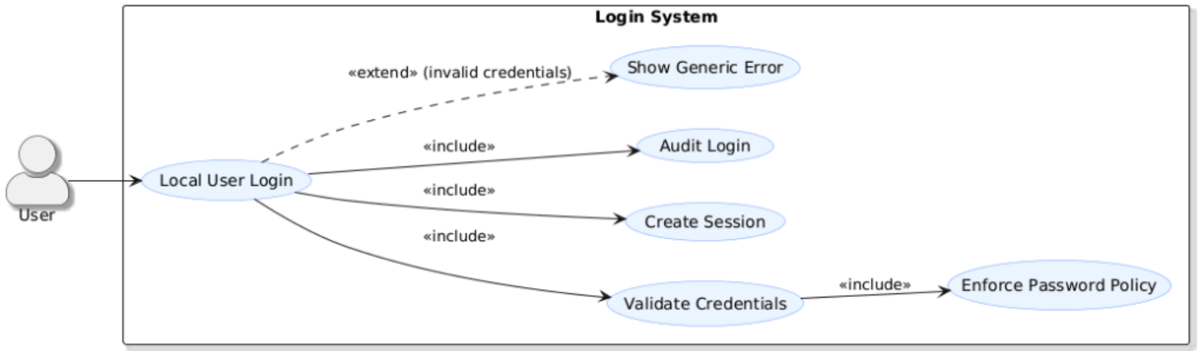
Maintenance Planning

Local User Login

Requirements Modeling & Specification

Selected Feature: Local User Login — username/email + password against DB.

Use Case Diagram



Software Requirement Specification (SRS) Snippets

Functional Requirements

Local Login (Email/Password)

- The system must allow users to sign in using their registered email and password in addition to the existing Google sign-in.
- The system must send the login credentials securely over HTTPS only.
- The system must verify the credentials using the Supabase Authentication service (`signInWithPassword`).
- The system must respond with a generic error message for invalid credentials to prevent user enumeration.
- The system must issue a HttpOnly, Secure, and SameSite=Lax cookie containing the session token upon successful authentication.
- The system must rotate the session ID after each successful login to prevent session fixation.
- The system must ensure that users can access all authorized features only after successful authentication.
- The system must log each login attempt (success/failure) with a timestamp and user ID for administrative monitoring.

Non-Functional Requirements

Security

- The system must not store plaintext passwords or secrets.
- The system must enforce a recommended password policy (≥8 characters, containing uppercase, lowercase, numbers, and special characters).
- The system must provide generic error messages to avoid exposing valid account information.

Performance

- Login requests must complete within 3 seconds under normal conditions.

Usability

- The login form must be accessible via keyboard and support screen readers.
- The system must ensure the login page is responsive across desktop and mobile devices.

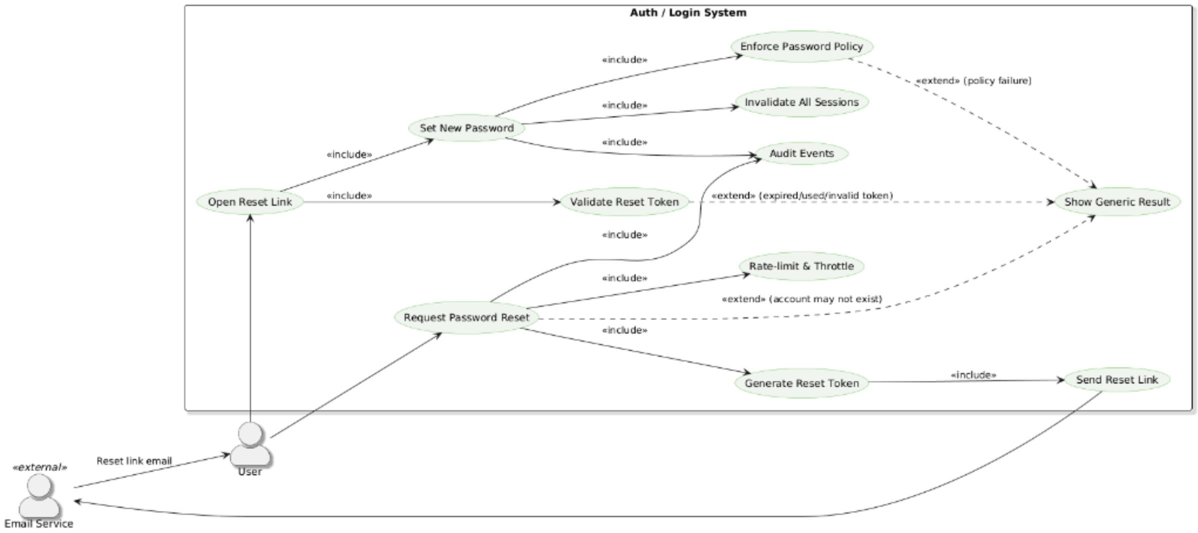
Self-Service Password Reset

Requirements Modeling & Specification

Selected Features

Self-Service Password Reset (SSPR) ← focus of this document

Use Case Diagram



Software Requirement Specification (SRS) Snippets

FR-SSPR-00 — Rate-limit & Throttle (cross-cutting, «include»)

