Step 1: Requirements

Functional requirements:

- Users can upload/download/view photos
- Users can perform searches based on photo/video titles
- Users can follow other users
- System can generate and display a user's news feed consisting of top photos from follows

Non-functional requirements:

- Highly available
- Acceptable latency of the system is 200ms for news feed generation
- Consistency can take a hit
- System should be highly reliable; photos or videos should never be lost

Step 2: Back of Envelope Calculations

- 500M total users, 1M daily active users
- 2M new photos every day, 23 new photos per second
- Average photo size = 200KB
- 2M*200KB=400GB per day for photos → 1425 TB per 10 years

Step 3: System Interface Definition

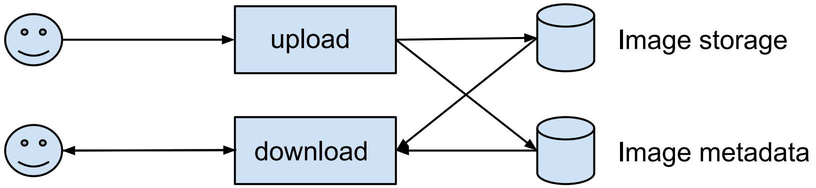
Skipped

Step 4: Define Data Model

- Photo with PK PhotoId(int): UserID, PhotoPath, Photo/User Latitude/Longitude, CreationDate
- User with PK UserID(int): Name, Email, DateOfBirth, CreationDate, LastLogin
- UserFollow with PK UserID1(int) and UserID2(int)
- Store photos in distributed file storage e.g., S3
- For Photo and User tables, we can use a wide-column store e.g., Cassandra

Step 5: High-Level Design

- Uploads can be much slower than reads, so separate upload and reads


upload
Image storage
download
Image metadata