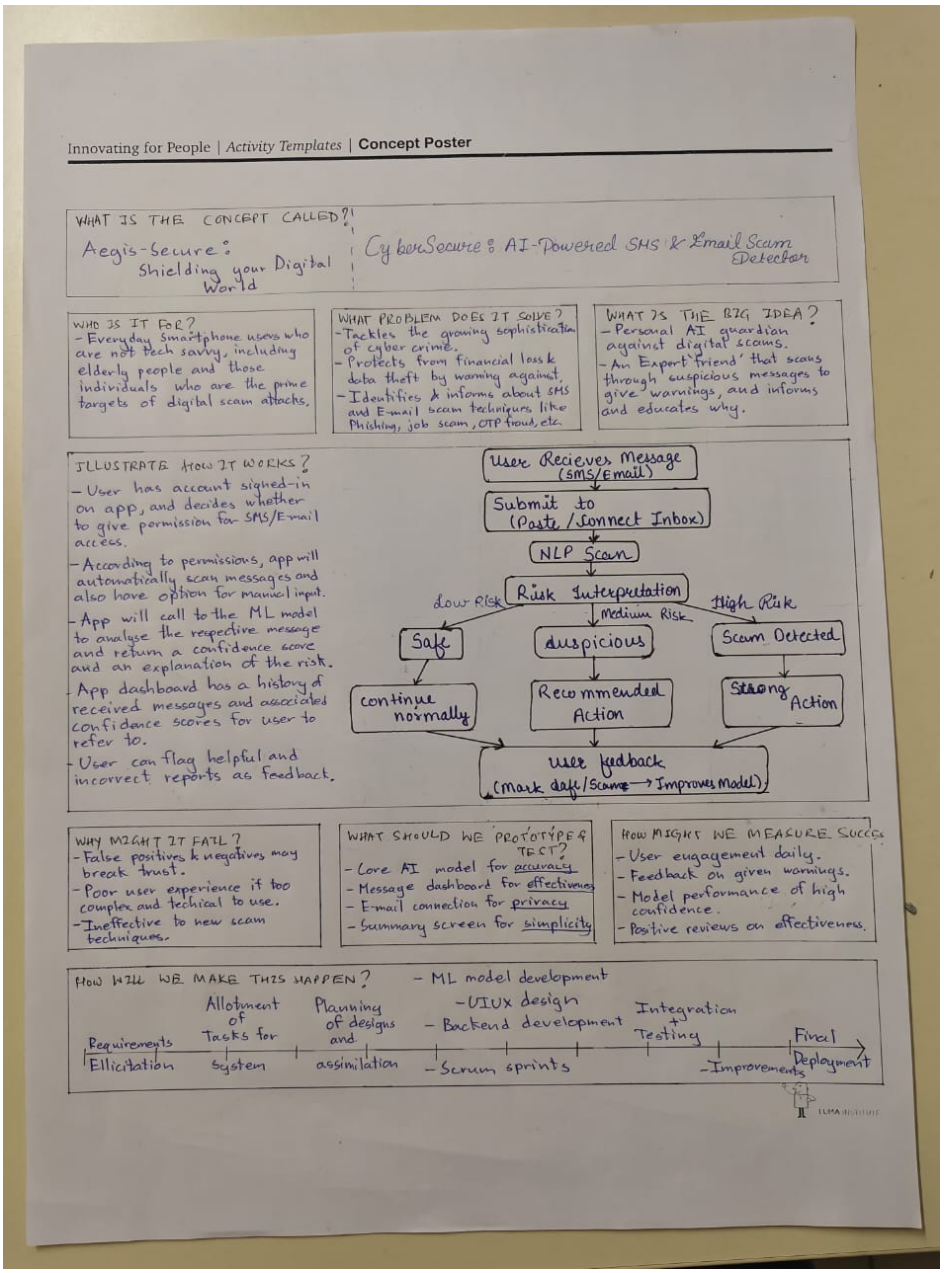
September 11, 2025

IT314 - Software Engineering
Prof: Saurabh Tiwari
Mentor: Shyam Patel

Contents

1 Concept Poster and Chart Paper





2 Requirements Elicitation

We had a vague idea of what was expected from such a cyber security application, but to find out what exactly our userbase would be, and how we can best help them, we employed multiple types of requirement elicitation techniques. Raw data received from the respective elicitation techniques can be found in the GitHub repository. The following were chosen:

2.1 Brainstorming

Why it was used: As a student team, we ourself had to establish our primary idea and vision. We decided a basis of features upon which we would build based upon the new

information that would be revealed to us. We also had to establish what we could do based on our own capabilities.

