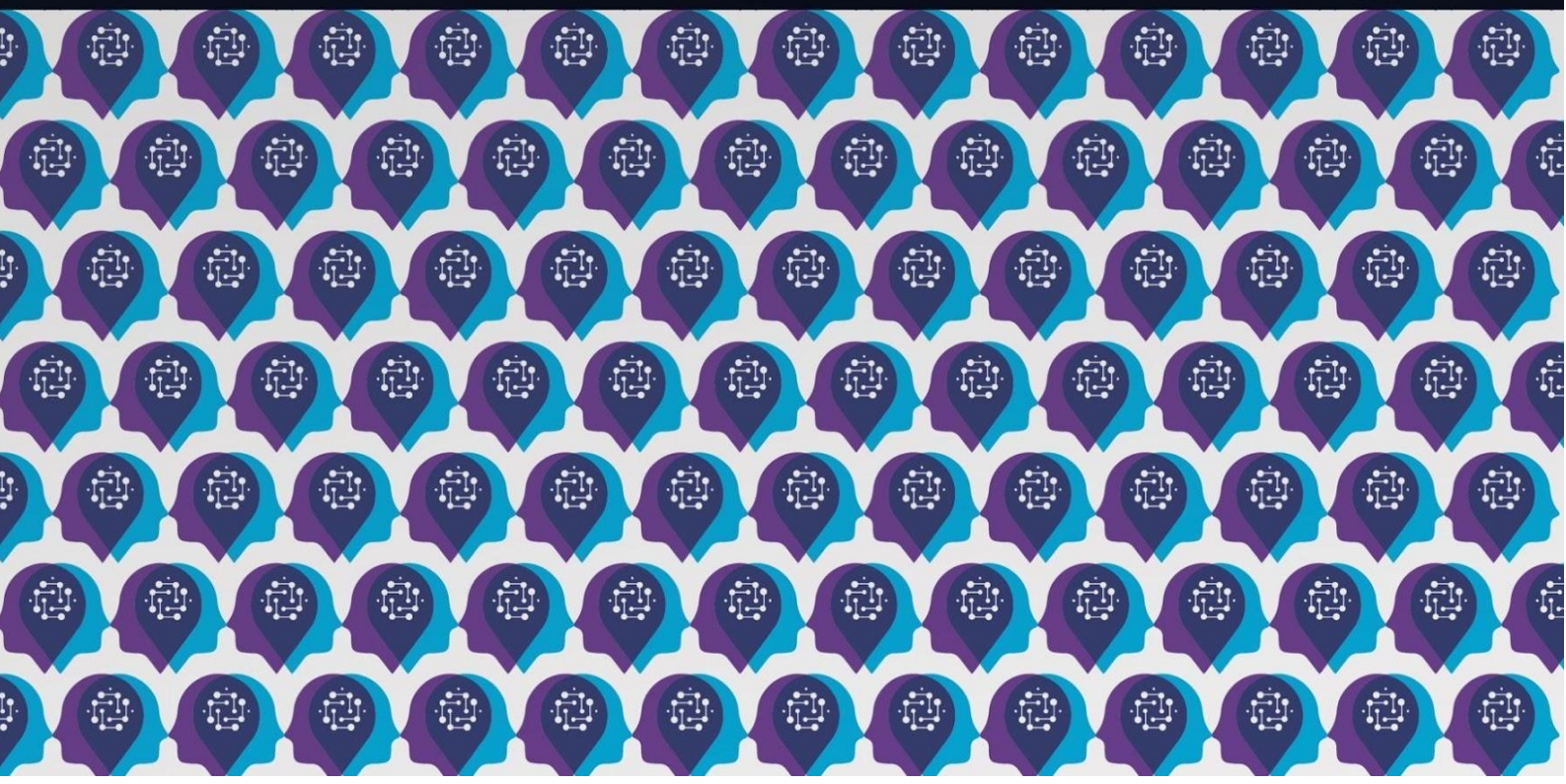
AI4Debunk

D10.3 REPORT ON THE DEFINITION OF THE APP

February 2026

Funded by
the European Union



Grant Agreement No.: 101135757
Call: HORIZON-CL4-2023-HUMAN-01-CNECT
Topic: HORIZON-CL4-2023-HUMAN-01-05
Type of action: HORIZON Innovation Actions

D10.3 REPORT ON THE DEFINITION OF THE APP

Project Acronym	AI4Debunk
Project Number	101135757
Project Full Title	Participative Assistive AI-powered Tools for Supporting Trustworthy Online Activity of Citizens and Debunking Disinformation
Work package	WP 10
Task	Task 3
Due date	28/02/2026
Submission date	28/02/2026
Deliverable lead	DOTSOFT
Version	1.0
Authors	Dr. Despina Elisabeth Filippidou (DOTSOFT), Karanasios George (DOTSOFT), Stavros Katsaridis (DOTSOFT), Katsakioris Dimitris (DOTSOFT), Nikas Marios (DOTSOFT), Simeonidou Anastasia (DOTSOFT), Nikopoulos George (DOTSOFT), Maragkos Christodoulos (DOTSOFT)
Contributors	
Reviewers	Georgi Gotev, Kalina Angelova (EUalive)
Abstract	AI4Debunk D10.3 specifies the AI4Debunk mobile app through detailed use cases, user stories, user journeys, and a dedicated security/privacy analysis, concluding with a technology-stack justification. The app enables multi-modal verification (text/URL, file upload, live audio, AR scanning) by sending requests to a Debunking API that returns a Disinfoscore, classification, and explanations, presented via a clear risk indicator and verdict.

Funded by the European Union

Keywords	Disinformation debunking; mobile application specification; multi-modal verification; Debunking API; Disinfoscore; onboarding and informed consent; augmented reality (AR) scanning; live audio capture; privacy and anonymity; security risk analysis; reporting to experts; Flutter cross-platform development.

DOCUMENT DISSEMINATION LEVEL

Dissemination level	
X	PU - Public
	SEN - Sensitive

DOCUMENT REVISION HISTORY

Version	Date	Description of change	List of contributor(s)
0.1	05/12/2025	First draft	DOTSOFT
0.2	21/02/2025	Final draft for internal review by EUalive	DOTSOFT
1.0	24/02/2025	Revised version after internal review by EUalive	DOTSOFT

STATEMENT ON MAINSTREAMING GENDER

The AI4Debunk consortium is committed to including gender and intersectionality as a transversal aspect in the project's activities. In line with EU guidelines and objectives, all partners – including the authors of this deliverable – recognize the importance of advancing gender analysis and sex-disaggregated data collection in the development of scientific research. Therefore, we commit to paying particular attention to including, monitoring, and periodically evaluating the participation of different genders in all activities developed within the project, including workshops, webinars and events but also surveys, interviews and research, in general. While applying a non-binary approach to data collection and promoting the participation of all genders in the activities, the partners will periodically reflect and inform about the limitations of their approach. Through an iterative learning process, they commit to plan and implement strategies that maximize the inclusion of more and more intersectional perspectives in their activities.

Funded by the European Union

DISCLAIMER

The AI4Debunk project has received funding from the European Union's Horizon Europe Programme under the Grant Agreement No. 101135757.

Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

COPYRIGHT NOTICE

© AI4Debunk - All rights reserved
No part of this publication may be translated, reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the written permission of the publisher or provided the source is acknowledged.

How to cite this report: Dr. Despina Elisabeth Filippidou, Karanasios George, Stavros Katsaridis, Katsakioris Dimitris, Nikas Marios, Simeonidou Anastasia, Nikopoulos George, Maragkos Christodoulos (2025). AI4Debunk D10.3: Report on the definition of the app.

The AI4Debunk consortium is the following:

Participant number	Participant organisation name	Short name	Country
1	LATVIJAS UNIVERSITATE	UL	LV
2	FREE MEDIA BULGARIA	EURACTIV	BE
3	PILOT4DEV	P4D	BE
4	INTERNEWS UKRAINE	IUA	UA
5	CONSIGLIO NAZIONALE DELLE RICERCHE	CNR-IRPPS	IT
6	UNIVERSITA DEGLI STUDI DI FIRENZE	MICC/UNIFI	IT
6.1	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI	CNIT	IT
7	BARCELONA SUPERCOMPUTING CENTER CENTRO NACIONAL DE SUPERCOMPUTACION	BSC	ES
8	DOTSOFT OLOKLIROMENES EFARMOGES DIADIKTIOY KAI VASEON DEDOMENON AE	DOTSOFT	EL
9	UNIVERSITE DE MONS	UMONS	BE
10	NATIONAL UNIVERSITY OF IRELAND GALWAY	NUIG	IE
11	STICHTING HOGESCHOOL UTRECHT	HU	NL
12	STICHTING INNOVATIVE POWER	IP	NL
13	F6S NETWORK IRELAND LIMITED	F6S	IE

Funded by the European Union

Funded by the European Union

Funded by the European Union

Funded by the European Union

LIST OF IMAGES

Funded by the European Union

ABBREVIATIONS

WP	Work Package
EC	European Commission
AR	Augmented Reality
API	Application Programming Interface
OCR	Optical Character Recognition
URL	Uniform Resource Locator
SUS	System Usability Scale
GPS	Global Positioning System
GDPR	General Data Protection Regulation
UI/UX	User Interface/User Experience
OS	Operating System
HTTP	Hypertext Transfer Protocol
QR	Quick Response

Funded by the European Union

EXECUTIVE SUMMARY

This deliverable defines the AI4Debunk mobile application at the "what the app does and how it behaves" level, organizing requirements into a full set of use cases (UC-01 to UC-16), user stories/epics, end-to-end usage journeys, a dedicated security concerns chapter, and a technology-stack justification.

The app's core experience is designed around a home screen with four primary entry points (Paste your Text or URL, Upload your File, Record your Sound, and Scan with AR), plus an optional global misinformation metric sourced from an analytics backend. Verification requests are sent to a Debunking API, which returns a Disinfoscore, a short classification, and explanations; results are presented with a clear visual risk indicator and a one-sentence verdict, with optional reporting for logged-in users.

A first-time onboarding flow explicitly explains purpose, supported functions, and data processing, and records informed consent before the user proceeds. Beyond URL/text checks, the specification covers live audio capture (record → review → accept/dismiss → analyse), AR scanning of offline/physical content (OCR/URL extraction with AR-overlay results), and escalation via reporting suspicious items to an expert backend (e.g., Disinfopedia) with minimal metadata and an optional user note.

Security analysis highlights anonymity as the central risk: persistent accounts (including social login) and synchronized histories make the system pseudonymous by default, while analytics/log retention and cross-service reporting can increase linkability and re-identification risk over time, motivating strong minimisation, retention controls, and clear user communication. Finally, Flutter is selected to deliver a consistent Android/iOS implementation with strong performance for camera/audio flows, supported by a mature plugin ecosystem (camera, microphone, secure storage, networking, and social login).

Funded by the European Union

1. INTRODUCTION

AI4Debunk is conceived as a user-facing mobile application that helps citizens assess whether the information they encounter is likely misleading or false, across both online and offline contexts. Building the solution as a smartphone app is a deliberate choice: it places disinformation checking directly where users consume and encounter content (e.g., while scrolling social media, reading messages, or being exposed to information in public spaces), enabling quick and practical verification "in the moment" through a user-friendly interface designed for all age groups. In practical terms, the app adopts an onboarding-first experience that explains the service purpose and data processing transparently and captures informed consent, while keeping the interaction model simple and accessible (clear entry points, minimal steps, and understandable results).

In line with the project's strong commitment to privacy and data minimisation, the AI4Debunk mobile application does not store or retain any personal data of its users. The system processes verification requests transiently and keeps only anonymous operational logs necessary for performance monitoring, statistical analysis, and security auditing. These logs contain no personally identifying information and are managed according to strict privacy and retention policies consistent with GDPR.

From a functional perspective, the app supports multi-modal analysis and allows the user to submit and receive a Disinfoscore for diverse entities, including digital social media posts (text/URLs), screenshots and images, uploaded video/audio clips, real-life audio recordings captured on the spot, and real-life printed material such as flyers, posters, or leaflets photographed with the camera (including AR-supported scanning/OCR when applicable). These verification flows leverage the APIs already defined in the related project tasks (notably the Debunking API and its supporting services), so that all modalities are handled through consistent backend interactions and results are presented with a clear risk indicator, short verdict, and explanatory signals suitable for non-expert users.

This report documents the definition of the AI4Debunk app by translating expected behaviour into a structured requirements view: detailed use cases (account lifecycle, onboarding/consent, verification flows, reporting, profile/history), user stories grouped into epics, and an end-to-end narrative of how the app is intended to be used in everyday scenarios. It further provides a focused security analysis, emphasizing anonymity, de-identification, and linkability risks, and concludes with a rationale for the chosen mobile technology stack and framework.

Funded by the European Union

1.1 OBJECTIVES

This deliverable aims to provide a complete definition of the AI4Debunk mobile app by (a) specifying its functional scope and expected behaviour through a structured set of use cases (UC-01 to UC-16), (b) translating requirements into user stories grouped by epics to support implementation planning, and (c) describing how the app will be used end-to-end through realistic user journeys.

In addition, it explicitly aims to (d) identify and discuss security and privacy concerns associated with the app's workflows, and (e) justify the proposed mobile technology stack and framework choice to ensure the implementation aligns with the defined requirements.

1.2 EXPECTED OUTCOME

The expected outcome of D10.3 is a single, implementation-ready reference that consolidates the AI4Debunk app definition into (i) detailed use cases, (ii) user stories grouped into epics, (iii) an end-to-end narrative of how the app will be used, (iv) a structured set of security concerns, and (v) a justified technology stack and framework choice, so all stakeholders (product, UX, engineering, and partners) share the same baseline for development and validation.

In practice, this deliverable is also meant to drive concrete design and implementation decisions: it surfaces anonymity/privacy risks and the need for explicit mitigations (e.g., minimisation/retention controls and careful handling of sensitive modalities), and it supports long-term delivery by framing the app in terms of maintainability, evolution, and testability of security-sensitive behaviours.

2. USER REQUIREMENTS ANALYSIS

This section consolidates the functional needs of the AI4Debunk mobile app and translates them into an implementation-ready requirements baseline. Following an Agile methodology, requirements were captured and refined iteratively in short feedback cycles with project stakeholders, allowing priorities and edge cases to be clarified early and continuously. The resulting backlog was expressed in clear user stories grouped into epics and complemented by detailed use cases, ensuring both a user-centric view ("what the user needs") and a system-oriented view ("how the app behaves"). In parallel, UI mockups/wireframes were produced directly from the user requirements to validate navigation, wording, and key interaction flows before development, and to maintain traceability between requirements, design decisions, and

subsequent testing. Overall, the goal was to provide a coherent, testable description of expected behaviour that the UX and engineering teams can use as a common reference throughout implementation.

