Software Requirements Specification (SRS) Document

Colour Auto Correction - Team 46
Team Members:
Abhinav Raundhal
Ishan Gupta
Harsh Gupta
Deekshitha Yattapu
Sujal Deoda

Brief problem statement

When photographers capture 1000s of photos in events, they select 100s of good photos after which they perform editing of these photos. Editing 100s of photos manually is hard. The aim of this project is to automatically perform editing on 100s of photos at one time and perform color correction on all of them at once. Here we need to implement the algorithms which calculate the factors by which we need to change or adjust the parameters such as brightness, contrast and color tone of given photos and apply these filters to 100s of photos.

System requirements

Front end - Web application with Python, java script

NOTE :- This is an additional feature to be added to the interface (already implemented by the client) which allows the user to edit the photos based on parameters such as brightness, contrast, saturation, exposure and color.

Back end - SQL database

NOTE :- This is to store the pre-computed parameters in an SQL database, inferred from already existing data sets like Adobe or datasets that are collected from the photographers consisting of raw and edited photographs.

Libraries:

Programming Languages: Majorly python

Libraries: opencv, cv2, numpy, pandas, matplotlib, pillow, sklearn, colour ,Tenserflow, Pytorch

Version Control: Git

Users profile

Professional photographer or Photography Studios:

- *Usage Mode*: These users will use the system in batch processing mode where they can upload hundreds of event photos at once for color correction.
- *Profile*: Proficiency and ease in using computers and photo editing software.

System Administer:

- *Usage Mode*: Be able to the following: database management (ensuring security of photographs' datasets), algorithms maintenance (update algorithms to enhance editing), handle system crashes, provide assistance to users (related to login, usage).
- *Profile*: Should be proficient with SQL database management and understanding web application development using Python and JavaScript, Ability to work collaboratively with a development team.

Feature requirements (described using use cases)

Functional Requirements:

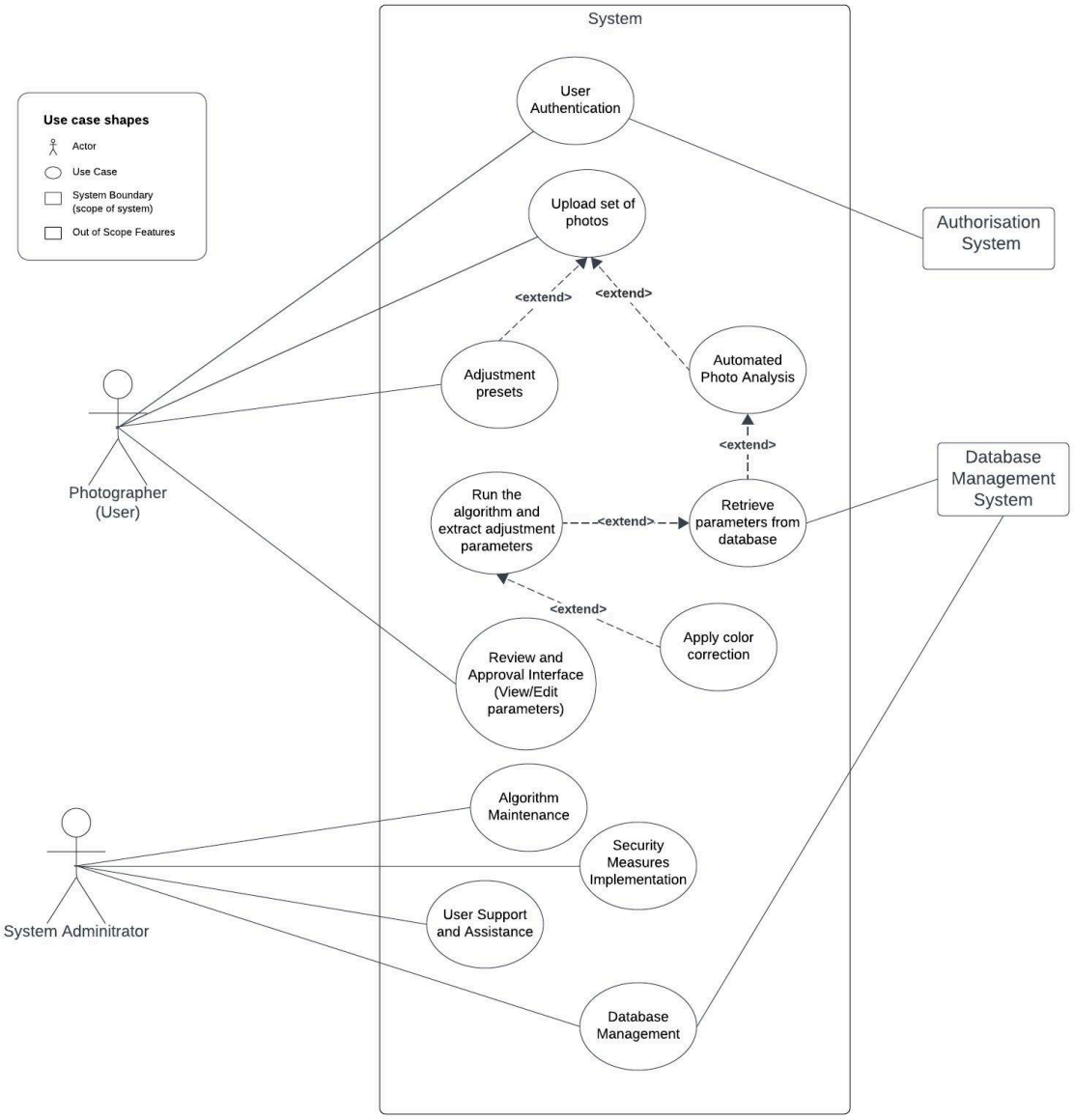
No.	User Case Name	Description	Release
1.	User Authentication	Users can log into the web application securely using their credentials. This ensures that only authorized individuals can access the editing features and review automated edits.	R1
2.	Upload set of photos	Users can upload a batch of photos to the system. The uploaded photos serve as input for the automated editing process.	R1
3.	Automated Photo Analysis	The system automatically analyzes each uploaded photo, extracting parameters such as brightness, contrast, saturation, exposure, and color tone from each photo.	R1
4.	Retrieve parameters from database	This use case involves the system retrieving precomputed parameters from the database to apply them to a new set of photos.	R1