How it was implemented: We held multiple meetings throughout the weeks given to us, where all group members would be present. We would share our ideas and personal ares of skill and interest.

How it helped: Brainstorming sessions allowed us to collaboratively generate the initial idea. It served as the foundation for our project, enabling us to define a clear scope for our application.

2.2 Survey Form

Why it was used: We needed to gather quantitative data to validate our core assumptions about the problem space. The survey would allow us to gather information from a broad source, in a manner of minimum effort.

How it was implemented: A google form was created with thirteen questions to verify various aspects of our project. This way we could tell which features would be valued. It also revealed which problems are actually faced by our prospective users that we should target.

How it helped: The survey provided crucial data points on how concerned the general population is about scams, what features they value most, and what their primary concerns are regarding performance and privacy. This helped us create our Functional and Non-Functional Requirements. We were able to gather data from different types of audiences.

2.3 Focus Groups

Why it was used: To complement the data from the survey, we needed to gather more insights to understand the "why" behind user preferences. A survey form has definite questions. By holding focus groups we would be able to retrieve open feedback and discussions that our audience would volunteer.

How it was implemented: Four focus group discussions were held, each targeting different demographics to help deeper understand the requirements, and get information on preferences that users would have when using such an app. Volunteers were allowed to put forward their opinions on the target of the discussion, and then inferences were made based on how we should tackle the problems.

How it helped: The focus groups were instrumental in uncovering nuanced requirements that a survey could not. They revealed the demand for post-detection guidance, community sharing features, and the critical importance of a modern UI for younger users. This technique allowed us to add depth and empathy to our requirements, ensuring the final product is not just functional, but also user-centric.

2.4 Interviews

Why it was used: The interviews would be more specific to target questions that our group members faced.

How it was implemented: Two interviews were held with people familiar with topics of cyber security. Their valuable inputs were recorded.

How it helped: The interviews helped us realise what exact type of issues that our app should be able to handle. This was vital as many of us were not well informed on such a topic.

2.5 Research & Documentation Review

Why it was used: For the practical realisation of our project, we needed to know exactly what types of implementation constraints we would face. Not only that, but it supported many different parts of our discussions.

How it was implemented: Within our group meetings, members would identify parts where we knew what to do, but were unclear exactly how it would be done. This would prompt us to look online to find such data.

How it helped: Research helped us identify ways in which such problems have been handled. It gave us information on stakeholders and how they are tied into our project. It also made clear to us what would be expected of us and set up the paths we need to take to make our project a reality.

3 Identification of Stakeholders and Users – Elicitation Techniques Used

User Stakeholder	Elicitation Techniques Used
Individual User	Survey and Focus Groups
Email/SMS Service Providers	Documentation Review
Cloud Service Provider	Documentation Review
Dataset Providers	Documentation Review
Cybersecurity Experts	Interviews/Surveys
API Key Providers	Documentation Review
Government Agencies/Regulators	Documentation Review
Developer Team (Students + Mentor)	Brainstorming Sessions
Database Providers	Documentation Analysis

Table 1: Stakeholders and Elicitation Techniques

- **Individual User**
  Application: Conducted surveys by sharing a Google form and performed focus groups to get valuable feedback.
  Justification: Direct feedback ensures accurate capture of user needs.

- **Email/SMS Service Providers**
  Application: Reviewed technical documentation and developer guidelines to identify integration constraints, permissions, and compliance requirements.
  Justification: Ensures app integration is compliant and feasible.

- **Cloud Service Provider**
  Application: Reviewed CSP manuals and pricing to assess scalability, security, and cost; performed a comparative study of AWS, Google Cloud, Microsoft Azure, and Salesforce.
  Justification: Helps select the best platform for reliability and cost-effectiveness.

- **Dataset Providers**
  Application: Collected metadata, dataset structures (CSV, JSON, Parquet), and compliance policies (GDPR, CCPA, copyright/licensing rules).
  Justification: Ensures legal, structured, and usable datasets for the app.

- **Cybersecurity Experts**
  Application: Conducted interviews to validate detection logic and confirm scam indicators match real-world patterns.
  Justification: Provides domain validation of detection methods.

- **API Key Providers**
  Application: Reviewed API guidelines, authentication procedures, and integration

manuals.
Justification: Ensures secure and proper API integration.