2.1 USER STORIES

Epic 0 – Registration, Login & Account Lifecycle

As a user	I want to continue as a Guest	so that I can quickly check content without creating an account
As a new user	I want to create an account	so that I can use the app while having a persistent profile
As a returning user	I want to log in with my existing credentials or identity provider	so that I can access my synced history and reports across devices
As a user	I want the app to remember my authenticated session between uses (with reasonable timeout and device-level protection)	so that I don't need to log in every time I open it
As a user who forgot my password	I want a secure password reset flow (e.g. email link)	so that I can regain access to my account without losing my data
As a user	I want to log out from the app on a device	so that other people using the same device cannot see my history or reports
As a user	I want to delete my associated personal data (subject to legal retention constraints)	so that I can withdraw from the service in line with GDPR

Epic A – Onboarding After Login & Home Screen Layout

As a newly registered/logged-in user	I want to see a short, accessible introduction explaining what AI4Debunk is, what its purpose is, what files it supports and how it works	so that I know why and how to use it
As a user	I want a clear explanation of what data is processed and for what purposes	so that I can give informed consent in line with privacy and ethics
As a user	I want to land on a home screen with clear entry points ("Paste your Text or URL", "Upload your File", "Record your Sound", "Scan with AR")	so that I can immediately start using the main functions
As a user	I want to see on the home screen the total number of files that have been identified as misinformation globally	so that I understand the overall impact of the system and feel that my own checks contribute to a wider collective effort against misinformation

Epic B – Verification of URL / Text

As a user	I want to select "Paste your Text or URL", paste a text or URL, and receive a disinformation assessment	so that I can quickly judge whether something is likely disinformation
As a non-expert user	I want to see a clear visual risk indicator and a one-sentence verdict on the result screen	so that I can understand the outcome at a glance without reading technical details
As a user who wants more context	I want to expand the result to see key signals, reasoning, and links to related debunked cases	so that I understand why the content was labeled that way
As a user	I want the app to inform me if there is no network or the API cannot be reached, and allow me to retry	so that I understand why no result is available

Funded by the European Union

Epic C – Verification of Files

As a user	I want to select "Upload your File", pick one from the file explorer, and receive a disinformation assessment	so that I can quickly judge whether something is likely disinformation
As a user	I want the app to inform me if it lacks the file explorer permission	so that I can still complete the check

Epic D – Verification of Sound Recordings

As a user who is hearing something suspicious right now (e.g. on TV or radio)	I want to tap a single record button to start recording and tap it again to stop	so that I can quickly capture the audio without dealing with complex controls
As a user	I want clear visual feedback (e.g. timer or recording indicator) while recording is active	so that I know the app is capturing sound and for roughly how long
As a user who has finished recording	I want the app to switch to a review state with my recorded clip and two buttons (dismiss / accept)	so that I can decide whether to discard the clip or send it for analysis
As a user	I want to be able to dismiss the current recording	so that I can cancel the attempt if I captured the wrong audio or changed my mind
As a user	I want to accept the recording with a single tap	so that the app can proceed to validate and analyze the audio without extra steps
As a user	I want the app to check that my recording is at least a minimum duration required to estimate whether it may contain misinformation	so that only meaningful samples are analyzed

Funded by the European Union

As a user whose recording is too short	I want to see a clear error message explaining that the sample is too short and prompting me to record a longer sample	so that I understand why no analysis is performed and how to fix it
As a user	I want the outcome of my recording to be shown on the same results page used for other checks	so that the behaviour is consistent regardless of how the audio was provided
As a user	I want the app to inform me if it lacks the microphone permission	so that I can enable it
As a privacy-conscious user	I want any dismissed or too-short recordings to be discarded according to the app's privacy rules and not uploaded to the server	so that unnecessary audio samples are not stored or processed

Epic E – Verification of Physical Content (Posters, Flyers, QR Codes)

As a user	I want to choose "Scan with AR", capture a photo with the camera or scan a QR code, and have the app extract data for analysis	so that offline disinformation can be checked like online content
As a user	I want the app to inform me if it lacks the camera permission	so that I can enable it

Epic F – Reporting to Experts & Ecosystem Participation

As a user	I want to report a piece of content by pressing a button in the analysis results screen	so that I can improve the integrity and reliability of the information shared across the platform
As a user	I want to see a clear confirmation that my report was submitted	so that I know my contribution entered the system

Epic G – Profile & Account Information

As a user	I want to see my profile screen with my profile image, username and email	so that I immediately recognize that I'm in my own account area

Funded by the European Union

As a user	I want to see my key stats on my profile screen (total files analyzed, total reports made, average misinformation rating)	so that I can understand how actively I use the app and how often I encounter misinformation
As a user	I want to see a chronological list of all files I've analyzed on my profile screen	so that I can review what I have checked in the past
As a user	I want each file entry in the history to show the file name, date/time of upload, file size, file type, and probability of being fake (percentage)	so that I can quickly recall what each file was and how risky it was assessed to be
As a user	I want a search bar above my file history where I can type part of a file name	so that I can quickly find a specific file I previously analyzed
As a user	I want clickable filter chips (e.g. "Images", "Audio", "Video") above the history list	so that I can narrow the list to only the type of files I'm interested in
As a user	I want the probability of being fake for each file to be clearly visible (as a percentage) directly in the list	so that I can quickly scan which items were most suspicious

2.2 USE CASES

UC-01 – REGISTER NEW ACCOUNT

Primary actor: User
Goal: Create a persistent account to sync checks and reports across devices.
Scope: Smartphone app + Authentication backend
Preconditions:

- App is installed and has network connectivity.
- User does not yet have an account (or is not logged in).

Main success scenario:

1. User opens the app.
2. User selects "Sign Up".
3. App shows a simple registration form (e.g., first name, email, password, and social login options).
4. User provides required information and confirms.
5. App sends registration data securely to the authentication backend.
6. Backend validates data, creates a new user record and returns a success response.

Funded by the European Union

7. App logs the user in automatically and navigates to the onboarding / home screen.

Postconditions:

- A new user account exists and is associated with a persistent profile.
- An authenticated session is created on the device.

Alternative / exception flows:

- 4a – Weak or invalid password: Backend returns validation errors; app shows user-friendly guidance (e.g., password strength rules) and allows correction.
- 4b – Email already registered: Backend indicates duplicate account; app suggests "Log in" or "Reset password" instead.
- 5a – No network / server error: App informs the user that registration failed due to connectivity and allows retry.

UC-02 – Log In with Existing Credentials or Identity Provider

Primary actor: Returning User

Goal: Access personal history and reports from any device.

Scope: Smartphone app + Authentication backend + Identity provider(s)

Preconditions:

- User has previously registered (via email/password or external IdP).
- App has network connectivity.

Main success scenario:

1. User opens the app.
2. App shows "Login" options (e.g., email/password and one or more identity providers).
3. User chooses a login method:
    a. Enters email and password, or
    b. Taps identity provider button (e.g., Google).
4. App sends credentials or IdP token to the authentication backend.
5. Backend validates identity and returns an authentication token and profile data.
6. App stores a session token securely on the device (respecting OS security features).
7. App navigates to the home screen with the user authenticated.

Postconditions:

- User is logged in and can access personal history and reports.
- A secure session is established on the device.

Alternative / exception flows:

- 4a – Invalid email/password: Backend rejects login; app shows a non-technical error and offers "Forgot password".
- 3b-1 – Identity provider cancelled: User cancels IdP login flow; app returns gracefully to login screen.

Funded by the European Union

- 4b – No network / IdP unreachable: App informs the user that login cannot be completed and suggests retry later.

UC-03 – Reset Forgotten Password

Primary actor: User who forgot their password
Goal: Regain access to the account without losing data.
Scope: Smartphone app + Authentication backend + Email service
Preconditions:

- User has an existing account with an email address.
- App has network connectivity.

Main success scenario:

1. On the login screen, user taps “Forgot Password?”.
2. App displays a password reset form asking for the registered email address.
3. User enters their email and submits.
4. App sends a password reset request to the authentication backend.
5. Backend generates a time-limited, single-use reset token and sends a reset link via email.
6. User opens the email and taps the reset link.
7. The system opens a secure reset page (in app or browser) where the user enters a new password.
8. Backend validates and stores the new password.
9. User is informed that the password has been changed and can now log in with the new credentials.

Postconditions:

- User’s password is updated.
- Existing data (history, reports) remains intact.

Alternative / exception flows:

- 3a – Email not associated with any account: Backend returns a generic response (e.g., “If an account exists, we’ve sent an email”) to avoid leaking account existence.
- 5a – Email not delivered / user can’t find it: User can request a new email; app respects throttling to avoid abuse.

- 8a – New password does not meet security policy: Reset page explains requirements and prompts user to try again.

UC-04 – Log Out from Device

Primary actor: User
Goal: Prevent other people using the same device from accessing personal checks and reports.

Funded by the European Union

Scope: Smartphone app
Precondition: User is logged in.
Main success scenario:

1. User opens the profile screen.
2. User taps “Log Out”.
3. App asks for confirmation (if necessary).
4. On confirmation, the app clears local session tokens and sensitive cached data.
5. App returns to the login screen.

Postconditions:

- No active authenticated session exists on the device.
- Next access to personal data requires login again.

Alternative / exception flow: 3a – User cancels logout: App remains logged in and returns to previous screen.

UC-05 – Delete Personal Data

Primary actor: User
Goal: Have personal data erased.
Scope: Smartphone app + User management backend
Preconditions:

- User is logged in.
- App has network connectivity.

Main success scenario:

1. User opens the profile / account screen.
2. User selects “Delete Personal Data”.
3. App shows a clear explanation of what will be deleted.
4. App asks the user to confirm deletion (e.g., “Type DELETE” and tap confirm).
5. App sends an authenticated delete request to the backend.
6. Backend deletes personal data according to policy, and returns confirmation.
7. App displays a final confirmation message.

Postconditions:

- Personal data of the user are no longer available.

Alternative / exception flows:

- 4a – User aborts deletion: User dismisses or cancels; no changes are made.
- 5a – Backend or network failure: App informs user that deletion could not be completed and suggests retry.

UC-06 – Complete First-Time Onboarding & Give Informed Consent

Primary actor: Newly registered / newly logged-in User
Goal: Understand what AI4Debunk is, what data it processes, and provide informed consent before use.
Scope: Smartphone app
Preconditions:

- User has successfully logged in.
- Either first login or policies have materially changed.

Main success scenario:

1. After login, the app presents an onboarding flow instead of going directly to the home screen.
2. The first screen briefly explains what AI4Debunk is and its purpose (e.g., helping identify disinformation).
3. One or more screens describe supported file types and main functions ("Paste Text/URL", "Upload File", "Record Sound", "Scan with AR").
4. A dedicated screen explains what data is processed, how it is used, and any sharing with trusted partners in accessible language.
5. User accepts by tapping "I consent".
6. App records consent (e.g., timestamp, app version, policy version) in the backend.
7. App transitions to the home screen.

Postconditions:

- User has been informed about the service and data processing.
- Consent status is stored for compliance and for future audits.

Alternative / exception flows:

- 6a – User declines consent: App informs user that core functionality cannot be provided without consent and may:
  - Offer a minimal mode with stricter limitations, or
  - Return to logout / exit.

- 7a – Network failure when recording consent: App stores consent locally and retries syncing later; user can still proceed, but the system queues the consent record.

UC-07 – View Home Screen & Global Misinformation Statistics

Primary actor: User
Goal: Start a verification quickly and understand the wider collective impact of AI4Debunk.
Scope: Smartphone app + Analytics backend
Preconditions:

- User is either authenticated or using Guest mode and has accepted the required consent.
- App has network connectivity (for live stats; core navigation can still work offline).

Main success scenario:

1. User lands on the home screen.
2. Home screen prominently displays the four main entry points:
    a. "Paste your Text or URL"
    b. "Upload your File"
    c. "Record your Sound"
    d. "Scan with AR"
3. App retrieves from the backend the aggregated count of files identified as misinformation and retrieves per-type counters (e.g., Text/URL, Image, Video, Audio, AR/Photo). The home screen displays the global counter plus a compact breakdown by content type.
4. App displays a clear summary metric (e.g., "Files Debunked: X").
5. User can tap any main entry point to start the corresponding verification use case.

Postcondition: User understands where to start and has a sense of the platform's global activity / impact.

Alternative / exception flow: 3a – Analytics backend unavailable: App gracefully falls back to cached statistics or hides the metric with a short message (e.g., "Stats temporarily unavailable") while keeping all main entry points functional.

UC-08 – Check of Online Content (URL / Text)

Primary actor: User

Goal: Quickly assess whether an online article, post or message is likely to be disinformation.

Scope: Smartphone app + Debunking API

Preconditions:

- The app is installed and has network connectivity.
- User has a URL or text snippet they want to check.

Main success scenario:

1. User opens the AI4Debunk app.
2. User selects "Paste your Text or URL".
3. User pastes a URL or text snippet (e.g., social media post, article excerpt).
4. App sends the content to the Debunking API.
5. Debunking API analyses the content and returns:
    a. Disinfoscore (risk score)
    b. Short classification (e.g., likely disinformation / unclear / likely reliable)
    c. Key signals and explanations
6. App displays a fact-check summary card, including:
    a. Visual indicator of risk (speedometer-style gauge segmented into green / yellow / orange / red ranges, with a needle/marker indicating the Disinfoscore)
    b. One-sentence verdict

Funded by the European Union

c. Percentage of how similar and opposing the content is with known credible sources or previously validated claims

7. User can optionally report the content, if they are logged in.

Postconditions:

- User has an immediate, understandable assessment of the content's reliability.
- A log entry is stored in the user's account, if they are logged in.

Alternative / exception flow:

- 1a – No network: App informs the user and offers to retry later.
- 4a – User chooses to continue in background: After submitting the URL/text (step 5), the app displays a non-blocking "Analysis started" state with a short message (e.g., "We're debunking this in the background.") and a "Back to Home" button.
  - 4a-1 – If the user taps "Back to Home", the app immediately navigates to the main screen while the Debunking API request continues in the background.
  - 4a-2 – The app creates a pending history entry (e.g., status = "Processing…") so the user can track the request from Profile → History.
  - 4a-3 – When results are ready, the app notifies the user (in-app banner and/or OS notification, if enabled).