- **Description:** The system shall limit reset requests to prevent abuse.
- **Rules (example: tune as needed):**
  - Per account: **≤ 5 requests/hour**; Per IP: **≤ 20 requests/hour**.
  - When exceeded → return **429/generic** response (see FR-SSPR-05) and **record an audit event** (FR-SSPR-04).

FR-SSPR-01 — Request Password Reset

- **Description:** Allow an unauthenticated user to request a password reset using a registered email.
- **Preconditions:** User is not authenticated; email format is valid.
- **Basic Flow:**

    - Validate email format and apply **FR-SSPR-00**.
    - Call **FR-SSPR-01a Generate Reset Token**.
    - Call **FR-SSPR-01b Send Reset Link**.
    - Return **generic** success message (FR-SSPR-05).
- **Postconditions:** A **single-use** token (stored as a **hash**) with **TTL = 15 minutes** is created; email is queued/sent; **Audit** recorded (FR-SSPR-04).
- **Alternates/Errors:**
    - Email not found → still returns **generic** success (no enumeration).
    - Email send failure → log + retry queue; still return generic success.

FR-SSPR-01a — Generate Reset Token («include»)

- Generate a cryptographically secure token (≥128-bit).
- Store: `token_hash`, `user_id`, `expires_at`, `used=false`.
- Ensure the **uniqueness** of `token_hash`; the token becomes invalid when expired or marked used.

FR-SSPR-01b — Send Reset Link («include»)

- Send link `/reset?token=…` via Email Service.
- Email copy must not reveal whether an account exists; state that the link **expires in 15 minutes**; include safety guidance.

FR-SSPR-02 — Open Reset Link / Validate Token («include»)

- **Description:** Validate the token and render the "Set New Password" page.
- **Preconditions:** Token exists, unexpired, unused; hash matches.
- **Postconditions:** Reset form is shown (optionally mark token as **pending**).
- **Errors (extend → FR-SSPR-05): invalid/expired/used** token → show generic "link invalid/expired" and provide a **Request New Link** action.

FR-SSPR-03 — Set New Password («include/extend»)

- **Description:** Accept the new password and complete the reset.
- **Preconditions:** Valid token; password meets policy.
- **Basic Flow:**
    - Verify token; call **FR-SSPR-03a Enforce Password Policy**.
    - Hash password with **Argon2id** (or strong bcrypt).
    - Atomically update user credential; set token `used=true`.
    - Call **FR-SSPR-03b Invalidate All Sessions**.
    - Record **Audit** (FR-SSPR-04) and show success (or redirect to login).
- **Errors:**

    - Weak password → show policy and allow retry.
    - Token invalid/expired (race) → use **FR-SSPR-05** and suggest a new request.

FR-SSPR-03a — Enforce Password Policy («extend» from Set New Password)

- Minimum length **≥ 8**; includes upper/lowercase, digit, special character (adjustable).
- (Optional) Breach check; prevent reuse of last **N** passwords (e.g., last 3).

FR-SSPR-03b — Invalidate All Sessions («include»)

- Revoke all active sessions/refresh tokens for the user immediately after a successful reset.

FR-SSPR-04 — Audit Events («include»)

- Record events: request submitted, rate-limit blocked, token issued, token invalid/expired/used, reset success/failure.
- Store: `user_id` (if known), `ip`, `user_agent`, `event_type`, `timestamp`.
- Provide a dashboard or export for security review.

FR-SSPR-05 — Show Generic Result («extend»)

- Use **generic** messages in any case that could reveal account existence or token state, e.g.:
    - "If this email exists in our system, we've sent a reset link."
    - "This link is invalid or expired. Please request a new one."
- Do **not** say "email not found" or disclose token specifics.

NFR-SSPR (Non-functional)

- **Security:** HTTPS only; token stored as **hash**; generic responses; rate-limit; CSRF protection on forms; cookies use **HttpOnly**, **Secure**, **SameSite**.
- **Performance:** P95 page load **≤ 3 s**; email dispatch typically **≤ 10 s**.
- **Availability:** SSPR endpoints meet overall Auth SLA.
- **Privacy/Compliance:** Emails avoid disclosing account existence; purge logs by policy.
- **Accessibility/UX:** Clear labels/messages; keyboard and screen-reader friendly; localized copy as needed.