- **Government Agencies/Regulators**
  Application: Referred to official guidelines, legal frameworks, and compliance documents.
  Justification: Guarantees compliance with laws and regulations.

- **Developer Team (Students + Mentor)**
  Application: Conducted group discussions to define requirements, align goals, and explore implementation strategies.
  Justification: Aligns team understanding and implementation approach.

- **Database Providers**
  Application: Reviewed database documentation to assess metadata, features, performance, and security/compliance policies.
  Justification: Ensures scalable, secure, and suitable data storage.

4 Requirement Identification (FRs and NFRs)

4.1 Identified Functional Requirements (FRs)

4.1.1 FR-001 — Scam Detection

Description: Analyze input (URLs / message text) and classify as Safe or Scam, returning a confidence score (0–100) and a one-line plain-language explanation.

Elicitation Techniques Used: Interviews, Brainstorming.

How the Techniques Produced the Requirement: Interviews gave real scam examples and user demand for clear explanations; surveys confirmed broad need for automated detection and confidence scores; brainstorming translated these into a concrete output format (label, confidence, explanation).

4.1.2 FR-002 — Result Dashboard

Description: Provide a dashboard listing recent scans with timestamp, risk category (Low/Med/High), confidence score, and brief explanation; support basic sorting and filtering.

Elicitation Techniques Used: Interviews, Surveys, Brainstorming.

How the Techniques Produced the Requirement: Interviews showed users want an at-a-glance summary; surveys validated which fields are important; brainstorming defined the dashboard columns and filter/sort behavior.

4.1.3 FR-003 — Manual Input (Ad-hoc Scan)

Description: Allow users to paste or type suspicious text/URLs and trigger an immediate scan; validate input (reject empty submissions).

Elicitation Techniques Used: Interviews, Surveys, Focus Group.

How the Techniques Produced the Requirement: Interviews revealed copy-paste scanning is common; surveys confirmed it as a priority; brainstorming specified validation rules and UI placement.

4.1.4 FR-004 — SMS / Email Integration (Permissioned)

Description: Integrate with user SMS/email only after explicit consent; enable reading of incoming messages for scanning when permission is granted; allow revoke and log access.

Elicitation Techniques Used: Interviews, Brainstorming.

How the Techniques Produced the Requirement: Interviews emphasised privacy and consent; surveys showed conditional acceptance; brainstorming produced the grant/revoke workflow and audit logging requirement.

4.1.5 FR-005 — Auto Scanning Service (Background)

Description: Run a background service that automatically scans new messages and triggers alerts/notifications for suspicious content.

Elicitation Techniques Used: Interviews, Surveys, Brainstorming.

How the Techniques Produced the Requirement: Interviews captured the desire to avoid manual scans; surveys supported background automation if privacy is respected; brainstorming converted this into a background-job design and notification policy.

4.1.6 FR-006 — User Feedback Loop

Description: Allow users to mark flagged items as Scam or Safe (with optional comment); store feedback linked to the original scan for model improvement.

Elicitation Techniques Used: Interviews, Focus Group, Brainstorming.

How the Techniques Produced the Requirement: Interviews showed users want correction ability; brainstorming specified linkage to scan records and an exportable feedback format for retraining.

4.1.7 FR-007 — Push Notifications

Description: Deliver push notifications for flagged threats that include the label, confidence score, and a visual severity cue (color); allow deep-link to the detailed view.

Elicitation Techniques Used: Interviews, Surveys, Brainstorming.

How the Techniques Produced the Requirement: Interviews defined useful notification content; surveys confirmed the need for immediate alerts; brainstorming specified payload structure and deep-link behavior.

4.1.8 FR-008 — Secure User Registration & Verification

Description: Provide signup/login via email (phone optional); enforce password strength (min. 8 chars, mixed types); verify accounts via email/SMS; prevent duplicate registrations.

Elicitation Techniques Used: Brainstorming.

How the Techniques Produced the Requirement: Brainstorming defined enforceable password and verification rules plus duplicate checks.

4.1.9 FR-009 — Settings & Account Preferences

Description: Provide user-configurable settings for notifications, language, and scanning behavior; persist preferences per account.

Elicitation Techniques Used: Focus Group, Surveys, Brainstorming.

How the Techniques Produced the Requirement: Interviews identified which preferences users need; surveys confirmed expectation of persistence; brainstorming defined storage and UI placement.

4.1.10 FR-010 — Multilingual Analysis (Detection)

Description: Detect and classify messages in multiple languages (initial set based on user need); if unsupported, present clear "unsupported language" feedback.