UC-09 – Check Screenshot or Image

Primary actor: User
Goal: Evaluate an image-based post (meme, screenshot of tweet, infographic, etc.) for potential disinformation.
Scope: App + Debunking API
Precondition: User has a screenshot or image stored on device or recently captured.
Main success scenario:

1. User opens the app and selects "Upload your File".
2. User chooses an image from the gallery.
3. App securely uploads the image to the Debunking API
4. API analyses embedded text and visual content, compares against knowledge graphs / known cases.
5. API returns Disinfoscore, classification, and matched known disinformation narratives (if available).
6. App displays result with:
   a. Risk indicator
   b. Explanation: why the image might be misleading (e.g., context mismatch, manipulated visual).

7. User can optionally report the content, if they are logged in.

Postcondition: The image is assessed

Alternative / exception flows:

- 2a – Permission denied: If the app doesn't have gallery/camera access, it prompts the user to enable
- 4a – User chooses to continue in background: After submitting the image file, the app displays a non-blocking "Analysis started" state with a short message (e.g., "We're debunking this in the background.") and a "Back to Home" button.
    - 4a-1 – If the user taps "Back to Home", the app immediately navigates to the main screen while the Debunking API request continues in the background.
    - 4a-2 – The app creates a pending history entry (e.g., status = "Processing…") so the user can track the request from Profile → History.
    - 4a-3 – When results are ready, the app notifies the user (in-app banner and/or OS notification, if enabled).
- 5a – Image too large or unsupported: App asks the user to crop/compress

UC-10 – Check Video or Audio Clip

Primary actor: User

Goal: Check short video or audio clips (e.g., speech snippets, short news videos) for potential disinformation.

Scope: App + Debunking API

Precondition: User has a short video/audio file.

Main success scenario:

1. User selects "Upload your File".
2. User uploads a clip from the device.
3. App sends the file to the API.
4. API performs speech-to-text and/or frame analysis, then compares content with knowledge graphs and debunked cases.
5. API returns Disinfoscore, a summary of the claims detected, and relevant debunked cases.
6. App presents:
    a. A verdict
    b. Key claims with per-claim reliability indicators.
7. User can optionally report the content, if they are logged in.

Postcondition: User receives an assessment of the clip's reliability.

Alternative flows:

- 2a – File too long: App prompts user to trim to a shorter duration (within policy limits).
- 4a – Speech recognition fails: App informs user and suggests manual text entry.

- 4b – User chooses to continue in background: After submitting the video or audio clip, the app displays a non-blocking "Analysis started" state with a short message (e.g., "We're debunking this in the background.") and a "Back to Home" button.
    - 4b-1 – If the user taps "Back to Home", the app immediately navigates to the main screen while the Debunking API request continues in the background.
    - 4b-2 – The app creates a pending history entry (e.g., status = "Processing…") so the user can track the request from Profile → History.
    - 4b-3 – When results are ready, the app notifies the user (in-app banner and/or OS notification, if enabled).

UC-11 – Record Live Audio for Verification

Primary actor: User who is currently hearing suspicious content (e.g., on TV, radio, public announcement).
Goal: Capture live audio quickly and submit it for disinformation assessment.
Scope: Smartphone app + Debunking API
Preconditions:

- User is logged in.
- App has microphone permission (or will request it).
- App has network connectivity to complete the check.

Main success scenario:

1. From the home screen, user selects "Record your Sound".
2. If microphone permission is not yet granted, the app requests it with a clear explanation.
3. App shows a simple recording interface with a single prominent "Record" button.
4. User taps "Record"; the app starts recording and displays clear visual feedback (timer, waveform or recording indicator).
5. User taps again to stop recording.
6. App switches to a review screen showing the recorded clip (duration) and two buttons: "Dismiss" and "Accept".
7. User taps "Accept".
8. App checks that the recording meets the minimum/maximum duration.
9. App offers an optional 'Recording name' field.
10. App sends the audio (or extracted features) securely to the Debunking API.
11. Debunking API performs analysis (e.g., speech-to-text + content checking) and returns a Disinfoscore, classification and explanations.
12. App displays the standard result screen:
    a. Visual risk indicator
    b. One-sentence verdict
    c. Key signals / explanations
13. User can optionally report the content, if they are logged in.

Funded by the European Union

Postconditions:

- Live audio has been captured, analysed, and presented with a clear verdict.
- A history entry is stored in the user's account, if they are logged in.

Alternative / exception flows:

- 2a – Microphone permission denied: App explains why the permission is needed, offers a link to system settings, and returns to home if the user still denies.
- 5a – Recording too short: At step 8, app detects duration below threshold; user sees a clear message ("The recording is too short for us to estimate properly if it is misinformation or not. Please record a bigger sample.") and can retry; too-short samples are discarded according to privacy rules.
- 7a – User dismisses recording: User taps "Dismiss"; app discards the recording; no audio is uploaded.
- 9a – No network / API unreachable: App informs the user that analysis cannot be completed and offers to retry.
- 10a – User chooses to continue in background: After submitting the URL/text (step 5), the app displays a non-blocking "Analysis started" state with a short message (e.g., "We're debunking this in the background.") and a "Back to Home" button.
    - 10a-1 – If the user taps "Back to Home", the app immediately navigates to the main screen while the Debunking API request continues in the background.
    - 10a-2 – The app creates a pending history entry (e.g., status = "Processing…") so the user can track the request from Profile → History.
    - 10a-3 – When results are ready, the app notifies the user (in-app banner and/or OS notification, if enabled).

UC-12 – Check Offline / Physical Content (Photo or QR Code)

Primary actor: User

Goal: Evaluate information from printed flyers, posters, newspapers, etc., related to targeted disinformation domains (Ukraine war, climate change, etc.).

Precondition: User is physically in front of printed or offline content.

Main success scenario:

1. User opens the app and selects "Scan with AR".
2. User takes a photo of the flyer/poster or scans a QR code printed on it.
3. App extracts text (OCR) or URL.
4. App sends extracted content to the Debunking API.
5. API returns Disinfoscore and related explanations / cases.

6. App displays results as an AR overlay on top of the captured image, without navigating to the standard results screen.
7. User can optionally report the content, if they are logged in.

Postcondition: Offline content has been transformed into analyzable text/URL and evaluated.

Alternative / exception flow:

- 4a – User chooses to continue in background: After submitting the URL/text (step 5), the app displays a non-blocking "Analysis started" state with a short message (e.g., "We're debunking this in the background.") and a "Back to Home" button.
    - 4a-1 – If the user taps "Back to Home", the app immediately navigates to the main screen while the Debunking API request continues in the background.
    - 4a-2 – The app creates a pending history entry (e.g., status = "Processing…") so the user can track the request from Profile → History.
    - 4a-3 – When results are ready, the app notifies the user (in-app banner and/or OS notification, if enabled).

UC-13 – Report Suspicious Content to Disinfopedia

Primary actor: User

Goal: Allow users to flag content for human fact-checkers when automated assessment is inconclusive or content appears especially harmful.

Scope: App + Disinfopedia backend + expert team

Precondition: User has selected some content or has a new item to report.

Main success scenario:

1. User taps the "Report" button.
2. App pre-fills report with any available content (URL, text, image, video) and metadata (language, platform type, timestamp).
3. User optionally provides a short description (why they think it is suspicious, perceived target group, where they saw it).
4. App sends the report to the Disinfopedia backend.
5. Backend assigns an internal ID and queues it for expert review.
6. App shows confirmation.

Postcondition: A new report exists in the system with a unique identifier and minimal necessary personal data.

Alternative flow: 4a – Report sending fails: App informs the user that the operation failed, encouraging them to try again.

UC-14 – View Profile & Usage Statistics

Funded by the European Union

Primary actor: User
Goal: Recognize their account area and understand how actively they use AI4Debunk and how often they encounter misinformation.
Scope: Smartphone app + User profile backend
Preconditions:

- User is logged in.
- User has performed at least one check (for non-zero stats).

Main success scenario:

1. User opens the profile screen from navigation (e.g., hamburger menu or header icon button).
2. App fetches user profile information from backend (or uses cached data).
3. Profile screen displays:
    a. Profile image (or default avatar)
    b. Username
    c. Registered email
4. Same screen displays key stats such as:
    a. Total files analyzed
    b. Total reports submitted
    c. Average probability of being fake (or related high-level metric)
5. User can briefly review their stats and identity information to confirm they are in their own account area.

Postcondition: User has a clear overview of their account identity and engagement level.
Alternative / exception flows:

- 2a – Backend unavailable: App shows last known cached profile and stats, with a banner that fresh data could not be retrieved.
- 3a – Some fields not configured (e.g., no avatar): App shows sensible defaults (placeholder image, “Not set”) without breaking the layout.

UC-15 – View Personal Check History

Primary actor: User
Goal: Review previously checked content and revisit results.
Precondition: User has performed at least one check.
Main success scenario:

1. User opens their profile screen.
2. App lists previous checks with minimal metadata (file name, date/time of upload, file size, file type, probability of being fake).
3. User selects an entry to see the full result.

Postcondition: The system has retrieved and displayed the full result of the selected past check to the user.

UC-16 – Search and Filter Personal Check History

Primary actor: User
Goal: Quickly find previously analysed items using search and file-type filters.
Scope: Smartphone app + History backend
Preconditions:

- User is logged in.
- User has performed multiple checks; history is non-empty.

Main success scenario:

1. User navigates to their profile and scrolls to the history section (list of analyzed files / items).
2. Above the list, app displays:
    a. A search bar.
    b. Clickable filter chips for file types (e.g., “Images”, “Audio”, “Video”, “Text/URL”).
3. User taps one or more filter chips (e.g., “Images”).
4. App updates the list to show only items of the selected type(s), maintaining key metadata (file name, date/time, size, type, probability of being fake).
5. User enters part of a file name into the search bar.
6. App filters the (already filtered) list in real time to match the search string.
7. User taps a specific entry to open the full result details.

Postcondition: User has efficiently located a specific past check and viewed its detailed assessment.
Alternative / exception flows:

1 5a – No items match filters / search: App shows an empty state message (e.g., “No results match your search”) and a clear way to clear filters.

2 2a – History temporarily unavailable (backend issue): App shows a friendly error and suggests retry, while preserving local cache if available.

Funded by the European Union

3. HOW THE APP WILL BE USED

3.1 First Contact: Installation, Registration and Login

A typical journey starts when a user installs the AI4Debunk app on their smartphone and opens it for the first time. Since many features depend on a persistent profile and synchronized history, the user is prompted to create an account or log in. New users are guided through a short registration process where they provide basic details such as name and email, choose a password, or opt for a social login with an identity provider like Google. Once the authentication backend has validated the information, an account is created and the user is automatically logged in.

Returning users see a simple login screen offering both traditional email/password access and identity provider buttons. When they authenticate successfully, a secure session token is stored on the device, so that they can come back later without re-entering credentials each time, subject to reasonable security timeouts. If they forget their password, they can trigger a password reset flow, where they can set a new password while preserving all their history.

In addition, users can log out from the profile screen when using a shared device, ensuring that their personal checks and reports are not visible to others. A data deletion path is also available. Before deletion, the app clearly explains what will happen to the user's data and what may remain in anonymized form for analytics or legal reasons, and then sends a secure deletion request to the backend.

3.2 Onboarding, Consent, and the Home Screen

On first launch, users can either sign in / sign up or continue as a Guest. Guest mode supports running checks (debunking) but does not provide access to profile, history, personal usage statistics, or reporting to experts. In either case, they are not thrown directly into a complex interface. Instead, they see an onboarding sequence that briefly explains what AI4Debunk is, its purpose, and how it supports them in identifying misleading or false information. The onboarding screens highlight the four core actions: pasting texts or URLs, uploading files such as images or videos, recording live audio, and scanning physical items using the camera.

A key part of this introductory journey is transparency. One or more screens describe which data is processed by the app, for what purposes, and which partners may be involved. The language is accessible, avoiding dense legal jargon, so that users can provide informed consent rather than blindly accepting terms they do not understand. The app records when the user gave consent and which version of the policies was in force at that time, supporting future audits and

compliance. If the user declines, the app explains that core functionalities cannot be provided without certain permissions, and may offer a highly restricted mode or guide them towards exiting the app.

Once onboarding is complete, the user lands on the home screen. The design is intentionally simple and action-oriented. Four large, clearly labeled tiles lead to the main verification flows: “Paste your Text or URL”, “Upload your File”, “Record your Sound”, and “Scan with AR”. The same screen fetches from the backend a global statistic and a breakdown by file type showing how many pieces of content have been identified as misinformation. This high-level metric helps users understand that they are participating in a broader, collective effort rather than acting alone. If the analytics service is temporarily unavailable, the app falls back to cached data or hides the metric and shows a brief message, while keeping all main actions fully usable.

3.3 Everyday Use Case: Checking Online Content (URL or Text)

A very common way the app will be used is when someone encounters a suspicious article, social media post, or message. Imagine a user scrolling through their social feed and spotting a claim about climate change or an election that feels exaggerated or inconsistent. They open AI4Debunk and tap “Paste your Text or URL” on the home screen. In the input view, they paste either the link to the content or a text snippet.

When they tap the button to proceed, the app sends this content securely to the Debunking API. The backend analyses the text using a combination of AI models, knowledge graphs, and previously debunked cases. It returns a structured result that includes a risk score (Disinfoscore), a short verdict such as “likely disinformation” or “likely reliable”, and key explanatory signals.

The user sees a results screen designed for non-experts: a prominent visual indicator (a speedometer-style gauge segmented into green / yellow / orange / red ranges, with a needle/marker indicating the Disinfoscore) shows the risk level at a glance, accompanied by a single-sentence conclusion. Beneath this summary, users who want more detail can expand the view to see which elements of the text triggered suspicion, how similar the content is to known credible or debunked sources, and links to related cases. This layered approach supports both quick decisions and deeper understanding.

If the network is unavailable or the API cannot be reached, the app clearly informs the user instead of remaining stuck in a loading state, and offers the option to retry. Regardless of outcome, a log entry is stored in the user’s profile so they can revisit the result later.

3.4 Checking Files: Images, Screenshots, Videos and Audio Clips

Another frequent use scenario involves content that is not purely textual. Users constantly encounter memes, infographics, screenshot-based posts, or short videos that may contain misleading information. In these situations, they select “Upload your File” from the home screen. The app opens the device’s file explorer or gallery, subject to the appropriate permissions being granted.

