Features for Project Proposal

Team Number: 4
Team Members:

- Jaiden Lo (93978203)
- Takumi Choi (37325289)
- Anilov Laxina (36694933)
- Kussh Satija (80384878)
- Kiichiro Suganuma (19743749)
- Aliff Razak (58423609)

1 Project Scope and Usage Scenario

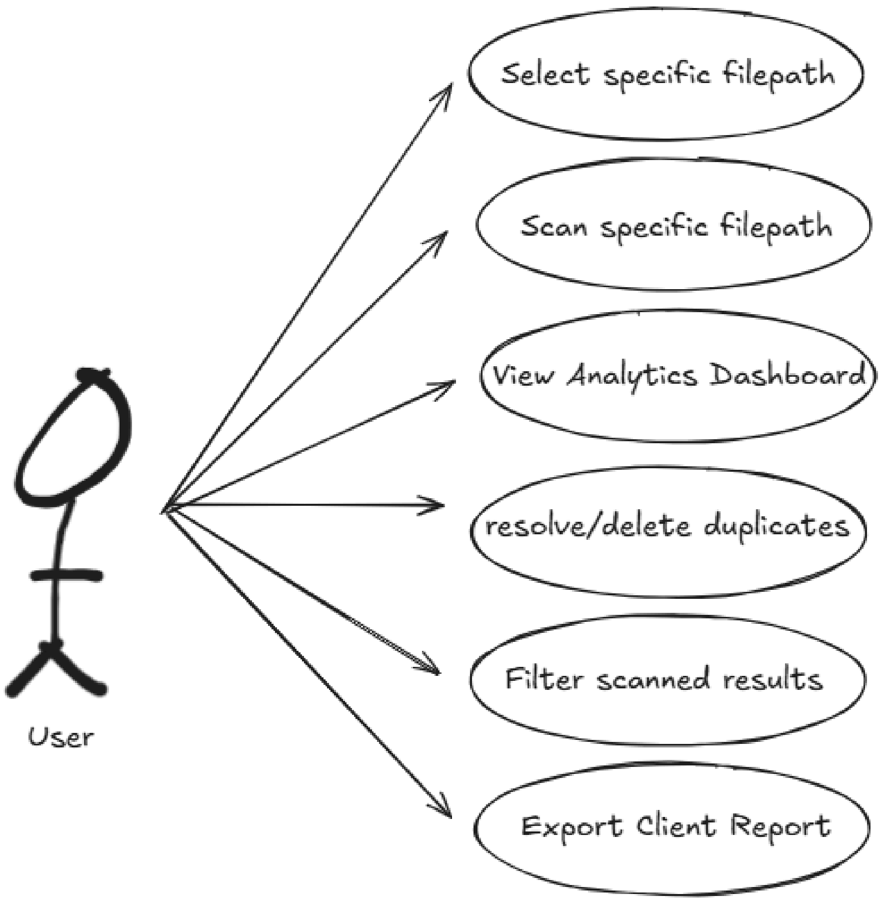
Our target users are photographers and designers, and our project focuses on helping them efficiently organize and analyze their growing media libraries. A photographer/designer selects a local folder or connected cloud drive such as Google Drive or OneDrive for the app to scan through. The system scans files such as `.png`, `.jpg`, `.psd`, `.heic` , extracts metadata such as file size, resolution, duration and EXIF data, detects duplicates, and groups assets by client or project. A project manager signs in to view dashboards that display counts by type, per-client totals, largest files, and recent uploads, and can also export a quick report for stakeholders. An admin configures storage integrations, manages user roles, and sets scan rules including include and exclude folders or privacy settings. Together these roles keep a shared media library organized, searchable, and ready for reporting without manual folder digging.

2 Proposed Solution

Our proposed solution is a media asset analyzer that scans and organizes creative files such as images, and other design artifacts. The system would extract metadata from each file (such as the file type, size, resolution, and creation date) and use this information to generate meaningful insights. For example, a photographer could quickly see how many client projects they worked on in a given month, what percentage of their portfolio comes from a certain camera.

The core features can include automated file scanning, metadata extraction using existing libraries (maybe libraries such as EXIF for photos) and a visual dashboard that displays breakdowns by file type or a project timeline. We want to make it simple for photographers to understand their creative workload at a glance. Compared to other teams our approach stands out because it blends technical analysis with an easy to understand UI. It's almost like a "Spotify Wrapped" (a year-end summary of the amount of music one listens to), but for creative assets.

3 Use Cases



Use Case 1: Configure Storage Integrations

- **Primary actor:** Admin
- **Description:** Connect Google Drive/OneDrive or select local folders; set include/exclude rules.
- **Precondition:** Admin is authenticated.
  Postcondition: Integration saved and verified; rules stored.
  Main scenario: Open Settings → choose provider/folder → authenticate (OAuth or path) → set rules → save → system verifies connection.

- **Extensions:** Invalid credentials / missing scopes / network error → show corrective message; prevent saving.

Use Case 2: Run / Schedule Scan

- **Primary actor:** Creator
- **Description:** Start an on-demand scan or schedule periodic scans.
- **Precondition:** At least one location configured.
- **Postcondition:** Asset metadata and hashes updated in DB.
- **Main scenario:** Click “Scan now” → progress visible → extract metadata + hashes → write to DB → notify completion.
- **Extensions:** Unreadable/corrupt files logged; partial success allowed; retry failed paths.

