

TEAM-1- Software Requirements Specification (SRS)

Team members: Arın Akbaş, Dinçer Efe Bolut.

1. Introduction

1.1 Purpose

The purpose of this document is to explain what the One Minute Digital Diary should do and how it should work. It outlines the features, user interactions, constraints, and quality attributes of the system to guide development, testing, and project management.

1.2 Document Conventions

“US-XX” → User Story ID

“FR-XX” → Functional Requirement ID

“NFR-XX” → Non-Functional Requirement ID

1.3 Intended Audience

This document is intended for project managers, developers, and testers.

1.4 Project Scope

The One Minute Digital Diary app allows users to record short daily videos (up to 1 minute for free users, 5 minutes for premium users), storing them privately. When a user records their daily video their streak will continue. Streak resets if the user skips a day. Users can view their past entries organized by date. A feature allows users to optionally make their daily video public for 24 hours, where other users can view and interact with them. Users manage their accounts via email/password authentication and can update profile details including a profile picture. Premium features are unlocked via payment processing handled by Apple StoreKit for iOS users and Google Play Billing for Android users. The system includes a daily recording limit and reminder notifications.

There won't be advanced video editing features beyond basic recording, direct messaging between users, real-time video streaming. Integration with external social media platforms for sharing.

1.5 References

Team-1 Software Development Plan (SDP)

2. Overall Description

2.1 Product Perspective

One Minute Digital Diary is a mobile application for iOS and Android, using Firebase for backend services (Authentication, Firestore Database, Storage, Cloud Functions for notifications). It integrates with the native in-app purchase systems for premium subscriptions. It relies on the device's camera and OS for permissions and notifications.

2.2 Product Functions

- User Authentication (Sign Up, Login, Password Reset - Note: Reset not implemented yet, Sign Out)
- Profile Management (View/Update Username, Profile Picture)
- Daily Video Recording (Flipping Camera, Pinch-to-Zoom, Timed Recording Limit)
- Streaks (After user recorded their first video their streak begins.)
- Gallery View (Newest to oldest list of user's videos)
- Public Feed View that displays public videos for 24 hours from being published. (Public videos are automatically removed from public feed after 24 hours but remain in the user's private archive unless deleted.)
- Video Interaction (Liking public videos)
- User Profile Viewing (View other users' public videos)
- Settings Management (Premium Subscription, Change Email, Change Password, Delete Account)
- Daily Recording Limit Enforcement
- Reminder Notifications

2.3 User Classes and Characteristics

- **Registered User (Free):** Can perform all core functions but limited to 1-minute recordings.
- **Registered User (Premium):** Can perform all core functions with a 5-minute recording limit. (Potentially added features)

2.4 Operating Environment

- **Mobile Devices:** iOS and Android smartphones capable of running Flutter applications and have cameras.
- **Network:** Requires an internet connection for Firebase interactions and payment processing.

2.5 Design and Implementation Constraints

- Must be built using Flutter.
- Must use Firebase for all backend services (Auth, Firestore, Storage, Functions).
- Must utilize the device's native camera capabilities via the camera package.
- Payment processing for premium subscriptions must be handled via integration with the Apple StoreKit for iOS users and Google Play Billing for Android users.
- UI should be responsive and user-friendly.

2.6 Assumptions and Dependencies

- Users have iOS or Android devices with functional cameras and microphones.
- Users accept permissions (camera, microphone, notifications).
- Users have a valid payment method if they wish to upgrade.
- Users have a stable internet connection for most features, including payment.
- Firebase services remain available and functional.
- The payment gateway services remain available and functional.
- Platform APIs (camera, notifications) remain stable.