If the user chooses an image, the app uploads it to the Debunking API. The backend analyses the image for manipulations or context mismatches, and compares the content to known disinformation narratives. The app then presents a familiar result screen: a risk indicator, a short verdict, and an explanation tailored to images, such as highlighting that the text was originally posted in a different context, or that the image has been used in unrelated incidents in the past.

For video or audio files, the process is similar but technically more involved. The user again selects “Upload your File”, this time choosing a short clip. The app may extract the audio track or representative frames and send them for analysis. The backend performs speech recognition, claim extraction, and cross-checking against knowledge bases and previously debunked material. The user receives not only a global verdict but also a breakdown of key claims in the clip, each with its own reliability indicator. They can see, for example, that a particular statement about an event date is demonstrably false, while other parts of the clip are considered plausible.

If a file is too long or uses an unsupported format, the app explains the limitation and guides the user to trim the clip or convert it, rather than simply failing silently. All such checks are reflected in the user’s history so that they can be revisited later.

3.5 Capturing Live Audio in the Moment

AI4Debunk is also designed for situations where misinformation is encountered in real time, such as listening to a talk show on the radio, a political speech on television, or a public announcement. In these moments, the user may not have a ready-made file; instead, they tap “Record your Sound” on the home screen to capture what they are hearing.

If microphone permission has not yet been granted, the app requests it with a clear explanation of why it is needed. Once permission is in place, the user sees a minimalist recording screen with a single large button. Tapping it starts recording, with a simple visual confirmation such as a timer

Funded by the European Union

or waveform. Tapping again stops recording and brings the user to a review screen that shows the duration of the clip and offers two options: dismiss or accept.

If the user dismisses the recording, the audio is discarded according to the app's privacy rules and is never uploaded to the server. This is important for protecting users who might accidentally capture private conversations or irrelevant content. If the user accepts, the app first checks whether the recording is longer than a minimum threshold needed for meaningful analysis. Very short clips generate a friendly error message explaining that the sample is insufficient and inviting the user to record again. Too-short recordings are likewise discarded rather than being stored.

For suitable recordings, the app sends the audio to the Debunking API, which performs speech-to-text and content analysis. The resulting verdict, explanations, and risk score are then displayed on the same standard result screen used for other checks, giving users a consistent mental model across different types of input. A history entry is also created, so the user can later revisit what was said in that broadcast and how it was assessed.

3.6 Checking Physical and Offline Material

Disinformation is not limited to digital platforms. Flyers, posters, leaflets, and even newspaper adverts can carry misleading narratives. AI4Debunk addresses this by allowing users to scan physical content using the "Scan with AR" option on the home screen.

When the user chooses this entry point, the app activates the camera (after requesting permission if needed). The user points their device at a flyer or poster and takes a photo, or scans a QR code printed on the material. The app then extracts text using OCR or decodes the QR code into a URL. This extracted content is sent to the Debunking API in a similar way to the online text or link checks.
The result is visualized in an augmented reality style: a simplified verdict (for example, a colored badge indicating risk) can be overlaid on the captured image, with detailed explanations and related cases accessible below. This makes it intuitive for users to see which specific poster or flyer was problematic and why. The ability to scan offline content is particularly relevant in targeted disinformation campaigns around topics like elections, public health, or climate change, where printed material may play a significant role.

3.7 Reporting Suspicious Content to Experts

Funded by the European Union

AI4Debunk is not only about automated verification. Sometimes the AI-based assessment is inconclusive, or the user believes that the content is particularly harmful and should be looked at by human fact-checkers. In those cases, the app offers a “Report” button from the results screen.
When the user taps “Report”, the app prepares a report package that includes the content itself (URL, text, image, video, or audio) along with minimal metadata such as language, platform type, and timestamp. The user can optionally add a short note describing why they found the content suspicious, who it seems to target, or where they saw it. After the user submits, the report is sent to a backend system (e.g., Disinfopedia) that assigns an ID and queues it for review by experts. The app then displays a clear confirmation that the report has been successfully submitted.
From the user’s perspective, this means that, beyond getting their own answer, they are contributing to the improvement of the broader disinformation monitoring ecosystem. Their reports can help identify new narratives, refine AI models, and support public responses to emerging campaigns.

3.8 Profile, Usage Statistics, and History

The profile section of AI4Debunk gives users a sense of their identity and activity within the app. When they open their profile from the menu or header, they see essential information: profile image or avatar, username, registered email, and an indication that a password is set (without revealing it).
Below this, the app presents a compact summary of their usage statistics: how many items they have analyzed in total, how many reports they have submitted, and an aggregate measure such as average probability of being fake among items they checked. These metrics help users reflect on their habits: whether they are frequently encountering questionable content, whether they are proactively reporting it, and how engaged they are in using the tool.

The same area provides access to their personal check history. The app lists previously analyzed items in chronological order, each row showing basic metadata like file or content label, date and time of analysis, file size (when relevant), type (image, video, audio, text/URL), and the risk percentage or similar indicator assigned during analysis. Tapping any entry takes the user back to the detailed result screen, allowing them to recall why something was assessed as it was.
As users accumulate more checks, the history interface includes a search bar and filter chips (e.g., “Images”, “Audio”, “Video”, “Text/URL”). Users can tap one or more filters to narrow the list to specific types of content and then type part of a title or filename in the search bar to quickly locate a past check. If no results match, the app shows a friendly empty state and makes it easy to clear filters.

Funded by the European Union

The history and statistics features allow AI4Debunk to function not just as a momentary checker but also as a personal archive of verification activity, which can be valuable for journalists, researchers, educators, or engaged citizens monitoring specific narratives over time.

3.9 Privacy, Permissions, and Data Lifecycle in Everyday Use

Throughout these workflows, AI4Debunk maintains a strong focus on privacy and control. The app requests permissions (microphone, camera, storage) only when needed and always with a clear justification, so that users understand the purpose of each access. If permissions are denied, the app provides guidance on how to enable them later and gracefully degrades functionality rather than crashing or behaving unpredictably.

Data lifecycle considerations are embedded into everyday interactions. For example, recordings that are dismissed or fall below the minimum duration are discarded rather than uploaded. Account deletion flows clearly communicate which personal data is erased and which anonymized analytics may persist. Consent to data processing is captured explicitly, with versioning, and can be revisited if policies change.

From the user's perspective, this means they can rely on AI4Debunk as a trustworthy companion: one that helps them navigate complex information environments, contributes to collective fact-checking efforts, and respects their autonomy and privacy. Day to day, they will use the app to quickly check suspicious posts, flyers, or broadcasts; to report particularly harmful cases; and to understand, via clear explanations, how to recognize manipulative patterns on their own.

In summary, AI4Debunk will be used as a multi-channel verification hub in the user's pocket: a tool they can reach for whenever they encounter questionable claims online or offline, a gateway to expert-led debunking ecosystems, and a practical support tool for strengthening media literacy in a world saturated with information.

In a future phase, once the mobile application has been fully developed, the focus group sessions will explicitly target citizens with vulnerabilities, including seniors, adolescents aged 13 and above, and individuals with a low educational background. The objective will be to assess usability and accessibility aspects under realistic usage conditions. To support a standardized and comparable evaluation of the user experience, we will administer the System Usability Scale (SUS) questionnaire at the end of each session. The SUS is a 10 item questionnaire with 5 response options.

- I think that I would like to use this system frequently.

Funded by the European Union

- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

The SUS uses the following response format:

- 1 - Strongly Disagree
- 2
- 3
- 4
- 5 - Strongly Agree

4. SMARTPHONE APP DESIGN AND MOCKUPS

This section presents the AI4Debunk smartphone app design and mockups as a coherent, end-to-end user experience, mapping the main functional components into concrete screens and navigation steps. It starts from the authentication and account entry points, including Sign Up (with explicit acceptance of Terms & Conditions / Privacy Policy and optional social providers) and Login, so that access to the app is both secure and familiar to users.

It then introduces the onboarding/tutorial flow, implemented as a short swipeable sequence that explains the purpose of AI4Debunk and the supported content types, before the user proceeds to the core app experience. From there, the mockups focus on the main screen as the central hub, offering four primary debunking actions (Paste your Text or URL, Upload your File, Record your Sound, and Scan with AR) and showing how users can reach their account/profile area via the UI navigation.

Finally, the section illustrates the debunking workflow itself across modalities (text/URL input, file selection, live audio recording with optional naming, and AR scanning), including intermediate states such as loading/cancel, and a unified results screen that summarizes the classification and signals and allows users to continue exploring (“Back to Home”) or follow up (e.g., reading more after AR overlay results). In the broader app structure, these flows connect back to the user’s account area, where profile information, usage statistics, and debunking history support revisiting past checks (with search/filtering when needed).

Funded by the European Union

Sign Up

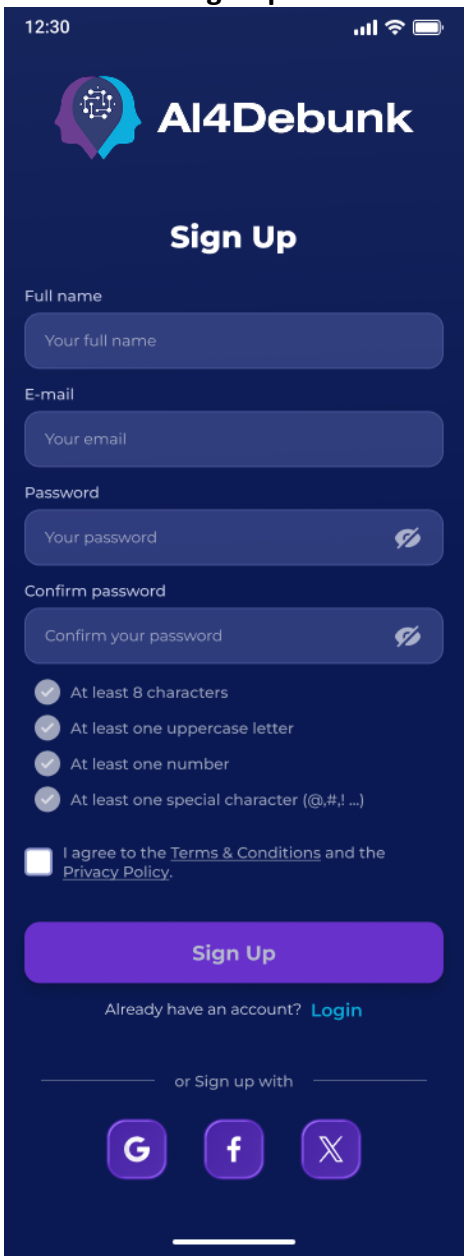


FIGURE 1: SIGN UP SCREEN

The user can create a new account by entering their full name, email address, and a password, then re-entering the password to confirm it. To complete registration, they must tick the checkbox confirming they agree to the Terms & Conditions and the Privacy Policy (both available as links). After filling everything in, they tap Sign Up to submit and create the account. If they already have an account, they can tap Login instead to go to the sign-in screen. Alternatively, the user can sign up using an existing external account (e.g., Google or Meta/Facebook) by selecting the corresponding provider button.

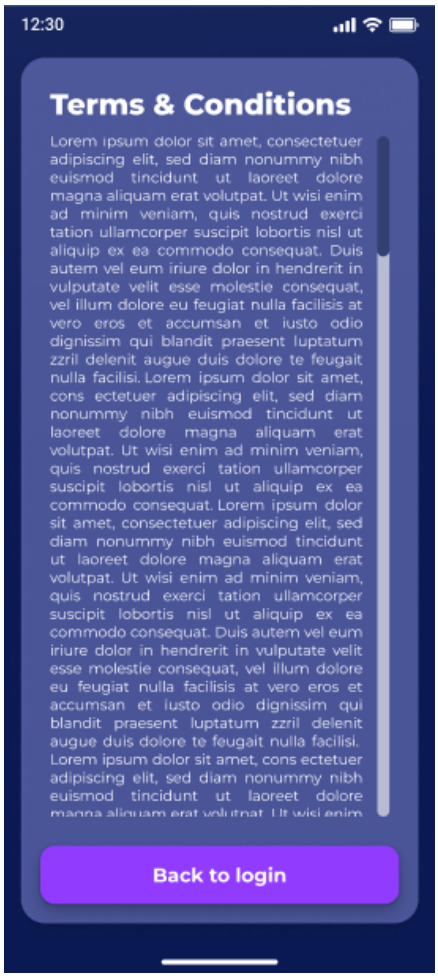


FIGURE 2: TERMS & CONDITIONS SCREEN

Tapping the Terms & Conditions (or Privacy Policy) hyperlink opens a dedicated screen that displays the full legal text in a scrollable view. At the bottom, the user can press a button to return to the registration screen and continue the sign-up process.

Login

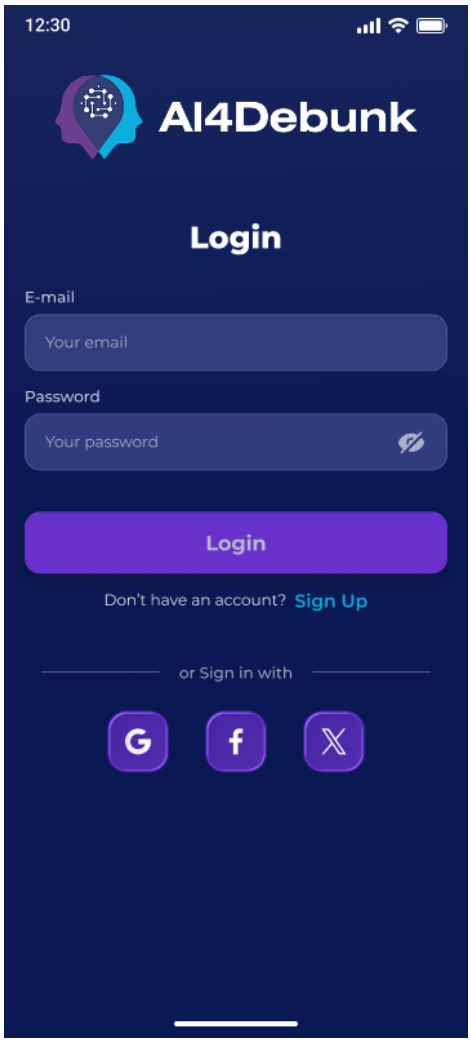


FIGURE 3: LOGIN SCREEN

The user logs in by entering their email address and password, then tapping Login to access the app. If they don't have an account yet, they can select Sign Up to go to the registration screen. Alternatively, the user can sign in using an existing external account (e.g., Google or Meta/Facebook) by tapping the corresponding provider button.