5.	Run the algorithm and extract adjustment parameters	Algorithms applied on image calculate adjustment factors based on the analyzed parameters(obtained in database) to optimize photo editing. The calculated factors are then stored for later use.	R1
6.	Apply color correction	The system applies the calculated adjustment factors to the batch of photos, creating edited versions. Users can review and approve these automated edits.	R1
7.	Review and Approval Interface (View/Edit parameters)	Users are presented with an interface to review the edited photos and either approve or request additional adjustments. This interface is integrated with the existing client interface.	R2
8.	Adjustment presets	The system offers predefined adjustment presets for common color correction scenarios. Users can choose from these presets to apply standardized adjustments quickly.	R1
9.	Algorithm Maintenance	System administrators oversee the functionality of algorithms responsible for calculating adjustment factors. Updates and maintenance are performed to enhance editing accuracy.	R2
10.	Database Management	System administrators can manage the SQL database storing pre-computed parameters. This includes ensuring data integrity, security, and efficient retrieval of stored parameters.	R2

Non functional Requirements:

1. Performance: The system should be able to process and apply color correction to a large number of photos within a reasonable timeframe, ensuring smooth and efficient operation even during peak usage periods.
2. Scalability: The system should be designed to handle increasing numbers of users and photos without significant degradation in performance.
3. Security: The system should implement robust security measures to protect user data, including authentication, authorization ensuring users can't tamper any data which they don't have access to.
4. Compatibility: The system should be compatible with a wide range of devices and web browsers, ensuring accessibility for users regardless of their preferred platform or technology.

Use case diagram



Use case description

Use Case Name:	UC-01
Use Case Name:	User Authentication
Overview:	Users can log into the web application securely using their credentials. This ensures that only authorized individuals can access the editing features and review automated edits..
Actors:	User, System, Authentication System
Pre condition:	The user has valid login credentials.
Flow:	Main (success) Flow: On entering the URL the user id taken to the web application and the Login page is displayed User enters username and password. System validates the credentials. If credentials are valid, the user is granted access.
	Alternate Flows: Invalid Credentials: If the credentials are invalid, the system prompts the user to re-enter them.
Post Condition:	The user is logged into the web application.

Use Case Number:	UC-02
Use Case Name:	Upload set of photos
Overview:	This use case involves users uploading a set of event photos to the system for color correction.
Actors:	Users, System
Pre condition:	The user must have access to the system and have the set of event photos ready for upload.
Flow:	1. User navigates to the upload section of the system. 2. User selects the required set of event photos from their storage. (a) User can upload the folder from their device/drive

	(b) User can drag and drop the folder containing the images 3. User starts the upload process. 4. System processes and stores the uploaded photos for color correction.
	Alternate Flows: Error in Photo Upload: • If there is an error in the photo upload process, the system provides an error message to the user.
Post Condition:	The selected set of event photos is successfully uploaded and ready for analysis and color correction.

Use Case Number:	UC-03
Use Case Name:	Automated Photo Analysis
Overview:	The system automatically analyzes each uploaded photo, extracting parameters such as brightness, contrast, saturation, exposure, and color tone.
Actors:	System
Pre condition:	The photos must have been successfully uploaded by the user
Flow:	1. System automatically analyzes the uploaded photos ,extracting parameters such as brightness,contrast,saturation and color tone. 2. System saves the extracted parameters for further processing.
	Alternate Flows:NA
Post Condition:	The system successfully analyzes the uploaded photo and saves the extracted parameters for further processing.