Elicitation Techniques Used: Surveys, Focus Group.

How the Techniques Produced the Requirement: Interviews with multilingual users showed the need; surveys identified priority languages; brainstorming set initial scope and fallback messaging.

4.1.11 FR-011 — Summary Screen & History

Description: Maintain a timestamped history of scans and present a summary visualization (pie chart) of classification distribution over selectable time ranges.

Elicitation Techniques Used: Interviews, Brainstorming.

How the Techniques Produced the Requirement: Interviews expressed need to audit past alerts; surveys supported visualization; brainstorming defined history schema and chart requirements.

4.1.12 FR-012 — Multilingual App UI

Description: Localize UI text and notifications to the user's selected language and detect preferred language on first-run where feasible.

Elicitation Techniques Used: Surveys, Brainstorming.

How the Techniques Produced the Requirement: Interviews emphasised full UI localization; surveys reinforced expectation for localized notifications/explanations; brainstorming defined detection and runtime translation strategy.

4.2 Identified Non-Functional Requirements (NFRs)

4.2.1 NFR-001 — Privacy & Data Minimization (Anonymous)

Description: By default do not store raw message bodies; store only analysis metadata (label, confidence, explanation, timestamp). Raw messages may be stored only with explicit user opt-in and an auditable flag.

Elicitation Techniques Used: Interviews, Surveys, Brainstorming, Focus Group.

How the Techniques Produced the Requirement: Interviews surfaced strong privacy concerns; surveys confirmed privacy as a priority; brainstorming produced the default-minimization policy and opt-in mechanism.

4.2.2 NFR-002 — Secure Connections & Encryption

Description: Enforce TLS (HTTPS) for all network traffic; store credentials with strong hashing (bcrypt/argon2); use a secrets manager for keys.

Elicitation Techniques Used: Interviews, Surveys, Brainstorming, Focus Group.

How the Techniques Produced the Requirement: Security-focused interviews demanded encrypted transport/storage; surveys indicated trust benefits from explicit security measures; brainstorming specified TLS enforcement, hashing, and secrets management.

4.2.3 NFR-003 — Accuracy (Model Performance)

Description: The detection model should meet project precision/recall targets (team to set numeric goals) and present confidence scores so users can interpret results.

Elicitation Techniques Used: Interviews, Brainstorming.

How the Techniques Produced the Requirement: Interviews clarified acceptable false-positive/false-negative tradeoffs; brainstorming converted tolerance into measurable model targets and a UI confidence display.

4.2.4 NFR-004 — Performance (Latency)

Description: Manual scans should return results within few seconds.

Elicitation Techniques Used: Interviews, Focus Group, Brainstorming.

How the Techniques Produced the Requirement: Interviews specified acceptable wait times; brainstorming defined testable latency metrics and monitoring approach.

4.2.5 NFR-005 — Simplicity & Accessibility (Usability)

Description: UI and explanations must be simple and accessible to non-technical users (plain language, minimal steps, basic accessibility checks).

Elicitation Techniques Used: Interviews, Focus Group, Brainstorming.

How the Techniques Produced the Requirement: Interviews with non-technical users revealed pain points with jargon; surveys confirmed broad need for plain-language explanations; brainstorming translated findings into style guidelines and minimal-step flows.

5 User Stories (Product Backlog)

Functional Stories

Story 1: Scam Detection

Front of the Card	Back of the Card
USER STORY **As a user,** I want the app to detect phishing links, scams so that I stay safe from fraud.	**SUCCESS CRITERIA** • Detects scam accurately. • Returns confidence score (0-100). • Explains results in simple language. **FAILURE CRITERIA** • Scam missed. • Confidence not shown. • Overly Technical explanation.

Story 2: Result Dashboard

Front of the Card	Back of the Card
USER STORY **As a user,** I want a simple dashboard, so that I can easily understand my scan results.	**SUCCESS CRITERIA** • Dashboard shows scan results clearly • Threats categorized by risk level. • Display confidence score (0-100). **FAILURE CRITERIA** • Dashboard confusing/empty. • Wrong or missing scan result.

Story 3: Manual Input

Front of the Card	Back of the Card
USER STORY **As a user,** I want to paste suspicious text, so that I can scan it manually.	**SUCCESS CRITERIA** • Manual input accepted. • Result shown within set time. **FAILURE CRITERIA** • Empty text accepted. • No response from app.