Onboarding

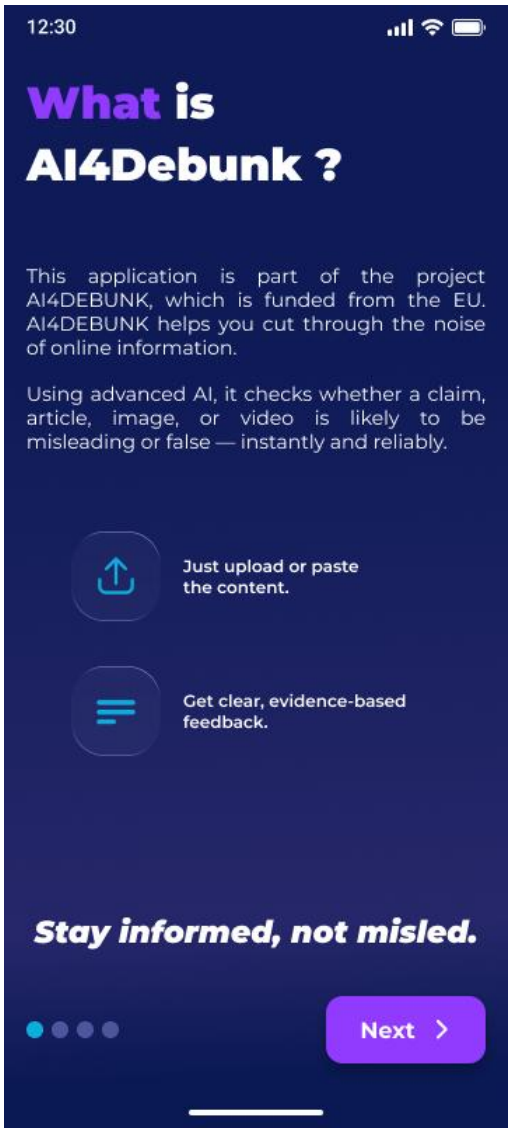


FIGURE 4: ONBOARDING SCREEN (STEP #1)

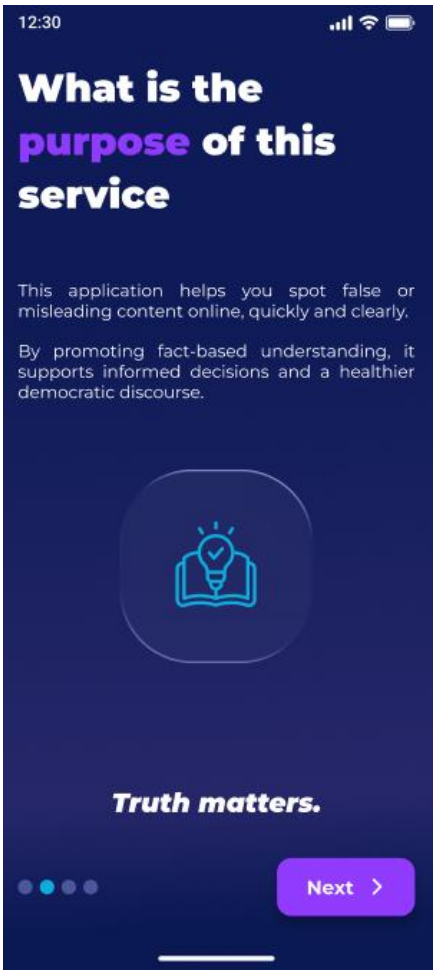


FIGURE 5: ONBOARDING SCREEN (STEP #2)

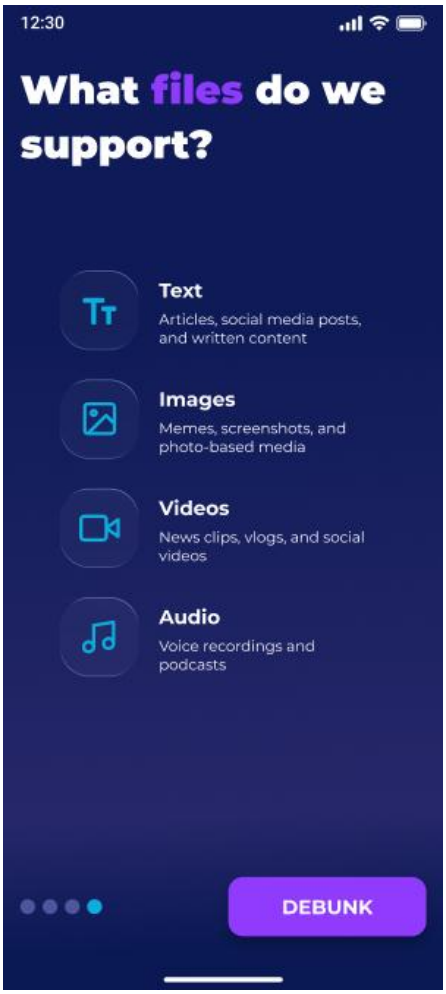


FIGURE 6: ONBOARDING SCREEN (STEP #3)

During onboarding, the user is presented with a short, swipeable introduction to the app across three screens. The first screens explain what AI4Debunk is and what the service is meant to achieve, with a “Next” button (and progress dots) to move forward. The final screen summarizes the types of content the app supports (text, images, videos, and audio) and the user taps “DEBUNK” to finish onboarding and proceed into the main app experience.

Main Screen

Funded by the European Union

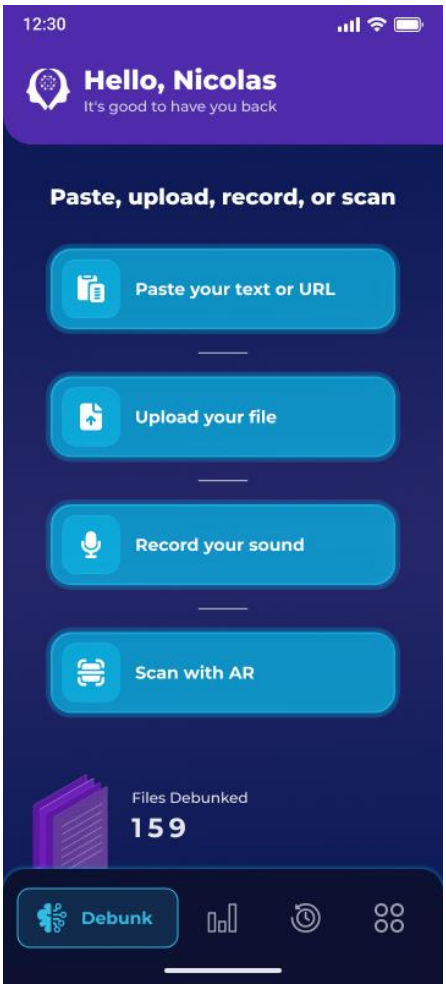


FIGURE 7: MAIN SCREEN

The main screen acts as the app's starting point after login/onboarding. It greets the user and provides four primary ways to begin a debunking check: Paste your text or URL, Upload your file, Record your sound, or Scan with AR. The user selects one of these options to start a new analysis, can access their account/profile from the top-right icon, and can navigate to other areas of the app using the bottom navigation bar. The user is also presented with the total number of debunked files.

Paste your text or URL

Funded by the European Union

FIGURE 8: ANALYZE TEXT OR URL SCREEN

When the user chooses “Paste your text or URL” from the main screen, they are taken to a screen where they can submit content for analysis. At the top, the app indicates the current focus topics (e.g., Climate Change and Russia-Ukraine War). The user then selects whether they want to analyze text or a URL, pastes or types the content into the input area, and presses the Debunk button to start the debunking process.

Funded by the European Union

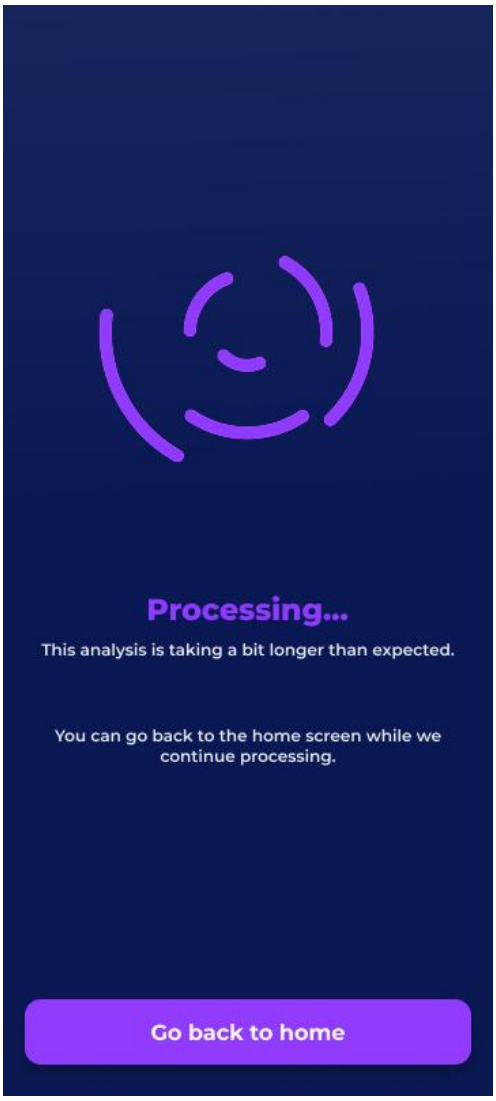


FIGURE 9: LOADING SCREEN

After the user taps Debunk, the app displays a loading screen showing that the analysis is in progress. A progress indicator communicates that processing is ongoing, and the user can press the “Go back to home” button to return to the main screen instead of waiting for the results.

Upload your file

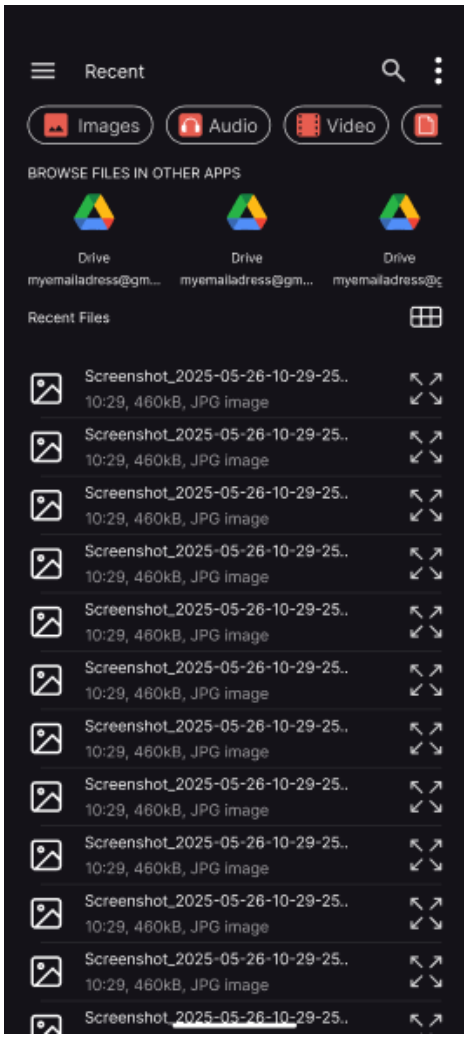


FIGURE 10: FILE PICKER SCREEN

When the user selects “Upload your file” from the main screen, the app opens the device’s file picker so they can choose content to analyze. The user can browse recent files or navigate storage locations (e.g., local folders or cloud drives), filter by file type (such as images, audio, or video), and select the desired file to return to the app and continue with the debunking process.

Record your sound

FIGURE 11: SOUND RECORDING SCREEN

When the user selects “Record your sound” from the main screen, they are taken to a recording page where they can capture audio for analysis. The screen explains that the recording should relate to the supported topics, and the user starts/stops recording using the main record button while a timer shows the recording duration. When they finish, the app moves to the next step where the user can optionally name the recording before it is uploaded for debunking.

Funded by the European Union

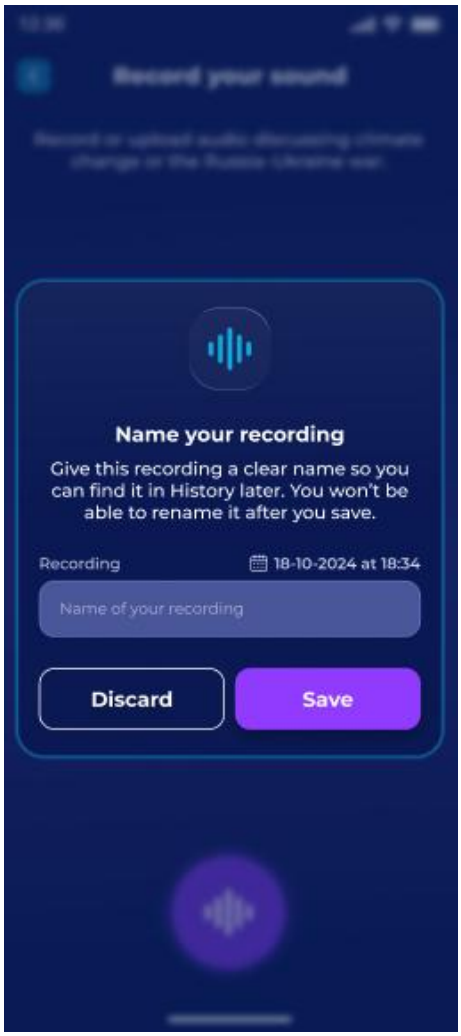


FIGURE 12: OPTIONAL FILE NAMING POPUP

After recording, the app prompts the user to optionally give the recording a clear name so it can be easily found later in their history (the name cannot be changed afterwards). The user types a name and taps “Save” to confirm and continue with uploading the audio for analysis, or taps “Discard” to skip naming and return without saving the recording.

Results Screen

Funded by the European Union

FIGURE 13: RESULTS SCREEN

After the analysis is completed (regardless of whether the user pasted text/URL, uploaded a file, recorded audio, or used AR), the app shows a results screen summarizing the outcome. The user sees the analyzed item, an overall classification (e.g., Likely Disinformation), and key metadata such as the title and source/URL (when available). The screen also explains why the content was flagged by listing the main signals detected, and the user can press the “Back to Home” button to go back to the home screen.

Scan with AR

Funded by the European Union

FIGURE 14: SCAN WITH AR SCREEN

When the user selects “Scan with AR”, the app opens the camera and asks them to point it at printed or on-screen content related to the supported topics. A framing guide helps the user align the content in view, and they tap the capture/scan button to analyze what’s on the screen.

Funded by
the European Union

FIGURE 15: RESULTS OVERLAY

After scanning, the app displays the result in a bottom sheet. A label indicates the assessment (e.g., “This looks fake!”), while the user can tap “Report result” to report the result or tap “Back to home” to return to the main screen.

