|  |  |
| --- | --- |
| https://avatars3.githubusercontent.com/u/37767905?s=460&v=4 | FotoLapse  Functional Spec  Authors: Chrohm |

**Spec Status:** [Draft, In Review, Accepted]

*This document is intended to provide focused details on the scenario and provide key indicators to what the scenario is expected to deliver and how it will be measured. It is intended to be quick to provide a minimal set of information for the project to be successful.*

# Overview

## Elevator Pitch / Narrative

Customers take photos to capture memories and measure the passage of time. A popular way to consume these photos as a medium is to assemble them into a time lapse. Currently, assembling a stabilized time lapse of a subject is manually done in a time intensive manner. Customers would like an easy and comprehensive experience for creating their time lapses that would deliver high quality time-lapse media in a timely manner.

## Customers

Amateur and professional photographers and visual story tellers.

## Problem Statement and Supporting Customer Insights

Cloud based FotoLapse generator. This was a problem proposed for a capstone class.

## Existing Solutions or Expectations

Currently there exist many third party apps for creating time lapses. However, they are not cloud based systems with promised security or privacy. Also, the quality of the time lapses, although not assessed, is probably basic and rely on a single truth input source (exactly 1 camera placed in a static location over the course of photography time frame).

## Goals

* Processing unit for assembling the gif or video
* Stabilization and normalizing photos for a photo-lapse
* Options for customizing photo-lapse (length of animation and output size
* Optimizations for lapse-assembly
* Web app for UI elements/ API contracts
* API: Upload image
* API: Download animation
* API: Recent image retrieval for location
* UI: Specify image metadata
* Backend photo storage and organization based on location and time range

## Non-Goals

* API: Account creation
* API: Account authentication
* API: Privacy restrictions on uploading/viewing/modifying photo stores
* UI: Location collection privacy
* Phone apps that delay upload until connection available