Story 4: SMS/Email Integration

Front of the Card	Back of the Card
USER STORY **As a user,** I want secure SMS/email integration, so that messages are scanned automatically if I give permission.	**SUCCESS CRITERIA** • Email/SMS access granted with permission. **FAILURE CRITERIA** • Email/SMS access granted without permission. • Email/SMS access not granted even with permission. • Random selection/deselection of permission.

Story 5: Auto Scanning Service

Front of the Card	Back of the Card
USER STORY **As a user,** I want automatic scanning of new messages, so that I don't have to scan manually every time.	**SUCCESS CRITERIA** • Background service scans instantly. • Alerts triggered for suspicious text. • Result shown in notification with number and colour showing confidence level. **FAILURE CRITERIA** • Auto-scan not functioning properly. • Alerts are not triggered. • Result not shown in notification with number and colour showing confidence level.

Story 6: User Feedback

Front of the Card	Back of the Card
USER STORY **As a user,** I want to confirm or deny flagged messages as scam or safe, so that the system can improve accuracy.	**SUCCESS CRITERIA** • Feedback stored with scan result. • Improves future detection. **FAILURE CRITERIA** • Feedback is not stored. • Feedback not linked to the right message. • Feedback form is not appearing.

Story 7: Push Notifications

Front of the Card	Back of the Card
USER STORY **As a user,** I want push notifications for threats, so that I can act accordingly.	**SUCCESS CRITERIA** • Result is shown in notification with number and color showing confidence level. **FAILURE CRITERIA** • Result not shown in notification with number and color showing confidence level. • Wrongly sends an alert for safe text. • Notification sent corresponding to wrong message.

Story 8: Secure User Registration

Front of the Card	Back of the Card
USER STORY **As a user,** I want to register with email/phone and a strong password, so that I can securely access the app.	**SUCCESS CRITERIA** • Registration validates unique email/phone and enforces password strength (min 8 chars, mix of types). • Confirmation email/SMS sent with verification link. • User profile created with default settings. **FAILURE CRITERIA** • Duplicate accounts created. • Weak passwords accepted. • No verification step, leading to unconfirmed access.

Story 9: Settings and Account

Front of the Card	Back of the Card
USER STORY **As a user,** I want account and preference settings, so that I can control how the app behaves.	**SUCCESS CRITERIA** • Preferences saved as per user. • Notifications and language adjustable. **FAILURE CRITERIA** • Settings reset to default. • Wrong account modified.

Story 10: Multilingual Analysis

Front of the Card	Back of the Card
USER STORY **As a user,** I want the different language messages analysed, so that I can detect scams regardless of language.	**SUCCESS CRITERIA** • Different language messages scanned. • Correct scan results across all languages. **FAILURE CRITERIA** • Language not supported. • Wrong scan results in other languages.

Story 11: Summary Screen with History

Front of the Card	Back of the Card
USER STORY **As a user,** I want a pie chart showing classification of messages and history of messages, so that I can track my safety over time.	**SUCCESS CRITERIA** • History visible with timestamps. • Summary pie chart shows classification of messages. **FAILURE CRITERIA** • History missing. • Pie chart shows wrong classification of messages. • No history or pie chart visible.

Story 12: Multilingual App UI

Front of the Card	Back of the Card
USER STORY **As a user,** I want the application user interface to display all the information in the selected language, so that I can understand the message and the reasoning behind it properly.	**SUCCESS CRITERIA** • The app detects the user's preferred language and shows UI in that language. • All notifications should be shown in the selected language. **FAILURE CRITERIA** • The app shows the UI in some other language. • The app doesn't change the UI.

Non-Functional Stories

Story 1: Privacy (Anonymous)

Front of the Card	Back of the Card
USER STORY **As a user,** I want my message content to remain private, so that my sensitive data is not stored unnecessarily.	**SUCCESS CRITERIA** • Only analysis output stored • No raw message saved **FAILURE CRITERIA** • Full message stored

Story 2: Secure Connections (Cyber Attack Proof)

Front of the Card	Back of the Card
USER STORY **As a user,** I want secure encrypted connections, so that my credentials and data remain safe.	**SUCCESS CRITERIA** • HTTPS + encrypted API calls • Passwords encrypted **FAILURE CRITERIA** • Plain-text data transmission • Password stored in plain text

Story 3: Accuracy

Front of the Card	Back of the Card
USER STORY **As a user,** I want accurate detection, so that false alarms are minimized.	**SUCCESS CRITERIA** • High confidence on known scams • Few false positives/negatives **FAILURE CRITERIA** • Many safe texts flagged wrongly • Scams missed repeatedly