3. Functional Requirements

3.1 User Stories

US-01: User Sign Up

- As a new user
- I want to create an account using my email, a chosen username, and a password
- So that I can start using the One Minute Digital Diary.
- Priority: Must Have
- **Acceptance Criteria:**
 - Given I am on the login screen and choose 'Sign Up'
 - When I enter a unique username, a valid email, and a password and confirm the password

- Then my account is created, my user data is saved, and I am logged in and taken to the home screen.
- Given I enter an email that already exists
- When I try to sign up
- Then I see an error message "Email already in use".
- Given I enter passwords that don't match
- When I try to sign up
- Then I see an error message "Passwords do not match".
- **US-02: User Login**
 - As a registered user
 - I want to log in using my email and password
 - So that I can access my private diary and other features.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I am on the login screen
 - When I enter my correct email and password and tap 'Login'
 - Then I am logged in and taken to the home screen.
 - Given I enter an incorrect email or password
 - When I tap 'Login'
 - Then I see an error message "Invalid credentials".
- **US-03: User Sign Out**
 - As a logged-in user
 - I want to sign out from the settings page
 - So that my account is secured.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I am logged in and on the Settings screen
 - When I tap the 'Sign Out' button
 - Then I am logged out and returned to the Login screen.
- **US-04: Record Daily Video**
 - As a registered user
 - I want to record a video using my phone's camera
 - So that I can capture my daily entry.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I haven't recorded today and tap the 'Record' tab
 - When the camera screen loads
 - Then I see a preview from the default camera.
 - Given I am on the camera screen

- When I tap the record button
 - Then the video recording starts and a timer appears.
- **US-05: Switch Camera Before Recording**
 - As a registered user
 - I want to switch between the front and back cameras before recording
 - So that I can choose the appropriate view.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I am on the camera screen and not recording
 - When I tap the 'Flip Camera' icon
 - Then the camera preview switches to the other available camera.
- **US-06: Pinch to Zoom**
 - As a registered user
 - I want to pinch the camera preview to zoom in or out
 - So that I can frame my recording better.
 - **Priority:** Should Have
 - **Acceptance Criteria:**
 - Given I am on the camera screen (recording or not)
 - When I perform a pinch gesture on the preview
 - Then the camera preview zooms in or out within the device's supported limits.
- **US-07: Timed Recording Limit**
 - As a user
 - I want the recording to automatically stop after my allowed time limit (1 min free / 5 min premium)
 - So that I don't accidentally record too long.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I am a free user and start recording
 - When the timer reaches 1 minute (01:00)
 - Then the recording stops automatically.
 - Given I am a premium user and start recording
 - When the timer reaches 5 minutes (05:00)
 - Then the recording stops automatically.
- **US-08: Save Recorded Video**
 - As a user
 - I want to save my recorded video with a title and choose if it's public

- So that it's added to my diary.
- **Priority:** Must Have
- **Acceptance Criteria:**
 - Given I have stopped recording a video
 - When I am taken to the 'Save Video' screen
 - Then I see a preview of the video, a field for the title, and a toggle for 'Make Public'.
 - Given I enter a title and tap 'Save & Upload'
 - Then the video (compressed) is uploaded to storage, metadata (title, URL, user info, public status, timestamp) is saved to Firestore, and I am returned to the home screen.
- **US-09: Daily Recording Limit Enforcement**
 - As a user
 - I want to be prevented from recording more than one video per day
 - So that the app's core concept is maintained.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I have already recorded and saved a video today
 - When I navigate to the 'Record' tab
 - Then I see a message "You have already recorded today. See you tomorrow!" instead of the camera preview.
- **US-10: View Private Diary**
 - As a logged-in user
 - I want to view my past video entries on the 'My Diary' tab
 - So that I can review my memories.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I am on the 'My Diary' tab
 - When the screen loads
 - Then I see a list of my videos, ordered chronologically (newest first), displaying the date and title.
 - Given I tap on a video preview
 - When the video loads
 - Then it starts playing within the list item and respects the aspect ratio.
- **US-11: View Public Feed**
 - As a logged-in user
 - I want to view videos marked as public by other users within the last 24 hours on the 'Public' tab

