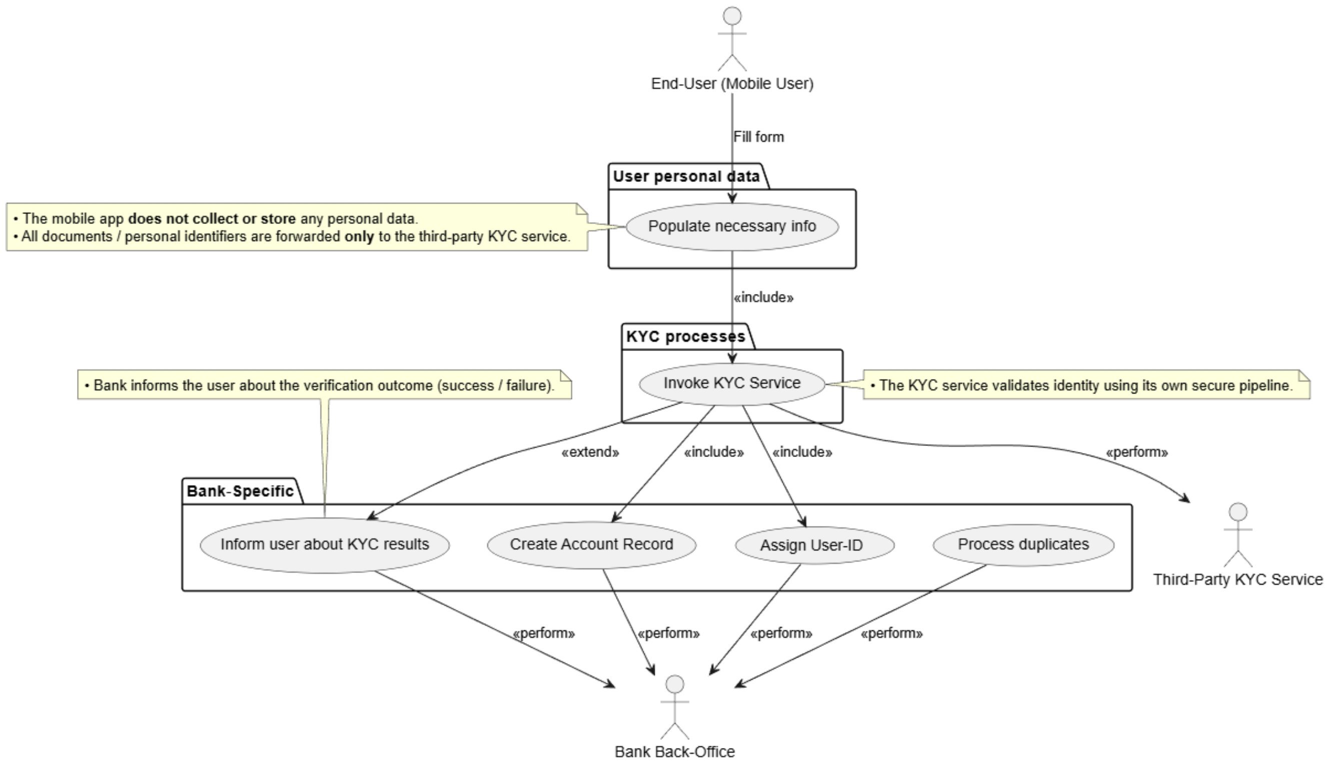
Use Case diagram



- The mobile app **does not collect or store** any personal data.
- All documents / personal identifiers are forwarded **only** to the third-party KYC service.

- Bank informs the user about the verification outcome (success / failure).

- The KYC service validates identity using its own secure pipeline.

US#1 - Mobile registration of the new user

As a New User, I want to register in mobile application so then I can use it for getting Bank services anywhere.

Acceptance criteria

1. The application must be installed with all necessary permission for it requested.
2. Initial checks
    a. the application was re-installed and any local files should be ignored if last action on the server side was uninstall (to avoid fraud).
    b. minimum requirements are met (hardware and software) on the app launch.
    c. major update of the application is available on the app launch and block further actions until app is updated.
    d. necessary permissions are granted (internet, A-GPS, notifications) and prompt to provide such permissions if not enabled. Notifications are not strictly required, other requirements are.
    e. local files include valid access token and bypass registration steps in the such a case.