Use Case Number:	UC-04
Use Case Name:	Retrieve parameters from database
Overview:	This use case involves the system retrieving precomputed parameters from the database to apply them to a new set of photos.
Actors:	System, Database
Pre condition:	Precomputed parameters must be available in the database.
Flow:	1. System retrieves the precomputed parameters from the database. 2. System applies the retrieved parameters to the new set of photos for color correction.
	Alternate Flows: Database Connection Failure: If the system cannot establish a connection to the database, it notifies administrators and attempts reconnection.
Post Condition:	Parameters are successfully retrieved for further processing.

Use Case Number:	UC-05
Use Case Name:	Run the algorithm and extract adjustment parameters
Overview:	Algorithms applied on image calculate adjustment factors based on the analyzed parameters(obtained from database) to optimize photo editing. The calculated factors are then stored for later use.
Actors:	System
Pre condition:	Photo analysis parameters must be extracted and available for calculation.

Flow:	1. System retrieves the analyzed parameters from the database 2. System algorithms calculate adjustment factors based on the analyzed parameters to optimize photo editing. 3. System stores the calculated adjustment factors for later use.
	Alternate Flows: If there is an error in the calculation process, the system logs the error, and administrators are notified.
Post Condition:	The system successfully calculates adjustment factors based on the analyzed parameters and stores them for later use in photo editing.

Use Case Number:	UC-06
Use Case Name:	Apply color correction
Overview:	This use case involves the system applying color correction filters to photos, adjusting color parameters like brightness, contrast, and color tone.
Actors:	System
Pre condition:	Photos must be uploaded and stored in the system and the parameters w.r.t which changes will be made must be present
Flow:	1. System retrieves the uploaded photos. 2. System applies color correction filters to each photo, adjusting parameters based on predefined algorithms. 3. System saves the color-corrected photos.
	Alternate Flows: NA
Post Condition:	The selected photos have been color-corrected according to the applied filters and they go for reviewing.

Use Case Number:	UC-07
Use Case Name:	Review and Approval Interface
Overview:	Users are presented with an interface to review the edited photos and either approve or request additional adjustments. This interface is integrated with the existing client interface.
Actors:	Users, System
Pre condition:	Edited photos must be available for review in the system.
Flow:	1. User accesses the review and approval Interface within the client Interface. 2. System displays the edited photos for review,displaying the applied adjustments and original side by side. 3. User reviews the edited photos and evaluates the Applied adjustments. 4. User selects either the Approve option or request for additional adjustments.
	Alternate Flows: Additional Adjustments Requested: • If the user requests additional adjustments, the system provides options for further editing.
Post Condition:	The user's adjustments to the color correction parameters are saved and applied to the selected photo/ photo being reviewed and User’s decisions are recorded.

Use Case Number:	UC-08
Use Case Name:	Adjustment presets

Overview:	The system offers predefined adjustment presets for common color correction scenarios. Users can choose from these presets to apply standardized adjustments quickly.
Actors:	Users
Pre condition:	Users must have access to the system and a photo available for editing.
Flow:	1. User selects a photo for color correction. 2. User navigates to the adjustment presets section of the system. 3. User browses and selects a predefined adjustment preset . 4. System applies the selected adjustment preset to the photo, automatically adjusting parameters such as brightness, contrast, and color tone. 5. User reviews the applied adjustments and confirms the selection.
	Alternate Flows: User Modification of Preset: If the user modifies the preset, the system applies the modified adjustments.
Post Condition:	The selected adjustment preset is successfully applied to the photo, and the user can proceed with further editing or save the changes.

Use Case Number:	UC-09
Use Case Name:	Algorithm Maintenance
Overview:	System administrators oversee the functionality of algorithms responsible for calculating adjustment factors. Updates and maintenance are performed to enhance editing accuracy.
Actors:	System Administrator, System

Pre condition:	The system is in operation.
Flow:	1. System administrators review the functionality of algorithms. 2. Updates and maintenance are performed as needed.
	Alternate Flows: Algorithm Update Failure: If the system encounters issues in updating algorithms, administrators are notified, and the previous version remains in use.
Post Condition:	Algorithms are maintained and updated for optimal performance.

Use Case Number:	UC-10
Use Case Name:	Database Management
Overview:	System administrators can manage the SQL database storing pre-computed parameters. This includes ensuring data integrity, security, and efficient retrieval of stored parameters.
Actors:	System Administrator, System
Pre condition:	The system is operational.
Flow:	1. System administrators manage the SQL database, ensuring data integrity and security. 2. Efficient retrieval of stored parameters is ensured.
	Alternate Flows:Security Implementation Failure: If necessary, system administrators perform maintenance tasks on the database.

Post Condition:	Database management tasks are successfully performed.