- So that I can see what others are sharing.
- **Priority:** Must Have
- **Acceptance Criteria:**
 - Given I am on the 'Public' tab
 - When the screen loads
 - Then I see a list of public videos created within the last 24 hours, ordered newest first.
 - And each video displays the creator's username, profile picture, title, date, and like button.
- **US-12: View User Profile**
 - As a logged-in user
 - I want to tap on a user's profile picture or username in the public feed
 - So that I can view their profile and their public video history.
 - **Priority:** Should Have
 - **Acceptance Criteria:**
 - Given I am viewing the public feed
 - When I tap on a user's profile picture or username
 - Then I am navigated to that user's profile screen.
 - Given I am on a user's profile screen
 - When the screen loads
 - Then I see their profile picture, username, and a list of their past public videos.
- **US-13: Like Public Video**
 - As a logged-in user
 - I want to 'like' a public video
 - So that I can show appreciation.
 - **Priority:** Should Have
 - **Acceptance Criteria:**
 - Given I am viewing a video on the public feed
 - When I tap the like icon
 - Then the icon changes its color to red, the like count updates, and my user ID is recorded as having liked the video in Firestore.
 - Given I have already liked a video
 - When I tap the like icon again
 - Then the icon changes back, the like count decreases, and my user ID is removed from the likes list.
- **US-14: Open Video Fullscreen**
 - As a user

- I want to tap a video preview on the profile page
- So that I can view it in a larger, full-screen player.
- **Priority:** Should Have
- **Acceptance Criteria:**
 - Given I am viewing a list of videos
 - When I tap on a video card
 - Then I am navigated to a new screen with a larger video player, and the video starts playing.
- **US-15: Access Settings**
 - As a logged-in user
 - I want to tap my profile picture in the app bar
 - So that I can access the settings screen.
 - **Priority:** Must Have
 - **Acceptance Criteria:**
 - Given I am logged in and on the home screen
 - When I tap the circular profile picture avatar in the top-right
 - Then I am navigated to the Settings screen.
- **US-16: Change Profile Picture**
 - As a logged-in user
 - I want to change my profile picture from the settings screen
 - So that I can personalize my profile.
 - **Priority:** Should Have
 - **Acceptance Criteria:**
 - Given I am on the Settings screen
 - When I tap 'Change Profile Picture' and select an image from my gallery
 - Then the image is uploaded, my profile picture URL is updated in Firestore, and the new picture appears in the app bar (and on my public videos).
- **US-17: Change Username**
 - As a logged-in user
 - I want to change my username from the settings screen
 - So that I can update my display name.
 - **Priority:** Should Have
 - **Acceptance Criteria:**
 - Given I am on the Settings screen
 - When I tap 'Change Username', enter a new valid username (≥ 4 chars), and confirm
 - Then my username is updated in Firestore and displayed correctly in the app.

- **US-18: Upgrade to Premium (Payment)**

- As a free user
- I want to upgrade to Premium by making a payment via app-store billing systems.
- So that I can record longer videos.
- **Priority:** Should Have
- **Acceptance Criteria:**
 - Given I am a free user on the Settings screen
 - When I tap 'Go Premium' and initiate the payment process
 - Then I am guided through the in-app purchase system's payment flow (e.g., entering card details securely).
 - Given the payment is successful
 - When the app receives confirmation
 - Then my user status is updated to 'isPremium: true' in Firestore, and I see confirmation in the app.
 - Given the payment fails
 - When the app receives confirmation
 - Then I see an error message, and my status remains free.
 - Given I am a premium user on the Settings screen
 - When I view the premium option
 - Then I see an indication that I am already a premium member.

- **US-19: Change Email/Password**

- As a logged-in user
- I want to change my account email or password from the settings screen, requiring re-authentication
- So that I can manage my account security.
- **Priority:** Should Have
- **Acceptance Criteria:**
 - Given I am on the Settings screen and tap 'Change Email' or 'Change Password'
 - When I successfully enter my current password and the new details
 - Then my email/password is updated in Firebase Authentication.