Use Case 3: View Analytics Dashboard

- **Primary actor:** Creator, Project Manager
- **Description:** View counts by type, per-client totals, largest files, recent activity.
- **Precondition:** A scan has run.
- **Postcondition:** None (read-only).
- **Main scenario:** Open Dashboard → set date/client filters → charts/tables render.
- **Extensions:** Empty dataset → show helpful “get started” state with link to run scan.

Use Case 4: Find & Resolve Duplicates

- **Primary actor:** Creator
- **Description:** Review duplicate/near-duplicate groups and take actions (open location, mark archived).
- **Precondition:** Hashes computed during scan.
- **Postcondition:** Duplicates flagged; actions recorded (e.g., archived).
- **Main scenario:** Open Duplicates → inspect groups (exact vs near) → select assets → apply action.
- **Extensions:** False positive near-dupes → allow ungroup/whitelist.

Use Case 5: Search / Filter Assets

- **Primary actor:** Project Manager
- **Description:** Find assets by client, date range, type, size, resolution/duration.
- **Precondition:** Indexed data exists.
- **Postcondition:** None.
- **Main scenario:** Enter filters → results list updates with previews/paths.
- **Extensions:** Very large result sets → paginate; invalid ranges → validation error.

Use Case 6: Export Client Report

- **Primary actor:** Project Manager

- **Description:** Export CSV/PDF of per-client inventory and highlights.
- **Precondition:** Filtered dataset ready.
- **Postcondition:** File generated and downloaded.
- **Main scenario:** Click Export → choose CSV/PDF → system generates → download.
- **Extensions:** Long export → background job with toast on completion.

Use Case 7: Manage Users & Roles

- **Primary actor:** Admin
- **Description:** Create users; assign roles (Creator/PM/Admin).
- **Precondition:** Admin authenticated.
- **Postcondition:** Users can log in with correct permissions.
- **Main scenario:** Add user → assign role → save → invite or credentials issued.
- **Extensions:** Duplicate email / invalid role → validation error.

4 Requirements, Testing, Requirement Verification

Tech Stack

Frontend: React + Tailwind + Recharts (React component).
Backend API: Node.js (Express).
Processing Service: Python (FastAPI + ffmpeg + EXIFread).
Database: MySQL
Queue: Redis + BullMQ.
Deployment: Docker Compose locally, with optional cloud deploy on Heroku.
Test framework: Pytest (backend), Jest
CI/CD: GitHub Actions

Functional Requirements Table

Requirements	Description	Test Cases	Who	H/D/E
Search and Scan	System scans a user selected file and records file metadata	• Scans folder with different filetypes (jpg, mp4, psd) • Scan invalid paths, returns error and handles • Accessing private	Jaiden, Takumi	Hard

		files, error failure message		
Metadata Extraction	Extract EXIF (images) codec/duration of videos, PSD properties	● Extract EXIF from jpg, metadata retained ● File is corrupted, skipped with errors logged ●	Kiichiro	Medium
Generate insights	Analyze stored metadata to create a dashboard	● Dashboard loads with real aggregated data ● Export the report that generates a CSV/PDF ● No photos uploaded should have “No insights available”	Anilov, Aliff	Hard
Insights dashboard	Charts that display the total number of files by type, the total number of files for each client, the largest files in the system and most recently added files	● The API returns the correct aggregated values for the requested data. ● The frontend correctly renders the charts and tables based on the API results. ● The filter options work as intended and update the displayed data accordingly.	Kussh	Easy
Project timeline creation	Showcases the time taken to complete a project with version history if available	● The system correctly calculates the project’s start and end dates using the earliest and latest file modification timestamps. ● The system updates the project timeline	Kussh, Kiichiro	Hard

		when new files are added or existing files are modified. ● The system shows version history entries when versioning information, such as Git commits, is available.		
Reports export	Export project/client statistics as a CSV or PDF	● Export CSV, if its correct rows, ● Export PDF, readable report with proper charts ● Invalid format request, error shown.	Anilov	Medium

Non-Functional Requirements Table

Requirements	Description	Test Cases	Who	H/D/E
Delivery Match Evaluation (Non-functional)	Evaluates how well the delivered photos align with the client’s written request.	● The evaluator verifies that the delivered set matches the required number of photos, file formats, aspect ratios, and resolution from the client brief, and the system marks the delivery as successful when all conditions are met. ● The evaluator scores the delivery against a rubric with categories such as subject, location, shot list, and style, and the	Takumi, Jaiden	Hard

		system records both per-category scores and the overall score.		
Usability (Non-functional)	Interfaces should be easy to use and readable (clean UI, keyboard accessible, etc.) Though it requires manual feedback from users/clients	● Letting a new user use application and see how quick they can access the desired tool, if its greater than 45 seconds then it is unsuccessful	Anilov	Easy
Performance (Non-Functional)	Application must scan and return products in a reasonable time	● Scan a folder with numerous kinds of files (jpg,mp4,psd) and verify if it properly outputs the product without crash or error.	Kiichiro	Medium