Story 4: Performance

Front of the Card	Back of the Card
USER STORY **As a user,** I want fast scan results, so that I can act quickly.	**SUCCESS CRITERIA** • Manual scan within 10s • Auto scan within 30s **FAILURE CRITERIA** • Delays longer than expected • App freezes during scan

Story 5: Simplicity and Accessibility

Front of the Card	Back of the Card
USER STORY **As a non-technical user,** I want the app to be simple, so that I can understand it easily.	**SUCCESS CRITERIA** • UI simple and intuitive • Explanations in plain language **FAILURE CRITERIA** • Complicated terms shown • Confusing UI flow

6 Grouping of userstories into EPICS (Sprints)

Product Backlog: Epics and User Stories Overview

Epic	ID	User Story Title
Epic 1: Scam Detection & Analysis	1	Scam Detection
	10	Multilingual Analysis
	15	Accuracy
	16	Performance
Epic 2: User Interaction & Visualization	2	Result Dashboard
	7	Push Notifications
	11	Summary Screen with History
	17	Simplicity & Accessibility
Epic 3: Input & Scanning Options	3	Manual Input
	4	SMS/Email Integration
	5	Auto Scanning Service
Epic 4: User Account & Personalization	8	Secure User Registration
	9	Settings & Account
	12	Multilingual App UI
Epic 5: Feedback & Continuous Improvement	6	User Feedback
Epic 6: Security & Privacy	13	Privacy (Anonymous)
	14	Secure Connections (Cyber Attack Proof)

Epic Summary

Epic Name	Story Count	Focus Area
Epic 1: Scam Detection & Analysis	4	Core detection functionality, accuracy, and performance
Epic 2: User Interaction & Visualization	4	User interface, dashboards, notifications, and accessibility
Epic 3: Input & Scanning Options	3	Different ways users can input and scan content
Epic 4: User Account & Personalization	3	User registration, settings, and localization
Epic 5: Feedback & Continuous Improvement	1	System learning and improvement mechanisms
Epic 6: Security & Privacy	2	Data protection and secure communications
Total	**17**	**Complete scam detection application**

7 Conflict Identification and Resolution

Conflicting Epics	Epic-5 (Advanced Detection Quality - Sprint 3) Epic-6 (Performance - Sprint 4)
Reason for Conflict	Achieving higher accuracy and multilingual analysis may slow scanning times, violating performance targets.
Resolution	Allow users to choose between two levels of scanning: 1. **Fast Basic Scan** – quick results. 2. **Detailed Deep Scan** – more accurate checking.

8 Proof of Concept (POC) for Sprint 1

UI design: Approximate vision of the app

A prototype of the application was designed in Figma. This serves as the visual blueprint for our development, ensuring a consistent and user-centric approach. The design prioritizes simplicity and clarity, adhering to the feedback gathered during our elicitation process.

Key Screens Designed:

- **Sign-in & Sign-up:** A clean and simple interface for user authentication.
- **Main Scan Screen:** A minimalist screen with a single, clear call-to-action for pasting and analyzing text.
- **Results Screen:** An easy-to-understand screen for displaying threat analysis.

User Authentication System: The Functional Gateway

We have successfully coded a functional user sign-in and sign-up system. This component serves as the secure entry point to the application and is a critical prerequisite for features planned in later sprints, such as saving a user's scan history.

Technology Used: [e.g., Flutter Dart]