- **US-20: Delete Account**

- As a logged-in user
- I want to delete my account permanently from the settings screen, requiring re-authentication
- So that all my data is removed.
- **Priority:** Should Have

- **Acceptance Criteria:**
 - Given I am on the Settings screen and tap 'Delete Account'
 - When I confirm the action and successfully enter my current password
 - Then my Firebase Auth user is deleted, and associated data are removed.
- **US-21: Receive Daily Reminder**
 - As a user who hasn't recorded today
 - I want to receive a push notification in the evening
 - So that I am reminded to record my daily video.
 - **Priority:** Should Have
 - **Acceptance Criteria:**
 - Given I have granted notification permissions and haven't recorded today
 - When it is a time that I set (e.g. 7.00 PM)
 - Then I receive a push notification with a reminder message.

3.2 Use Cases

UC-01: Record and Save Daily Video

- **Actor(s):** Registered User (Free), Registered User (Premium)
- **Preconditions:**
 - User is logged in.
 - User has not recorded a video today.
 - Camera and microphone permissions are granted.
- **Main Flow:**
 - User taps the 'Record' tab/icon on the home screen.
 - System displays the camera preview screen.
 - User (optionally) switches between front/back camera using the flip icon.
 - User (optionally) uses pinch gesture to adjust zoom.
 - User taps the record button.
 - System starts recording video and audio.
 - System displays a timer showing elapsed recording time and a progress bar indicating time remaining until the limit (1 min or 5 min).
 - User taps the stop button OR the time limit is reached.

- System stops recording and navigates to the 'Save Video' screen, passing the recorded video file.
- User enters a title for the video.
- User (optionally) toggles the 'Make Public' switch.
- User taps 'Save & Upload'.
- System compresses the video file.
- System uploads the compressed video to Firebase Storage.
- System retrieves the video's download URL.
- System saves video metadata (URL, title, public status, user info, timestamp) to Firestore.
- System navigates the user back to the home screen (e.g., 'My Diary' tab).
- System displays a success message.
- **Alternative Flows:**
 - **8a. User cancels recording (if a cancel button exists):** Recording stops, video file is discarded, user remains on the camera screen.
 - **12a. Upload fails (network error, etc.):** System displays an error message. User remains on the 'Save Video' screen and can retry.
 - **16a. Saving metadata fails:** System displays an error message. (Requires logic for handling potentially orphaned video files in storage).
- **Postconditions:**
 - A video file is stored in Firebase Storage.
 - A corresponding video document is created in Firestore.
 - User is returned to the home screen.

UC-02: Upgrade to Premium

- **Actor(s):** Registered User (Free), App Store Billing Systems
- **Preconditions:**
 - User is logged in.
 - User's isPremium status in Firestore is false..
 - User has a stable internet connection.
- **Main Flow:**
 - User navigates to the Settings screen.
 - User taps the 'Go Premium' option.
 - System displays a dialog outlining premium benefits.
 - User taps 'Upgrade Now'.

- System initiates the payment process with the app-store billing system, potentially redirecting the user or displaying the billing system's secure payment interface.
- User enters their payment details into the billing system's interface and confirms the payment.
- The billing system processes the payment.
- The billing system sends a success confirmation back to the application.
- System verifies the payment success.
- System updates the user's document in Firestore, setting isPremium true.
- System displays a success message to the user.
- System updates the Settings screen UI to reflect the premium status.
- **Alternative Flows:**
 - **4a. User taps 'Maybe Later':** Dialog closes, user remains on the Settings screen with free status.
 - **7a. Payment fails (insufficient funds, incorrect details, etc.):** The billing system sends a failure confirmation back to the application.
 - **9a. System receives payment failure:** System displays an appropriate error message to the user. User remains on the Settings screen with free status.
- **Postconditions:**
 - (Success): User's isPremium status in Firestore is true. User has access to premium features.
 - (Failure): User's isPremium status remains false.

3.3 Functional Requirements List

- **FR-1:** System shall provide user registration via email, username, password.
- **FR-2:** System shall authenticate users via email and password.
- **FR-3:** System shall allow users to sign out.
- **FR-4:** System shall allow users to record video using front/back cameras.
- **FR-5:** System shall support pinch-to-zoom during camera preview.