5. FOCUS GROUP

To validate the functional definition of the AI4Debunk mobile application and ensure that the specification reflects cross-work-package constraints, the requirements documented in this deliverable were refined through two internal focus-group consultations. These sessions utilized the evolving set of user stories, use cases and end-to-end user journeys as shared discussion material.

Focus Group 1 - Work package leaders: This focus group brought together the leaders of the relevant work packages to review the proposed app scope, confirm feasibility against the available backend services (e.g., authentication, the Debunking API and reporting to experts), and identify interdependencies, assumptions and risks early. The discussion focused on the completeness of the verification modalities (text/URL, file upload, live audio and AR-supported scanning), the clarity of the result presentation (Disinfoscore, verdict and explanations), and the alignment with privacy-by-design principles, especially regarding consent, data minimisation and retention.

Focus Group 2 - Consortium members and the project's Steering Committee: A second consultation was conducted at consortium level, involving AI4Debunk partners and the Steering Committee, to validate that the app definition matches the overall project objectives and provides a coherent, citizen-oriented experience. Particular attention was paid to usability and wording (so that non-experts can understand outcomes at a glance), transparency and informed consent during onboarding, and the role of reporting and escalation workflows as part of the wider AI4Debunk ecosystem.

Across both focus groups, feedback was consolidated into actionable refinements that are reflected throughout this deliverable. Concretely, the consultations helped prioritise the core flows, clarify edge cases and exception handling, and strengthen the consistency between requirements, the usage narrative and the dedicated security/privacy considerations.

6. TESTING AND DEBUGGING

Testing and debugging will be conducted as an iterative activity throughout implementation, to ensure that the app's end-to-end user journeys (authentication, onboarding/consent, multi-modal debunking flows, reporting, profile/history) remain correct, usable, and robust under realistic conditions. This section builds on the project's validation mindset already reflected in the internal focus-group consultations, which are intended to refine functionality and usability prior to wider release.

From a technical perspective, the testing approach will combine automated and manual validation. Automated tests will be used to continuously verify critical behaviors (e.g., navigation, permission handling, error states, and security-sensitive flows), leveraging unit, widget, and integration testing to prevent regressions as the codebase evolves. Manual testing will complement automation by covering device-specific and context-dependent scenarios

Funded by the European Union

(camera/microphone behavior, file picker interoperability, offline/poor connectivity cases, and usability of result explanations).

To support controlled validation and reliable debugging, three distinct environments will be used:

- **Development environment (internal)**: This environment will be used by the development team for immediate testing and rapid debugging. It will host the latest builds and enable fast iteration (frequent deployments, verbose logging where appropriate, and diagnostic tooling) so that defects can be reproduced quickly and fixed early. The development environment will also be used to validate new features incrementally against the defined use cases before they are promoted to wider testing.
- **Quality environment (focus groups)**: This environment will be reserved for structured testing by the focus groups and other non-developer testers, providing a stable and representative setup for usability and acceptance evaluation. It will run release-candidate builds and will be used to execute test scripts derived from the documented user journeys, while collecting feedback and issues in a controlled manner. Importantly, the quality environment will allow the consortium to validate user experience (including standardized usability instruments where applicable) before public release.
- **Production environment (public release)**: The production environment will be activated when the app is made publicly available through the official app distribution channels (Apple App Store and Google Play Store). In production, the emphasis shifts from exploratory testing to operational assurance: monitoring stability, detecting crashes and performance issues, and applying a disciplined hotfix process when needed. Production configurations will remain strictly separated from non-production environments to avoid accidental exposure of test settings, test data, or overly verbose diagnostics.

Across all environments, debugging will follow a consistent lifecycle: capture and reproduce (using logs and clearly defined steps), isolate root cause, apply a fix, and confirm through regression testing before promotion to the next environment.

7. SECURITY CONCERNS

Security and privacy are central to the success of AI4Debunk, because the system encourages users to submit content they encounter in their online and offline lives, often related to politically sensitive topics, health, elections, or personal communications. While the functional documentation already anticipates several privacy-friendly behaviours (e.g. discarding too-short audio recordings, explicit consent, account deletion), a deeper analysis reveals a number of security concerns, with user anonymity and de-identification at their core.

Below, we examine these concerns systematically across the main layers of AI4Debunk: identity and accounts, content submission and analysis, logging and analytics, reporting to experts, device-level security, and data lifecycle. The emphasis is on risks to user anonymity and linkability of actions, but we also touch on broader security implications where they interact with anonymity.

7.1 Identity, Registration and Persistent Profiles

AI4Debunk is designed around persistent user accounts, created through email/password registration or identity providers such as Google, and used to synchronize checks and reports across devices.

From a security and anonymity perspective, this raises several concerns:

1. **Loss of true anonymity by design**
   The core workflows assume that most users are logged in while using the app, and that checks and reports are tied to a persistent profile. This allows rich features like personal history and usage statistics, but it also means that AI4Debunk is, by default, a pseudonymous or even fully identified system rather than an anonymous one. The email address, and possibly the real name provided during registration, serve as stable identifiers.
2. **Third-party identity providers and cross-service tracking**
   When a user chooses to log in via an identity provider (e.g. Google), that provider learns that this account is using AI4Debunk, and potentially when. This weakens anonymity because:
   - The IdP can link AI4Debunk usage to its broader profile of the user across services.
   - Any compromise or misuse at the IdP side can indirectly expose the association between real-world identity and the AI4Debunk account.
3. **Account lifecycle and residual identifiers**
   The documentation includes an account deletion flow where users can withdraw from the service and have personal data erased, while some anonymised analytics may be retained. The key anonymity concern is whether “anonymised” truly means that no re-identification is possible:
   - If debunk checks, timestamps, and content characteristics are stored, these may be unique enough to re-link to a user, especially for rare or sensitive cases.

- If account IDs or hashed identifiers are retained for analytics, they may still be linkable under certain threat models (e.g. if combined with IP logs or external datasets).

4. **Session persistence on shared devices**
   AI4Debunk is meant to remember authenticated sessions between uses, subject to reasonable timeouts, so users do not have to log in every time. On shared or poorly protected devices, this creates a confidentiality and anonymity risk:
   - A second person using the same device could access the user's history of checks and reports, potentially revealing what topics they are concerned about, what posters they scanned, or which broadcasts they recorded.
   - If the user forgets to log out, the system state may be indistinguishable from being the original user, effectively eroding anonymity in a local, physical sense.

7.2 Anonymity Risks in Content Submission and Analysis

AI4Debunk accepts a wide variety of content: URLs, text snippets, images and screenshots, videos, audio clips, live recordings, and photos of physical material (posters, flyers, QR codes).

This richness is crucial for functionality but introduces significant anonymity concerns.

7.2.1 URLs and Text Snippets

When a user checks a URL or text snippet, the content itself may:

- Contain personal data (names, private conversations, email addresses).
- Be drawn from closed groups or semi-private spaces (e.g. a private chat screenshot shared via copy-paste).

If the system stores raw content or detailed logs, then:

- The combination of content + timestamp + profile can uniquely identify the user's online circles or behaviour.
- If the same URL or text is only accessible within a small group (e.g. a private messaging channel), knowing that it was submitted can reveal which group the user belongs to.

7.2.2 Image and Screenshot Uploads

For image-based content (memes, infographics, screenshots of tweets, etc.), the system uploads the file to the Debunking API for OCR and analysis.

Funded by the European Union

Anonymity-related concerns include:

- **Embedded identifiers in screenshots**
  Screenshots may include:
  - Usernames, profile pictures, or parts of the device UI.
  - Notification bars with personal messages, battery level, time and location hints. These elements can reveal the user's identity or context if stored or visible to operators.
- **Metadata (EXIF) in photos**
  Pictures taken by the camera may include EXIF metadata such as GPS coordinates, device model, and serial numbers. Unless explicitly stripped before upload, this conflicts with user anonymity: the backend could infer precise location and device identity.
- **Faces and other biometric signals**
  Posters, rallies or physical events may include the faces of the user or bystanders; if these images are stored, they become biometric data that can be used for re-identification.

7.2.3 Video and Audio Files

The system allows users to upload short video or audio clips and have them checked for disinformation.

Security and anonymity concerns:

- **Voice as a biometric identifier**
  Voice recordings are not just content; they are a biometric signature. If AI4Debunk stores raw audio, then, in principle, that data could be used to recognise the same speaker across different submissions or even across other systems.
- **Background and environmental sound**
  Audio can capture names, addresses, or other identifiable context (e.g. someone reading a phone number). It may also capture the voices of people who did not consent to analysis.
- **Visual identifiers in video**
  Videos may reveal locations, faces, car plates, and other highly identifying details, hence anonymisation is harder than for pure text.

The documentation does state that unsupported or too-long files trigger user-facing guidance rather than silent failure, and that personal history logs are created for checks.

From an anonymity standpoint, however, this history becomes a central risk point: it compiles a time-ordered view of the user's informational environment.

7.2.4 Live Audio Recording ("Record your Sound")

The "Record your Sound" feature is designed to capture broadcasts or speeches heard in real time. The specification already includes privacy-friendly behaviour: dismissed or too-short recordings should be discarded and not uploaded to the server.

Remaining concerns:

- **Trust boundary between app and backend**
  The anonymity properties depend on the app actually enforcing that short/dismissed recordings never leave the device. Any deviation (e.g. misconfiguration, logging at lower layers, or debug modes that upload everything) undermines this promise.
- **Contextual identification**
  Even if a recording is of a public broadcast, the fact that a specific user submitted this particular fragment at this time can reveal which channel they watched, at what hour, and therefore aspects of their media consumption habits.

7.3 Scanning Physical Content and Location Inference

"Scan with AR" allows users to photograph flyers, posters, or scan QR codes, then send the extracted content for analysis.

This introduces several anonymity issues:

- **Implicit location data**
  Physical disinformation campaigns are often geographically targeted. Knowing which poster a user scanned and when can reveal their likely city, neighbourhood, or even exact place (e.g. a specific bus stop or campus noticeboard). Combined with account data, this becomes a strong quasi-identifier.
- **QR codes with tracking parameters**
  QR codes sometimes embed unique tracking identifiers. If AI4Debunk sends these unmodified URLs to backends or external services, the original campaign organisers might learn that a person scanned that code from a specific IP or time, possibly deanonymising users.
- **Overlays and AR visuals**
  While AR overlays are mainly a UI feature, they may tempt designers to store both the original image and processed overlays for debugging or analytics. This multiplies the number of stored data artifacts that could reveal the user's location or the presence of other people in the scene.

Funded by the European Union

7.4 Logging, Global Statistics and Analytics

The home screen displays a global metric such as "total number of files identified as misinformation" fetched from an analytics backend, and the profile section includes user-level statistics and detailed history.

From an anonymity standpoint, the key concerns are:

1. **Backend logs vs. anonymised metrics**
   Even if the published metric is aggregated, the underlying analytics infrastructure typically stores events: which user analysed which content when, from which device/IP. Improper anonymisation or retention can make re-identification possible:
   - Rare events (e.g. checking a very unusual URL) may act as fingerprints.
   - Combining analytics with other logs (authentication, error monitoring) can rebuild full user timelines.
2. **User history as a map of interests**
   The profile and history expose past checks, file types, and risk scores to the user. If an attacker gains access to the backend or to a user's unlocked device, they obtain a rich profile of:
   - Political, health, or social topics the user is worried about.
   - Offline spaces they move through (from posters and flyers).
   - Media channels they follow (from live recordings).

   Even without explicit names, this behavioural pattern can be identifying.
3. **Linkability across sessions and devices**
   Because AI4Debunk synchronizes data across devices, a user profile effectively aggregates activity wherever they use the app. While functionally beneficial, this increases the impact of a single account compromise: an attacker can see not just one device's checks but the full merged history.

7.5 Reporting to Experts and External Ecosystems

The system allows users to "Report" suspicious content to a backend such as Disinfopedia, including the content and minimal metadata like language, platform type and timestamp, plus an optional user note.

Anonymity-related concerns:

Funded by the European Union

1. **Exposure of reporter identity to external parties**
   Even if AI4Debunk only sends minimal metadata, the receiving system (Disinfopedia or similar) may see:
   - An internal user identifier.
   - IP addresses or network identifiers.
   - Patterns of repeated reporting by the same person.
2. **Content of the optional description**
   The user's free-text note ("why they think it is suspicious, where they saw it, who it targets") may inadvertently include personal data about themselves or others. If these descriptions are stored or shared with third-party partners, they can be used to infer who the reporter is.

7.6 Device-Level Security and Local Anonymity

Although most threats discussed so far involve backends and networks, device-level security also affects user anonymity:

1. **Local storage of tokens and cached data**
   Session tokens, cached results, offline consent records and possibly parts of the history may be stored on the device to improve UX. If:
   - The device is shared.
   - The OS is rooted/jailbroken.
   - Malware is present.

   Then an attacker can read this data and reconstruct what the user has been checking, even if the server-side logs are well protected.
2. **Screenshots and screen recording**
   Result screens and history views may be captured via OS-level screenshots or screen recording tools. From the app's perspective, this is outside its control, but in threat modelling terms it means that sensitive information can easily leak into other apps, backups or cloud galleries.
3. **Notification content**
   If future versions of AI4Debunk use push notifications (e.g. "Your report has been reviewed"), these messages may appear on the lock screen. Depending on their wording, they may reveal to anyone nearby what topics the user is interested in.

7.7 Consent, Transparency and User Expectations

The onboarding flow already includes screens explaining what data is processed, for what purposes, and involving which partners, and records consent with timestamp and policy version.

Security and anonymity concerns here are more indirect, but still important:

1. **Mismatch between perceived anonymity and actual data flows**
   Users may assume that because they are "just checking content" and not posting publicly, their activity is anonymous. In reality:
   - Their checks are tied to a login.
   - Reports may be forwarded to external systems.
   - Audio, images and text can have rich personal context.

   If the UX does not explicitly counter this misconception, users might engage with high-risk content under a false sense of anonymity.
2. **Complexity of explaining multi-party data flows**
   AI4Debunk interacts with an authentication backend, a debunking API, analytics services, and possibly Disinfopedia or other expert systems. Explaining all of these in simple language is challenging, yet necessary so that users can make an informed judgment about anonymity versus functionality trade-offs.
3. **Policy evolution over time**
   As models, partners, or regulations change, data usage patterns may evolve. If policy updates are not clearly surfaced, earlier consent may no longer reflect the user's expectations about anonymity and security.

