TOBB ETU
Economy & Technology University
BIL 481

Acceptance Tests and Criteria
Group: EYAY
Project: HAYAI ET

EYAY

Ali Türkücü

Esra Alparslan

Muhammed Yusuf Kartal

Yağız Can Akay

Contents

1. Acceptance Criteria

These criteria define the "Definition of Done" for the four core scenarios selected for the demo. They are derived directly from the Functional Requirements (FR) and Constraints (SWC/NFR) defined in previous assignments.

Scenario 1: Login & Sign-up

- **AC 1.1:** The system must prevent registration without a valid parent email/verification code (Traceability: FR-1).
- **AC 1.2:** Passwords must be hashed using Argon2 or Bcrypt; no plain text passwords shall be stored in the database (Traceability: FR-1, FR-12, NFR-2).
- **AC 1.3:** Successful login must return a valid JWT (JSON Web Token) for session management (Traceability: FR-2).

Scenario 2: Upload Drawings & Transform with AI

- **AC 2.1:** The system must accept only .jpg or .png files under 10 MB in size (Traceability: FR-3, Constraint SWC-004).
- **AC 2.2:** The AI transformation process must complete and return a generated result within 120 seconds (Traceability: NFR-3, Constraint SWC-003).
- **AC 2.3:** Both the Original and AI-generated images must be saved to the user's personal library immediately after generation (Traceability: FR-5).
- **AC 2.4:** The system must automatically screen content using AI moderation. Inappropriate content must be flagged as "Rejected" and blocked from display (Traceability: FR-11).

Scenario 3: Search Friends & View Their Libraries

- **AC 3.1:** The search function must return users based on partial username matches (Prefix matching) (Traceability: FR-9).
- **AC 3.2:** A user cannot view another user's "Private" or "Rejected" posts; only "Public" and "Approved" posts are visible in the gallery grid (Traceability: FR-6, NFR-5).
- **AC 3.3:** Users must have the ability to toggle the visibility of their posts ("Public"/"Private") to control sharing (Traceability: FR-6).

Scenario 4: Like & Preset-Comment

- **AC 4.1:** Users can "Like" a post. The action must be idempotent (clicking twice does not add two likes) (Traceability: FR-7).

- **AC 4.2:** Users can ONLY submit comments from the predefined list. Any free-text input sent via API manipulation tools must be rejected by the backend (Traceability: FR-8, FR-11, Constraint SWC-002).

2. Acceptance Test Cases

The following test cases were designed to verify the acceptance criteria on the deployed environment.

Test Case 1: Safe Registration & Authentication
- **Associated Scenario:** Login & Sign-up
- **Correspondence to AC:** Verifies AC 1.1, AC1.2 and AC 1.3.
- **Pre-conditions:** User "DemoChild1" does not exist in the database.
- **Test Steps:**
    1. Navigate to the /register page.
    2. Enter Username: DemoChild1, Password: Password123, Parent Email: parent@test.com.
    3. Submit the form.
    4. Log in with the newly created credentials.
- **Expected Result:** Registration returns HTTP 201. Login returns HTTP 200 with a valid Access Token. User is redirected to the Dashboard.

Test Case 2.1: AI Transformation Flow
- **Associated Scenario:** Upload Drawings & Transform with AI
- **Correspondence to AC:** Verifies AC 2.1, AC 2.2 and AC 2.3.
- **Pre-conditions:** User is logged in. A valid drawing file (sun_drawing.jpg, 2MB) is available.
- **Test Steps:**
    1. Click the "Upload" button and select sun_drawing.jpg.
    2. Select the Theme "Space Adventure" from the dropdown.
    3. Click "Reimagine" and wait for the process to complete.
    4. Navigate to "My Library".
- **Expected Result:** The loading spinner appears. Within 2 minutes, the result page displays the original drawing side-by-side with a space-themed AI version. The image is persisted in the library.

Test Case 2.2: AI Safety & Policy Enforcement

- **Associated Scenario:** Upload Drawings & Transform with AI
- **Correspondence to AC**: Verifies AC 2.4 and AC 3.2.
- **Pre-conditions:** User is logged in. A test image containing policy-violating content is available.
- **Test Steps:**
  1. Upload the flagged image and click "Reimagine".
  2. Observe the UI notification and API response.
  3. Navigate to "My Library" to verify visibility.
- **Expected Result:** The backend immediately blocks the process with an HTTP 400 Bad Request error displaying a policy violation message; the image is marked as "rejected" and "private" in the database and remains completely invisible in the user's "My Library" grid.

Test Case 2.3: File Size Limit

- **Associated Scenario:** Upload Drawings & Transform with AI
- **Correspondence to AC:** Verifies AC 2.1.
- **Pre-conditions:** User is logged in. A test file exceeding the limit is available.
- **Test Steps:**
  1. Navigate to the "Upload" page.
  2. Select and attempt to upload the large_file.jpg.
  3. Observe the error message and system response.
- **Expected Result:** The system immediately rejects the upload with an HTTP 413 Payload Too Large error, and no record is created in the database for this file.

Test Case 3: Gallery Visibility & Privacy

- **Associated Scenario:** Search Friends & View Their Libraries
- **Correspondence to AC:** Verifies AC 3.2.
- **Pre-conditions:** User A has 1 Public Post and 1 Private Post. User B is logged in.
- **Test Steps:**
  1. User B searches for User A.
  2. User B clicks on User A's profile to view their gallery.
  3. Count the visible posts in the grid.
- **Expected Result:** User B sees exactly **one** post (the Public one). The Private post is hidden from the view.

Test Case 4: Social Safety Constraint (Anti-Free-Text)

- **Associated Scenario:** Like & Preset-Comment
- **Correspondence to AC:** Verifies AC 4.2.

- **Pre-conditions:** User is logged in and viewing a friend's post (ID: post_123).
- **Test Steps:**
    1. **UI Test:** Click "Comment" and select "Great Job!" from the dropdown. Verify it posts.
    2. **Security Attack Test:** Use Postman to send a POST request to /api/posts/post_123/comment with a custom payload: {"content": "This is a custom bad word"}.
- **Expected Result:**
    - UI Test: Comment appears successfully.
    - Security Attack: Backend returns 422 Validation Error or 400 Bad Request. The custom text is **NOT** saved to the database.

3. Document-Specific Task Matrix

Document Task	Esra	Yağız	Yusuf	Ali
Document Planning			✅	
Acceptance Criteria			✅	✅
Acceptance Test Cases			✅	✅
Formatting & Final Review	✅	✅	✅	✅