- **FR-6:** System shall enforce a 1-minute recording limit for free users.
- **FR-7:** System shall enforce a 5-minute recording limit for premium users.
- **FR-8:** System shall automatically stop recording when the time limit is reached.
- **FR-9:** System shall allow users to save recorded videos with a title.
- **FR-10:** System shall allow users to mark videos as public or private upon saving.
- **FR-11:** System shall upload video files to Firebase Storage.
- **FR-12:** System shall save video metadata (URL, title, UID, public status, timestamp, username, profilePicUrl, likes) to Firestore.
- **FR-13:** System shall prevent users from initiating a new recording if one has already been saved for the current day.
- **FR-14:** System shall display a user's private videos chronologically.
- **FR-15:** System shall display public videos created within the last 24 hours chronologically.
- **FR-16:** System shall allow users to 'like' and 'unlike' public videos.
- **FR-17:** System shall display a user's profile with their username, profile picture, and public videos.
- **FR-18:** System shall allow users to navigate to other users' profiles from the public feed.
- **FR-19:** System shall allow users to view videos in a full-screen player.
- **FR-20:** System shall provide a settings screen accessible via the user's profile picture.
- **FR-21:** System shall allow users to change their profile picture.
- **FR-22:** System shall allow users to change their username.
- **FR-23:** System shall process premium subscription payments via the native in-app purchase system's payment gateway.
- **FR-24:** System shall allow users to change their email (requires re-authentication and verification).
- **FR-25:** System shall allow users to change their password (requires re-authentication).
- **FR-26:** System shall allow users to delete their account (requires re-authentication).
- **FR-27:** System shall send daily reminder notifications to users who haven't recorded.

4. Non-Functional Requirements

4.1 Performance Requirements

- **NFR-1:** Screen transitions should feel smooth (< 300ms).
- **NFR-2:** Video playback should start within 3 seconds on a stable connection.
- **NFR-3:** Public feed should load initial videos within 4 seconds.
- **NFR-4:** Camera preview should display at a minimum of 24 FPS.
- **NFR-5:** Video upload time depends on network, but compression should be used to minimize size (< 50MB for 1 min, < 250MB for 5 min target).
- **NFR-6:** Payment processing via the native in-app purchase system should complete or provide feedback within 10 seconds under normal network conditions.

4.2 Security Requirements

- **NFR-7:** User authentication must be handled securely via Firebase Authentication.
- **NFR-8:** The application must comply with KVKK.
- **NFR-9:** Passwords must not be stored in plain text.
- **NFR-10:** Firestore rules must prevent unauthorized access to private videos and user data modification.
- **NFR-11:** Firebase Storage rules must prevent unauthorized file uploads/deletions.
- **NFR-12:** Sensitive operations (change email/password, delete account) must require re-authentication.
- **NFR-13:** Firebase App Check should be configured for production release to prevent abuse.
- **NFR-14:** Payment processing integration with native in-app purchase system must be implemented securely.

4.3 Usability Requirements

- **NFR-15:** The UI should be intuitive and follow standard mobile design patterns (iOS/Android).
- **NFR-16:** Key actions (record, save, like, upgrade) should be easily discoverable.
- **NFR-17:** Error messages (including payment errors) should be clear and user-friendly.
- **NFR-18:** The app should provide visual feedback for ongoing processes (e.g., uploading, payment processing).

4.4 Reliability and Availability

- **NFR-19:** The app relies on Firebase backend availability (typically > 99.9%).
- **NFR-20:** Premium features rely on native in-app purchase service's availability.
- **NFR-21:** The app should handle network errors gracefully (e.g., show messages, allow retries where appropriate, especially during payment).
- **NFR-22:** Unhandled exceptions should be minimized through proper error handling.