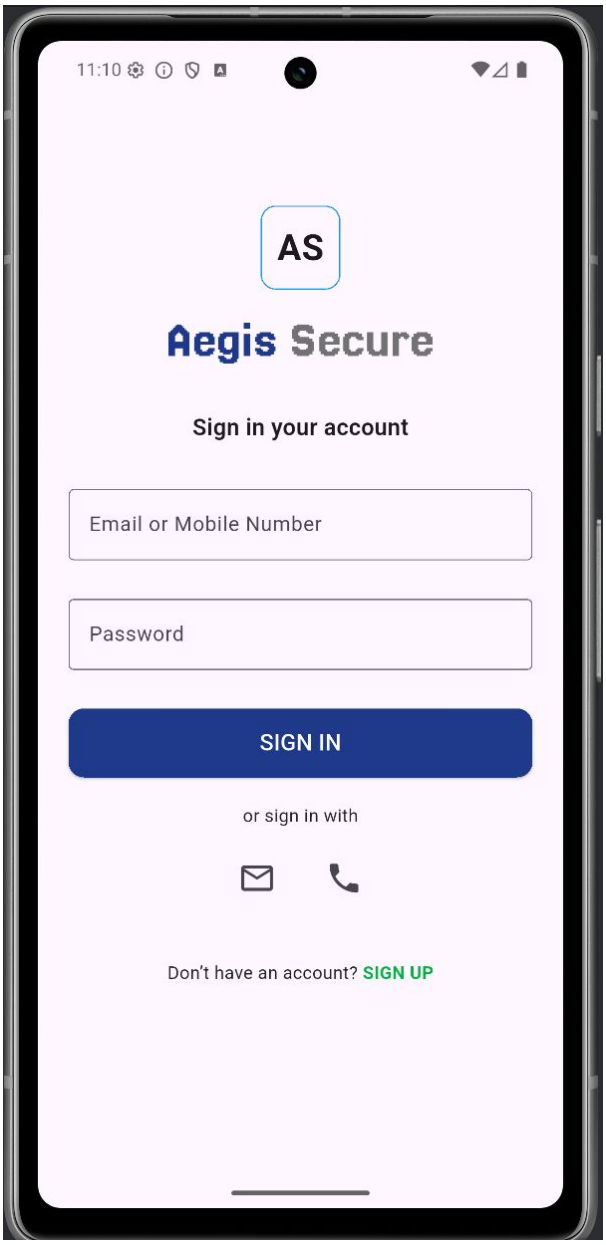


Figure 1: *
Sign-in Page

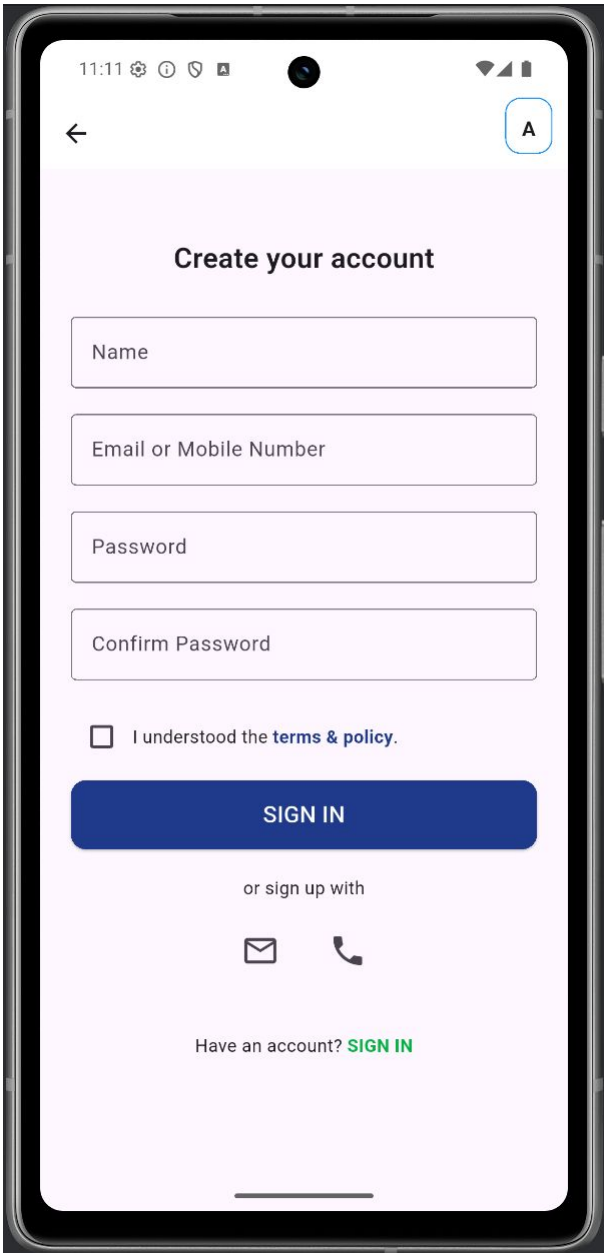


Figure 2: *
Account Creation page

Figure 3: Figma prototype of the Authentication screens.

Core Analysis Engine: A ML Model

The "brain" of the Aegis-Secure app, a machine learning model capable of detecting scams, has been developed and tested. This represents the core technological innovation of our project.

Model Type: Model is a basic MLP(Multi Layer Perceptron) which takes message as input and calculate probability of scam as output

Training Data: The model was trained on a curated dataset of lakhs of real-world phishing messages, SMS scams, and legitimate texts.

