Low-Level Design Document for Processing the Images

1. System Overview

This system efficiently processes image data from CSV files. It provides an asynchronous pipeline to:

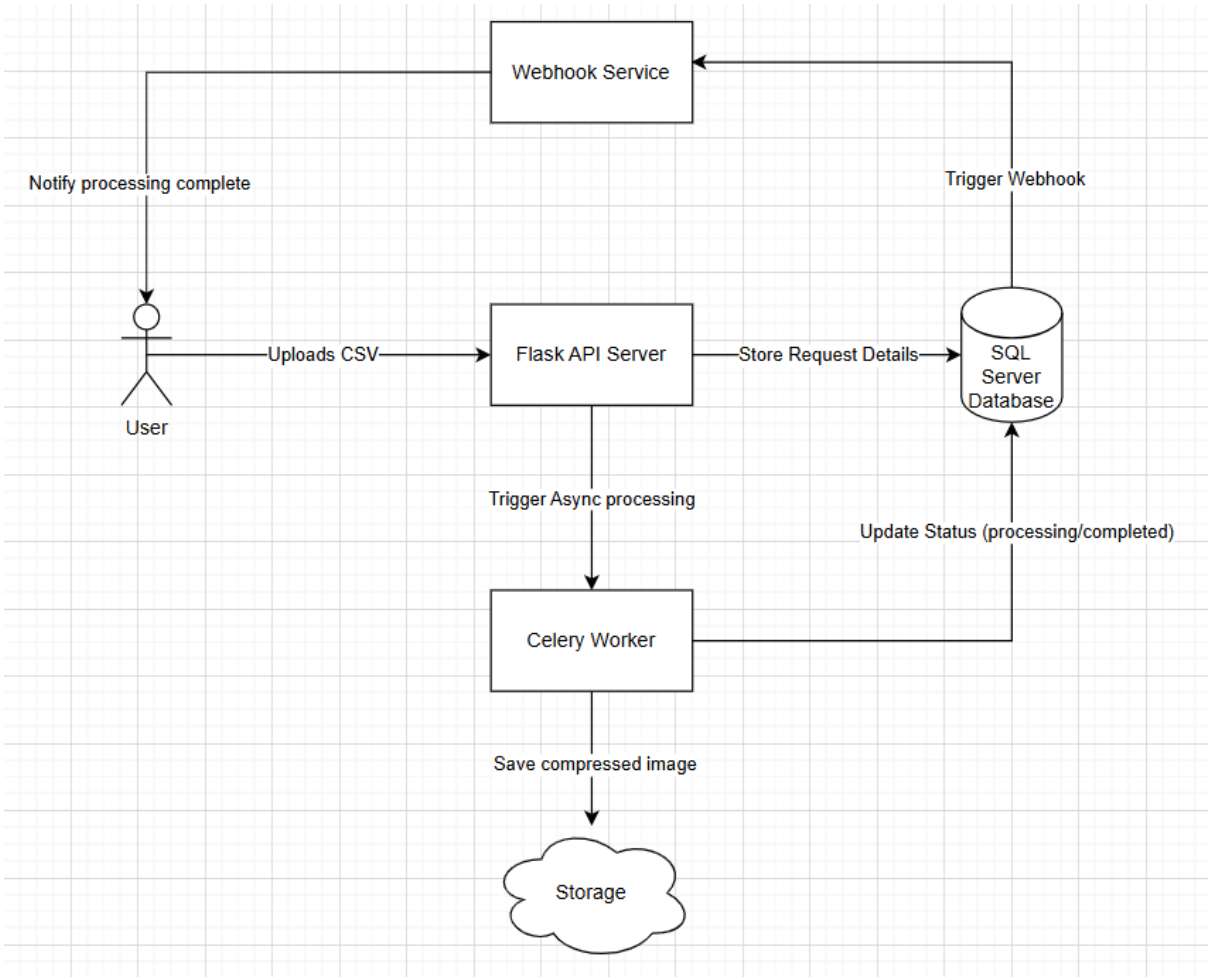
- Accept CSV files
- Validate data
- Compress images by 50%
- Store processed images in a database
- Provide APIs for upload, status checks, and webhook notifications

2. System Architecture

Components

1. **Flask API Server**: Handles CSV upload, status checks, and webhook notifications.
2. **Celery Worker**: Processes images asynchronously.
3. **SQL Server Database**: Stores request and image data.
4. **Storage (Local / Cloud)**: Saves compressed images.
5. **Webhook Service**: Notifies users upon completion.

Flow Diagram :



3. Database Schema

Requests Table

Column Name	Type	Description
request_id	UUID	Unique ID for each request
status	VARCHAR	Pending, Processing, Completed
timestamp	DATETIME	Request timestamp

Products Table

Column Name	Type	Description
product_id	UUID	Unique ID for each product
request_id	UUID	Foreign key linking to Requests
product_name	VARCHAR	Product name from CSV

Images Table

Column Name	Type	Description
image_id	UUID	Unique ID for each image
product_id	UUID	Foreign key linking to Products
input_image_url	TEXT	Original image URL
output_image_url	TEXT	Compressed image URL
status	VARCHAR	Processing / Completed

Processing Logs Table

Column Name	Type	Description
log_id	UUID	Unique log entry
request_id	UUID	Foreign key linking to Requests
image_id	UUID	Foreign key linking to Images
step	VARCHAR	Step in the processing pipeline
status	VARCHAR	Success/Failure
timestamp	DATETIME	Log timestamp

4. API Endpoints

1. Upload API (POST /upload)

- Accepts CSV file
- Validates format
- Returns request_id

2. Status API (GET /status/{request_id})

- Checks processing status
- Returns Pending, Processing, or Completed

3. Webhook API (POST /webhook)

- Triggered when processing is complete
- Sends notification with request details

5. Asynchronous Image Processing

- **Task Queue**: Celery in Python
- **Compression**: Python Imaging Library
- **Storage Options**: Local Storage