7.8 Data Minimisation, Retention and Re-identification Risks

Many anonymity concerns ultimately reduce to how long data is kept and how granular it is:

1. **Retention of raw content vs. derived features**
   If the system keeps raw submissions (full text, full images, raw audio), then even with pseudonymous IDs, re-identification risks are high. Derived features (e.g. claim vectors, risk scores, narrative categories) are less directly identifying, but can still act as fingerprints in edge cases.
2. **Linkability over time**
   Long-term retention of user-level histories and analytics makes it possible to:
   - Reconstruct behavioural trajectories.
   - Perform cross-event correlation ("this user checks only content from platform X about topic Y"), which can reveal identity indirectly.
3. **Combination with external datasets**
   Attackers or even legitimate investigators might combine AI4Debunk logs with external data (e.g. ISP logs, social media traces) to deanonymise users. For instance, if a specific

controversial video is known to be accessible only in a certain group, and the system logs a check of that video at a specific time, this may reduce the anonymity set drastically.

7.9 SUMMARY

AI4Debunk is, by design, a powerful verification tool that touches many aspects of a user's informational life: what they read, which flyers they encounter in the street, which broadcasts they listen to, and what they decide to report to experts.

This breadth creates a correspondingly rich attack surface for user anonymity:

- At the identity layer, persistent accounts, social logins and synchronized histories make the system fundamentally pseudonymous rather than anonymous.
- At the content layer, submissions can embed direct personal data (faces, voices, screenshots) and indirect contextual clues (locations, closed-group material).
- At the logging and analytics layer, detailed histories and global metrics risk turning individual behaviour into a long-lived profile, even if only anonymised aggregates are formally exposed.
- At the ecosystem layer, reports forwarded to third-party expert systems may expose reporter identity or context in subtle ways.
- At the device layer, cached data, persistent sessions and notifications can leak sensitive information on shared or compromised devices.

Addressing these concerns does not necessarily mean abandoning the rich feature set of AI4Debunk, but it does require explicit design choices: clear communication that the service is primarily pseudonymous; strong separation between identity and content where possible; strict data minimisation and retention limits; robust encryption and access controls; and careful handling of audio, image and location-bearing data.

Only by treating user anonymity as a first-class requirement, not a secondary afterthought, can AI4Debunk maintain user trust while operating in a domain (misinformation and debunking) that is often politically charged and personally sensitive.

Funded by the European Union

8. MOBILE APPLICATION TECHNOLOGY STACK AND FRAMEWORK CHOICE

8.1 RATIONALE FOR SELECTING FLUTTER

Given the breadth of AI4Debunk's functionalities, ranging from user registration and consent management to multi-modal content verification (text, images, video, audio, QR codes) and secure handling of personal histories, the mobile app requires a robust, high-performance and maintainable technology stack that can serve both major mobile ecosystems consistently.

To meet these requirements, the project adopts Flutter as the primary framework for developing the AI4Debunk mobile application. Flutter is a cross-platform UI toolkit maintained by Google, enabling developers to build native-compiled applications for both Android and iOS from a single codebase. This choice directly supports the project's goal of wide user reach, ensuring that users on both platforms have access to the same features, user interface, and security guarantees without maintaining two separate native implementations.

From a performance perspective, Flutter renders its UI using a high-performance graphics engine (Skia) and compiles to native ARM code. This allows the app to deliver smooth animations and responsive interactions, which is particularly important for camera-based flows ("Scan with AR"), live audio recording ("Record your Sound"), and rich result screens that visualise risk indicators and explanations. At the same time, Flutter's declarative, widget-based architecture facilitates the design of a coherent and accessible interface, aligning with the requirement that non-expert users should be able to understand results at a glance.

8.2 FIT WITH AI4DEBUNK FUNCTIONAL REQUIREMENTS

Flutter provides a mature ecosystem of plugins and packages that cover the main device capabilities required by AI4Debunk:

- Camera and gallery access for uploading images, screenshots, and videos, as well as capturing photos of physical content and QR codes.
- Microphone access and audio APIs to implement the "Record your Sound" feature with clear, real-time feedback and reliable recording controls.
- Secure local storage and integration with platform-level keychains/keystores for persisting authentication tokens, consent flags, and minimal cached data in line with the security and privacy requirements.

Funded by the European Union

- Network and HTTP clients for communicating with the authentication backend, the Debunking API, analytics services, and external ecosystems such as expert review platforms.
- Support for social login flows (e.g. Google Sign-In) through well-maintained community and official plugins, which fits the documented authentication scenarios.

Because these capabilities are available through a unified abstraction layer, the implementation of each use case (e.g. uploading a file, scanning a poster, recording audio, viewing history) can be developed once and reused across iOS and Android. This reduces the risk of platform-specific inconsistencies, such as a feature behaving differently or being less secure on one platform compared to the other.

8.3 Maintainability, Evolution and Testing

Choosing Flutter also supports the long-term maintainability and evolution of the AI4Debunk app:

- A single shared codebase reduces development and maintenance effort, allowing the project team to react more quickly to evolving requirements, policy updates, or new verification flows (for example, new content types or additional risk explanations).
- Flutter's hot reload and rapid iteration capabilities enable faster prototyping and refinement of user flows, which is valuable when adjusting UX details for onboarding, consent, and explanation screens based on user feedback and usability studies.
- The framework's strong support for modular architecture and automated testing (unit, widget and integration tests) facilitates systematic validation of security-sensitive behaviour, such as proper discarding of too-short recordings, correct handling of permissions, and consistent anonymisation of content before network transmission.
- The active community and backing by a major vendor provide a stable and future-proof ecosystem, making it more likely that critical dependencies (e.g. secure storage, camera, OAuth libraries) will be kept up to date with platform and OS changes over the lifetime of the project.

In summary, Flutter is selected as the core mobile framework because it aligns with the project's cross-platform reach, performance, usability, and privacy/security requirements. It allows AI4Debunk to deliver a consistent, high-quality experience on both Android and iOS, while keeping the implementation maintainable and adaptable as the underlying debunking services and regulatory landscape evolve.

Funded by the European Union

9. ALIGNMENT WITH SSH KEY RECOMMENDATIONS FOR TOOL DEVELOPERS

This section maps each SSH recommendation from the WP10 item list to the AI4Debunk mobile app definition (D10.3), stating whether it applies to the app and why. For applicable recommendations, we summarise how the app specification addresses them.

9.1 User-Centric Design

9.1.1 Design intuitive interfaces that make navigation and interpretation of data simple for both technical and non-technical users (including young and elderly people). Include tooltips, visual aids (graphs, videos).: Applicable

We designed a simple, intuitive UI with clear primary actions and straightforward navigation so both technical and non-technical users can use it easily. To make results easy to interpret, we rely on visual aids like a prominent risk indicator (speedometer-style gauge) plus short, readable explanations, and we support first-time users with a brief onboarding/tutorial flow.

9.1.2 Enable customization: Users should tailor analysis depth, alerts, and display preferences. Provide "basic" and "advanced" modes to cater to diverse audiences.: Applicable

We support customization by letting users tailor both analysis depth and presentation: a "quick" debunking flow provides a simple result for basic use, while users can expand into a more detailed view ("Read More") for advanced interpretation and supporting evidence, effectively covering basic vs. advanced usage needs within the same workflow.

9.1.3 Provide comprehensive user support through integrated FAQs, contextual help, and multimedia learning aids (tutorials, videos, interactive walkthroughs): Applicable

We provide user support mainly through an in-app onboarding/tutorial walkthrough (short swipeable sequence) and contextual guidance inside the flows, e.g., screens that explain what to do (like the recording screen), clear permission justifications, and friendly messages that guide the user when something is unsupported (e.g., long/unsupported files). Given the goal that the software should be self-explanatory, this approach reduces the need for a heavy standalone FAQ section while still covering help and learning needs within the UI.

9.2 Gender Equality and Inclusivity

9.2.1 Ensure gender-neutral design and eliminate bias in algorithms and datasets. Conduct regular gender audits to test for bias in detection outcomes.: Applicable

We ensured a gender-neutral UX by using inclusive, non-gendered wording and avoiding assumptions in user-facing text, with the consortium explicitly reviewing usability and wording as part of the app definition process. For algorithm/dataset bias and gender audits, the mobile front-end is not where detection happens: the app sends verification requests to the Debunking API and only presents the returned score/verdict/explanations, so outcome-bias auditing is handled at the backend/model layer. At project level, we also align with the consortium's commitment to mainstream gender and intersectionality through ongoing monitoring and periodic evaluation practices.

9.2.2 Also make sure that all vulnerable groups can access the tool and interface (people living in remote areas or in the countryside, elderly people, people from different minorities eg. Russian speaking diaspora but not only…): Applicable

We designed AI4Debunk to be broadly accessible by delivering the same experience on both Android and iOS (wide reach via a single cross-platform implementation), supporting checks that start from offline/physical material (e.g., scanning flyers/posters), and handling remote/poor connectivity gracefully (clear "no network / API unreachable" messaging with retry, plus explicit testing of offline/poor connectivity cases). We also plan dedicated usability/accessibility validation with vulnerable groups (including seniors, adolescents 13+, and low educational background users), and the reporting flow carries language metadata to support diverse linguistic communities.

9.2.3 Collaborate with gender experts and advocacy groups during design and testing to ensure the tool serves all genders equitably: Not Applicable

Not applicable as a software/UI deliverable requirement, since the app is a technical client for the Debunking API and does not itself implement gender-specific decision logic; any equity validation belongs primarily to the model/dataset governance and evaluation activities rather than front-end design.

9.2.4 Support inclusivity: Offer multi-language interfaces, culturally neutral symbols, and accessibility features (screen readers, keyboard navigation, voice commands): Applicable

We support inclusivity by keeping the interface clear and culturally neutral (action-oriented navigation with four large, clearly labeled entry points) and by using accessible language in onboarding/consent instead of dense jargon. To enable multi-language support, the reporting flow already captures language metadata (so content can be handled appropriately and the UI can be localized), and accessibility will be validated in focus groups explicitly targeting vulnerable users (e.g., seniors, low educational background). Additionally, using Flutter helps us deliver the same experience on both Android/iOS and leverage OS-level accessibility services (e.g., screen readers/voice control) during implementation.

Funded by the European Union

9.2.5 Comply with global accessibility standards and optimize tools for low-tech or low-bandwidth environments: Applicable

We optimized AI4Debunk for accessibility and low-bandwidth/low-tech conditions by keeping the interaction model simple and usable for all age groups (clear entry points, minimal steps, understandable results), adding graceful degradation when connectivity is limited (e.g., "no network / API unreachable" messaging with retry instead of failure), and explicitly testing offline/poor connectivity cases as part of the validation plan. We also support "in-the-moment" use in low-resource contexts via scanning offline/physical material (posters/flyers/QR), and we plan dedicated usability/accessibility evaluation with vulnerable groups using SUS-based focus groups.

9.2.6 Use region-specific data and partner with local experts to make tools sensitive to cultural and linguistic disinformation trends: Not Applicable

This aspect is outside the scope of the front-end definition in D10.3, but the project can still address it operationally through the wider ecosystem: the app supports "Report to experts" workflows where suspicious items are forwarded to expert backends (e.g., Disinfopedia) together with minimal metadata such as language, platform type and timestamp, so human reviewers can spot emerging local narratives and feed improvements back into the system. In practice, this would be done by the expert backend operators / human fact-checkers and relevant consortium partners, with iterative refinements informed by consortium/Steering Committee validation and structured testing feedback loops.

9.3 Integration with Social Media

9.3.1 Implement API-based real-time analysis for major platforms (Facebook, X/Twitter, Bluesky, TikTok, YouTube, Instagram): Not Applicable

Not applicable for the front-end definition, because the mobile app does not directly integrate with social-media platform APIs for real-time monitoring; instead it submits user-provided content to the AI4Debunk Debunking API and displays the returned results, while any platform-specific, real-time integrations would belong to separate backend/services outside this deliverable's scope.

9.3.2 Ensure cross-platform tracking to trace how disinformation spreads across ecosystems: Not Applicable

Not applicable for the front-end definition, because cross-platform tracking and spread analysis is an ecosystem-level backend function (data collection, correlation, analytics) rather than something implemented in the mobile UI. The app's role is to submit items for verification to the Debunking API and present results; tracing propagation across platforms would be handled by dedicated backend/analytics components outside this deliverable's scope.

9.3.3 Access metadata (timestamps, geolocation, engagement) to understand virality and amplification dynamics while maintaining GDPR-compliant privacy protections: Not Applicable

Not applicable in the sense of actively harvesting virality metadata (geolocation, engagement signals, cross-platform amplification) from platforms: AI4Debunk follows data minimisation and only uses minimal metadata where needed (e.g., reports include language, platform type, timestamp), without collecting geolocation/engagement data. The privacy approach is GDPR-aligned by design (permissions only when required, explicit consent, and account deletion flows), and processing is intended to be transient with only anonymised operational logging.

9.3.4 Monitor emerging and niche platforms (e.g., Telegram, Gab, Parler) and adapt to new platform features like short-lived content or encrypted messaging: Not Applicable

Not applicable for the front-end definition, because continuous monitoring of niche/emerging platforms and adapting to platform-specific features (e.g., encrypted or ephemeral content) requires platform-level integrations and backend monitoring pipelines, which are outside the scope of the AI4Debunk mobile app. The app is designed as a client that submits user-provided content to the Debunking API and presents the outcome, rather than operating as a cross-platform monitoring tool.

9.4 Tackling Coordinated Inauthentic Behaviour (CIB)

9.4.1 Provide real-time alerts and dashboards that visualize spikes in engagement, bot activity, or suspicious hashtag use: Not Applicable

Not applicable because AI4Debunk is designed as a user-driven verification app (submit content → receive a debunking result from the Debunking API), not as an analytics platform that ingests network-wide signals to generate dashboards/alerts for engagement spikes, bots, or hashtags. This is also out of scope for the current workplan; if a separate backend service and hosting later provide such aggregated monitoring outputs, the app could potentially integrate them as an add-on, but it is not part of the current AI4Debunk topology.

9.4.2 Educate users through built-in guidance on recognizing coordinated behaviour (e.g., identical posts, synchronized activity): Not Applicable

Not applicable for the current AI4Debunk app definition, since the app focuses on content verification via the Debunking API rather than training users to detect coordination patterns in social behaviour. This kind of educational material is better suited to dedicated engagement/learning components (e.g., the comic-book or VR experiences) rather than the core verification UI.