Demonstrated Capability: The standalone model can successfully take a text input and output a classification (e.g., "Scam," "Safe") along with a confidence score, proving its viability. This successful test confirms that our core analysis engine is effective..

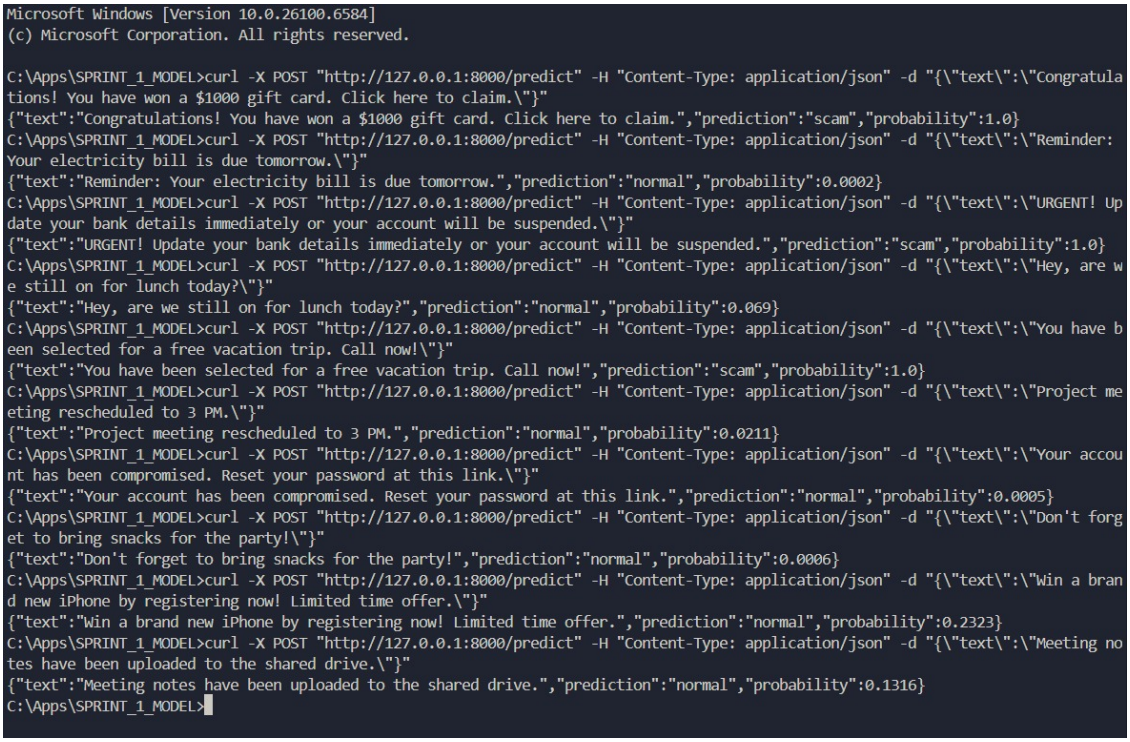


Figure 4: Output showing the ML model correctly classifying a scam message.

For more information, please refer to the GitHub repository. Link provided below.

9 GitHub Project Repository Link

Github Repository: Aegis-Secure

The provided link to the repository contains all the information provided in this document, and all the deliverables asked for in the Mid-evaluation.
It also contains the raw data and documentations made from the various elicitation techniques employed.

10 Team Member Contributions

All Team members have contributed across all aspects of the projects so far. Though the team has been split up for implementation sprints, the pre-planning and requirements elicitation has common contribution from all. The below table only shows areas in which significant work or leadership has been taken.

Roll No.	Name	Contribution
202301484	Soham Mevada	Activity Diagram, ML Model Development, FR
202301463	Suryadeepsinh Gohil	Interviews, Stakeholder identification, requirements
202301476	Neelabh Rana [L]	Figma Design, Frontend, Sprint Concept
202301441	Hrithik Patel	Focus Group management, Functional requirements, Survey design
202301445	Pransu Vadsmiya	Document review and Analysis, Frontend, Focus Group
202301095	Dhruv Patel	Survey Inferences, Backend, Stakeholders
202301480	Jenish Bhagiya	Epics, User Stories, Conflicts
202301460	Akshat Bhatt	ML Model development, Epics, User Stories
202301436	Vrajkumar Makwana	Document review, and consolidation, Frontend, Focus Group
202301479	Mihirsinh Chavda	Stakeholders, Interview, NFR, Sprint Concept

Table 2: Team Member Contributions