4.5 Maintainability Requirements

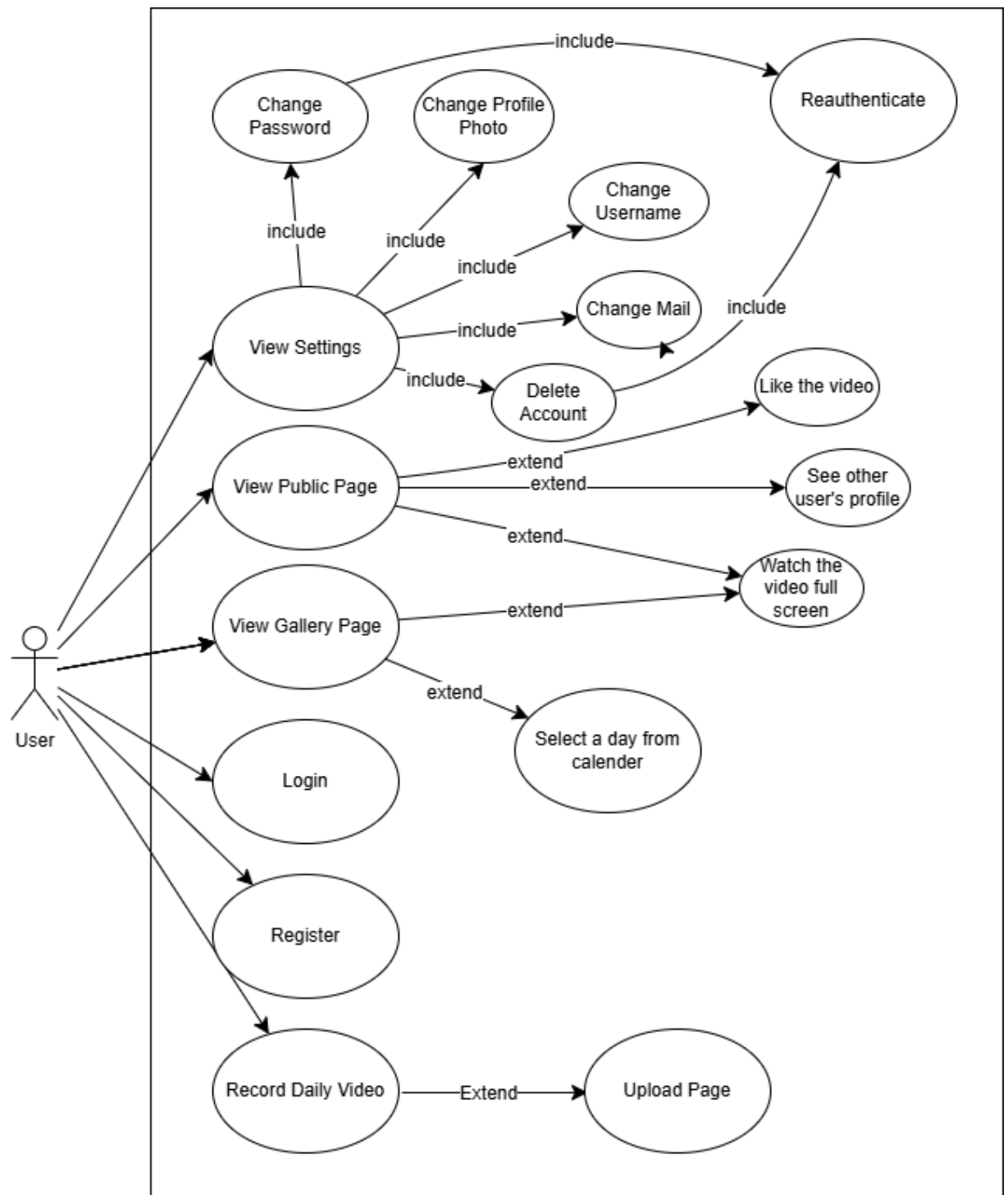
- **NFR-23:** Code should follow effective Dart conventions and Flutter best practices.
- **NFR-24:** Code should be organized logically into modules/layers (screens, widgets, services).
- **NFR-25:** Major functions, including payment integration points, should be documented with comments.

4.6 Compatibility Requirements

- **NFR-26:** Target latest stable versions of iOS and Android supported by Flutter.
- **NFR-27:** Ensure compatibility with common device screen sizes and aspect ratios.
- **NFR-28:** Ensure Native in-app purchase system is compatible with target platforms.

5. System Models (Diagrams)

5.1 Use Case Diagram



5.2 User Flow Diagram