Funded by the European Union

9.4.3 Integrate network visualization tools to help users identify clusters and influencers driving disinformation: Not Applicable

Not applicable because network visualizations (clusters/influencers) require graph analytics and data aggregation across users/platforms, which is outside AI4Debunk's topology and scope; the app is a client that submits items for verification to the Debunking API and presents the result, not a network-monitoring or influence-mapping tool.

9.4.4 Allow user-driven monitoring and flagging, enabling journalists and analysts to investigate specific narratives and provide corrective feedback: Applicable

We support user-driven flagging through the "Report" flow, where logged-in users can submit suspicious items to an expert backend (e.g., Disinfopedia) together with minimal metadata and an optional note, so journalists/analysts can review cases and feed corrections back into the monitoring ecosystem. The app also keeps a searchable, filterable history of past checks, which helps users (including journalists/researchers) monitor specific narratives over time and escalate the most relevant items for expert investigation.

9.5 Ethical and Transparent AI

9.5.1 Prioritize data privacy and user consent. Collect and store only necessary public data, ensuring transparency about use and limitations.: Applicable

We prioritised privacy-by-design by using an onboarding-first consent flow that explains what data is processed, for what purpose and with which partners, and records consent with timestamp + policy/app versioning for auditability and transparency. Data collection is minimised: verification requests are processed transiently, the app does not retain user personal data and keeps only anonymous operational logs under GDPR-aligned retention policies. We also request permissions only when needed and discard non-needed inputs (e.g., dismissed/too-short recordings), while the account deletion flow clearly explains what is erased and what anonymised analytics may remain. When sending reports externally, we include only minimal metadata (e.g., language/platform/timestamp) alongside the content.

9.5.2 Avoid algorithmic bias by maintaining diverse, balanced datasets and documenting model decision-making processes: Not Applicable

Not applicable for this deliverable, because D10.3 defines the mobile app as a client that submits user content for verification and displays the Debunking API output (Disinfoscore/classification/explanations), while dataset composition, model transparency, and training-time bias mitigation belong to the backend AI modules rather than the front-end scope. In line with the consortium discussions already noted (opaque pre-trained LLMs → unknown training data), bias cannot be fully eliminated upfront; instead, it is treated as a formal project risk and addressed via dedicated bias testing planned in Task 11.6.

Funded by the European Union

9.5.3 Provide transparency in flagging: Explain why content is flagged and offer mechanisms for users to dispute or verify flagged material: Applicable

We provide transparency in flagging by showing, on the results screen, not only the overall verdict/score but also the main signals that triggered suspicion, and we let users open a deeper "Read More" view to see the supporting explanations/evidence behind the assessment. The dispute/appeal mechanism is handled in the disinformation platform, while the app supports escalation via "Report" to expert backends (e.g., Disinfopedia) when users want human verification.

9.5.4 Establish ethical safeguards to prevent misuse of detection tools for censorship or political manipulation: Not Applicable

Not applicable as a front-end requirement in D10.3: AI4Debunk is specified as a user-facing verification tool (it helps citizens assess content) and is not designed to enforce removals, suppress speech, or perform political monitoring, so "anti-censorship" safeguards mainly sit in governance, deployment policy, and ethics/risk management rather than UI mechanics. That said, the app supports the intent by emphasizing transparency and informed consent (clear explanation of data flows/limits) and strict data minimisation/retention controls, which reduce the risk of the tool being repurposed for surveillance-like use.

9.6 Multilingual and Cultural Adaptation

9.6.1 Train NLP models per language to recognize regional slang, idioms, and context-sensitive disinformation patterns: Not Applicable

Not applicable for the front-end definition, because training/tuning NLP models per language (slang/idioms/context) is a backend/model responsibility: the app's role is to submit user-provided content to the Debunking API and present the returned score/verdict/explanations. To still support future language-specific improvements, the reporting workflow forwards suspicious items with language metadata (plus platform type and timestamp) to the expert backend, so the platform/model owners can adapt models based on real user feedback.

9.6.2 Enable cross-language analysis to track narrative migration across languages and regions: Not Applicable

Not applicable because cross-language narrative migration analysis is a backend monitoring/analytics capability, while the AI4Debunk app is specified as a client that submits content to the Debunking API and displays the returned Disinfoscore/classification/explanations. Although the reporting flow can forward suspicious items with minimal metadata (including language, platform type, timestamp) to the expert backend,true cross-language tracking would require dedicated translation/correlation pipelines; given the constraints you noted, this would

Funded by the European Union

realistically rely on existing/open-source translation models plus benchmarking, rather than retraining large LLMs or depending on proprietary ones.

9.6.3 Use culturally aware machine translation for languages without native models, ensuring meaning and intent are preserved: Applicable

We will support languages that don't have a native model by using a translation-based fallback on the backend (machine-translate to a supported analysis language and run the same Debunking API pipeline), while clearly treating this path as experimental/pending until it is validated with benchmarks and quality checks. At app level, we already preserve the language context by including language (along with other minimal metadata) in the reporting package sent to expert backends, so translation/routing can improve iteratively based on real cases.

9.7 Stakeholder and Expert Engagement

9.7.1 Establish regular consultation mechanisms (workshops, focus groups, advisory boards) with journalists, researchers, fact-checkers, and policymakers: Applicable

We have already set up structured consultation mechanisms through two internal focus-group rounds (one with WP leaders and one at consortium level including the Steering Committee) to validate the app scope against cross-WP constraints and refine requirements based on consolidated feedback. In parallel (via other WPs), the project runs broader engagement through workshops/webinars/events and surveys/interviews, and additional focus groups are planned once the app is fully developed to assess usability/accessibility in realistic conditions.

9.7.2 Run diverse beta testing programs involving participants from different cultural, linguistic, and professional backgrounds: Applicable

We address this through a staged "beta" approach in combination with other WPs: first, the app definition was validated via two internal focus-group consultations (WP leaders, then consortium members/Steering Committee) to refine usability and feasibility across partners, and the plan then moves to a dedicated quality environment for structured testing by focus groups and other non-developer testers. In a later phase, once the app is fully developed, focus-group testing explicitly targets diverse citizen profiles with vulnerabilities (seniors, adolescents 13+, low educational background) and uses a standardized usability instrument (SUS) to make feedback comparable across groups.

9.7.3 Continuously integrate feedback from stakeholders into iterative tool updates: Applicable

We continuously integrate stakeholder feedback through an Agile, iterative process where requirements are refined in short feedback cycles with project stakeholders and then validated via consortium focus groups (including the Steering Committee), with the consolidated feedback turned into actionable refinements across the deliverable. In implementation/testing, this

Funded by the European Union

continues via iterative testing across environments (including a quality environment for focus groups) so issues and usability findings are captured and fed into the next updates.

9.8 Continuous Improvement

9.8.1 Update algorithms regularly to reflect new disinformation tactics (deepfakes, AI-generated content, bot evolution): Applicable

We address this primarily at the Debunking API layer, since the app is a client that sends user content for analysis and displays the returned Disinfoscore/verdict/explanations (so new tactics are handled by updating the API's underlying models/knowledge sources without redesigning the mobile UI). When new verification flows or richer explanations are introduced, the Flutter codebase is explicitly designed for maintainability and fast evolution, so the app can be updated quickly to match the API's capabilities.

9.8.2 Expand and diversify datasets using regional and topical sources, verified fact-checking data, and emerging content types: Applicable

We support dataset expansion mainly through the Debunking API: analyses already leverage knowledge graphs and previously debunked cases, and the app's Report-to-experts flow (e.g., Disinfopedia) is designed to funnel real user cases (with minimal metadata like language/platform/timestamp) back to experts so new narratives and examples can be reviewed and used to refine models/datasets over time. Since AI4Debunk already covers emerging content types (text/URL, images, video/audio, live audio, offline/physical scans), the same approach can be extended later (e.g., for the browser extension) by prioritising regional/topical sources and verified fact-check corpora at the backend.

9.8.3 Maintain feedback loops: allow users to report false positives or missed detections and receive updates on actions taken: Applicable

We maintain a feedback loop by letting logged-in users report a result directly from the results screen, sending the content plus minimal metadata (language/platform/timestamp) and an optional note, so users can flag suspected false positives or missed detections for human review; the expert backend then assigns an ID, queues the case for review, and the app confirms submission. Updates on actions taken are handled at the platform/expert-backend layer (e.g., a future "your report has been reviewed" notification/status), which the app can surface once available.

9.8.4 Collaborate with experts in AI, disinformation, and digital ethics to ensure tools evolve responsibly and effectively: Applicable

We ensure responsible evolution by continuously involving consortium expertise across WPs: requirements were refined through two internal consultations, one with work-package leaders

to assess feasibility, risks, and privacy-by-design considerations, and a second with consortium partners and the Steering Committee to validate usability, transparency, and informed consent for citizens. This runs alongside an Agile feedback cycle where stakeholder input is regularly consolidated into actionable refinements.

9.9 Overall Recommendation

9.9.1 Developers should approach disinformation detection as a living process, emphasizing adaptability, transparency, inclusivity, and collaboration. By integrating ethical AI design, user engagement, and continuous learning, tools can remain robust against evolving disinformation landscapes and build user trust globally.: Applicable

Applicable at project level (even if not specific to WP10): WP10 supports this “living process” principle by defining a user-facing tool that is transparent and inclusive by design (clear results/explanations, consent-first onboarding, accessibility-focused testing), while enabling continuous learning through reporting/feedback loops that route real cases to expert backends so the overall system can adapt as disinformation tactics evolve.

Review Sheet of Deliverable/ Milestone Report

D10.3 Report on the Definition of the App

Editor(s):	Dr. Despina Elisabeth Filippidou (DOTSOFT), Karanasios George (DOTSOFT), Stavros Katsaridis (DOTSOFT), Katsakioris Dimitris (DOTSOFT), Nikas Marios (DOTSOFT), Simeonidou Anastasia (DOTSOFT), Nikopoulos George (DOTSOFT), Maragkos Christodoulos (DOTSOFT)
Responsible Partner:	DOTSOFT
Status-Version:	v1
Date:	05/12/2025
Distribution level (CO, PU):	Public
Reviewer (Name/Organization)	Georgi Gotev, Kalina Angelova (EUalive)
Review date	24/02/2026

Disclaimer: This assessment reflects only the author's views and the European Commission is not responsible for any use that may be made of the information contained therein"

Mark with X the corresponding column:

Y= yes	**N= no**	**N = not applicable**

ELEMENT TO REVIEW	Y	N	NA	COMMENTS
FORMAT: Does the document … ?				
…include editors, deliverable name, version number, dissemination level, date, and status?	Y			
…contain a license (in case of public deliverables)?			NA	
…include the names of contributors and reviewers?	Y			
….has a version table consistent with the document's revision?	Y			

… contain an updated table of contents?	Y			
… contain a list of figures consistent with the document's content?	Y			
… contain a list of tables consistent with the document's content?			NA	
… contain a list of terms and abbreviations?	Y			
… contain an Executive Summary?	Y			
… contain a Conclusions section?		N		
… contain a List of References (Bibliography) in the adequate format, if relevant?			NA	
… use the fonts and sections defined in the official template?	Y			
… use correct spelling and grammar?	Y			
… conform to length guidelines (50 pages maximum (plus Executive Summary and annexes)	Y			
… conform to guidelines regarding Annexes (inclusion of complementary information)			NA	
… present consistency along the whole document in terms of English quality/style? (to avoid accidental usage of copy&paste text)	Y			
About the content…				
ELEMENT TO REVIEW	**Y**	**N**	**NA**	**COMMENTS**
Is the overall style of the deliverable correctly organized and presented in a logical order?	Y			
Is the Executive Summary self-contained, following the guidelines and does it include the main conclusions of the document?	Y			
Is the body of the deliverable (technique, methodology results, discussion) well enough explained?	Y			
Are the contents of the document treated with the required depth?	Y			
Does the document need additional sections to be considered complete?		N		
Are there any sections in the document that should be removed?		N		
Are all references in the document included in the references list?			NA	
Have you noticed any text in the document not well referenced? (copy and paste of text/picture without including the reference in the reference list)		N		
SOCIAL and TECHNICAL RESEARCH WPs (WP4, 5, 12, 13, 14)				
ELEMENT TO REVIEW	**Y**	**N**	**NA**	**COMMENTS**

Funded by the European Union

Is the deliverable sufficiently innovative?	Y			
Does the document present technical soundness and its methods are correctly explained?	Y			
What do you think is the strongest aspect of the deliverable?				The deliverable clearly translates social goals into concrete app functionality.
What do you think is the weakest aspect of the deliverable?				
Please perform a brief evaluation and/or validation of the results, if applicable.			NA	
AI AND TECNOLOGICAL WPS (WP6 – WP11)				
ELEMENT TO REVIEW	**Y**	**N**	**NA**	**COMMENTS**
Does the document present technical soundness and the methods are correctly explained?	Y			
What do you think is the strongest aspect of the deliverable?				The deliverable has a comprehensive, structured, and implementation-ready use case architecture.
What do you think is the weakest aspect of the deliverable?				Limited information on how the disinfoscore is calculated.
Please perform a brief evaluation and/or validation of the results, if applicable.			NA	
DISSEMINATION AND EXPLOITATION WPs (WP15 – WP17)				
ELEMENT TO REVIEW	**Y**	**N**	**NA**	**COMMENTS**
Does the document present a consistent outreach and exploitation strategy?	Y			
Are the methods and means correctly explained?	Y			
What do you think is the strongest aspect of the deliverable?				The app is not just as a technical tool, but it is a part of a broader ecosystem involving users, experts, and collective reporting
What do you think is the weakest aspect of the deliverable?				
Please perform a brief evaluation and/or validation of the results, if applicable.			NA	
DISSEMINATION AND EXPLOITATION WPs (WP18)				
ELEMENT TO REVIEW	**Y**	**N**	**NA**	**COMMENTS**
Does the document present the main ethical aspects regarding the use of methods and human involvement?	Y			
What do you think is the strongest aspect of the deliverable?				

What do you think is the weakest aspect of the deliverable?				
Please perform a brief evaluation and/or validation of the results, if applicable.			NA	

SUGGESTED IMPROVEMENTS

PAGE	SECTION	SUGGESTED IMPROVEMENT

CONCLUSION

Mark with X the corresponding line.

X	Document accepted, no changes required.
	Document accepted, changes required.
	Document not accepted, it must be reviewed after changes are implemented.

Please rank this document globally on a scale of 1-5 (1 = poor, 5= excellent) – using a half point scale. Mark with X the corresponding grade.

Document grade	1	1.5	2	2.5	3	3.5	4	4.5	5
									X