1. When launch checks pass and no active user is found, the system must prompt a registration screen.

2. The User must be able navigate between registration screen, see the current step of the progress and be able to update information before application.
3. The system must use 'secondary' communication channels to verify user's e-mail and phone number with one-time passwords.
    a. password TTL is 5 minutes
    b. password must be unique and hard to guess but easy to use (6 digits)
    c. the system must block further steps (KYC) before e-mail and phone number are verified
4. Embedded KYC web view
    a. The KYC provider’s URL must be loaded in an in-app WebView configured with:
    b. HTTPS with TLS 1.3 only
    c. The WebView must enforce certificate pinning to the KYC provider’s public key.
    d. The provider’s page must set X-Frame-Options: SAMEORIGIN to prevent click-jacking.
    e. The KYC web view must load (i.e., receive the first HTML response) within 3 seconds on a 3G connection for 95 % of sessions under normal load.
    f. All KYC-related API calls must be logged with a correlation ID that can be traced back to the stored session token for forensic review.
5. Session persistence
    a. Upon successful launch of the KYC web view, the server returns a KYC session token (UUID).
    b. The token must be stored encrypted in the Android EncryptedSharedPreferences (or iOS Keychain) under the key kyc_session_token.
    c. On app resume or re-launch, the stored token must be read; if it is still valid (server confirms), the app must restore the UI to the last pending step.
    d. When the KYC flow finishes (success or failure), the token must be cleared from storage.
6. Security of data in transit
    a. No plain-text logs or screenshots may capture the KYC payload; any debug output must be stripped of PII before being written.
7. The app must never write scanned documents or personal data to the device’s external storage or cache directory.
8. After a successful upload, the in-memory buffer containing the document must be overwritten and the reference released
9. Error handling & fallback
    a. If the WebView cannot be rendered (e.g., network unavailable, certificate error), display a retry dialog with an explanatory message and retain the stored session token.
    b. If the KYC provider returns an unrecoverable error, the app must navigate the user back to the registration wizard with a clear error description.

US#2 - KYC invoke process inside the mobile app

As the banking platform, I need to embed the third-party KYC verification flow inside the mobile app and persist the user's progress so that the user can complete identity verification without leaving the app or re-entering data.

Acceptance criteria

1. The system must be able to invoke a KYC process as a third-party web-application inside the banking application for smooth experience.
2. The system must memorize current progress to support disconnected sessions (when user is distracted and returns back to the app later).
3. The system must provide a secured connection inside the app so then all data cannot be intercepted.
4. The system must process KYC results asynchronous and deliver e-mail/SMS to the user with results as a short link to continue registration or use of the application.
5. Upon successful launch of the KYC web view, the server returns a KYC session token (UUID).
6. The token must be stored encrypted in the Android EncryptedSharedPreferences (or iOS Keychain) under the key kyc_session_token.
7. On app resume or re-launch, the stored token must be read; if it is still valid (server confirms), the app must restore the UI to the last pending step.
8. When the KYC flow finishes (success or failure), the token must be cleared from storage.
9. All HTTP/HTTPS traffic between the app and the KYC provider must use TLS 1.3. No plain-text logs or screenshots may capture the KYC payload; any debug output must be stripped of PII before being written.
10. The app must never write scanned documents or personal data to the device's external storage or cache directory.
11. After a successful upload, the in-memory buffer containing the document must be overwritten and the reference released.
12. If the WebView cannot be rendered (e.g., network unavailable, certificate error), display a retry dialog with an explanatory message and retain the stored session token.
13. If the KYC provider returns an unrecoverable error, the app must navigate the user back to the registration wizard with a clear error description.
14. The KYC web view must load (i.e., receive the first HTML response) within 3 seconds on a 3G connection for 95 % of sessions under normal load.
15. All KYC-related API calls must be logged with a correlation ID that can be traced back to the stored session token for forensic review.