

# <Name of Project> Lab Notebook

Note to the reader: At [LightStep](#), we use Lab Notebooks as a way to document and communicate about the work we do on the engineering team, and especially work that happens outside of the code base. You can read more about the process in [this blog post](#). Let us know if you find it useful too [@LightStepHQ!](#)

## Commander's Intent

- Encourage incremental approaches to project design and implementation
- Include stakeholders and experts from across the company in the discussion without a rigid sign-off process
- Support retrospectives and postmortems by recording important decisions for later

A successful project lab notebook accomplishes these goals by:

1. Recording questions, hypotheses, assumptions, experiments, and important decisions related to the project *as they occur*, as well as the results of these experiments and decisions
2. Serving as a log of interactions with – and future questions for – people that are not members of the project
3. Acting as a resource for future team members (as well as our future selves) to help us recall and understand why we built the things we did

A lab notebook replaces an engineering design doc (as well as possibly other docs). A lab notebook is *not* meant to serve as documentation of the end result of a project (for example, product documentation, oncall training).

How to use this document:

1. Make a copy, update the title and product brief link, and put it in the project folder.
2. Read the blue text if you haven't done so lately.
3. Delete the blue text. You can always go back and read it in the [template](#).
4. Fill in the [Current Design and Implementation](#) with a 1-2 page summary of the current design.
  - a. This can change in fidelity over time but should ~always be up to date.
5. Use the [format below](#) to add new log entries. They are a great way to prepare for meetings as well as to capture any discussion or decisions that are made during meetings.
  - a. Annotate old entries as you learn about the consequences of those decisions.

## Suggestions

Focus on issues related to *what* is being built and include the reasons *why* a particular decision was made or approach was taken. Avoid duplicating information that's found in a ticket or other documents (including, for example, sprint planning and work assignment).

The list below provides some guidance on the kinds of issues that should be considered in this doc; this list is certainly not exhaustive. Topics can be revisited many times over the course of the project.

- Project scope – goals and non-goals not explicit in the product brief
- Previous related work
- Technical architecture – dependencies on existing components, services, and data stores
- Information architecture
- Schema and durable storage
- Testing
- APIs/UI/UX
- User impact – required application or satellite changes, backward incompatible changes to APIs, reduced functionality
- Internal impact – changes to how we design, develop, or support our product
- Critical milestones
- Resources and cost – rough estimate of how this project will impact COGS
- Security and compliance – new data that we will be ingesting, storing, or transmitting
- Qualification
- Monitoring
- Rollout and rollback plans
- Future work

Some templates have been partially filled in below as suggestions for early consideration. These need not all be done at the very beginning of the project, but all of them should be done at some point and should be done before any significant work that depends on them.

## **Current Design and Implementation**

Last updated: <date>

Describe the *current* design (technical, product, etc.) of the project in 1-2 pages. Include what an engineer, product manager, or designer who's joining the project *today* would need to know *today*. This need not be comprehensive; think of this as a FAQ. Some suggestions:

- Refer to the product brief when appropriate
- Include a link to the current mocks
- Take a picture of a whiteboard
- Link to a .proto or other file that defines and documents an API

The scope of what's included in this section can shift over time. For example, early in the project sketching out data structures might be important information that's not captured elsewhere; later this information can simply be linked to.

Keep this up-to-date on a ~weekly basis so that new team members can use it effectively and to provide context for other readers of this document.

## Log

*Thursday, 8/1/19 - @michelle - Resources and cost*

After simulating the expected load on our new service, *Fuzzy Carrot*, we estimate that we will need to run 20 instances of this service, each with 3 CPUs and 10GB memory.

Action Items:

- @michelle: ensure that we have capacity in our Kubernetes cluster for this new service. If not, work with @sarah from the infrastructure team to increase the size of our cluster in preparation for release.

*Wednesday, 7/31/19 - @alice - Data retention discussion*

As we begin to ingest *foo* data, we need to make sure that it is subject to a data retention policy mirroring that of the other forms of data in the system. @joe from the infrastructure team suggested that it should be relatively easy to use the existing infrastructure as long as we store data in a bucket structure compliant with the existing system.

Action Items:

- @matt: investigate the bucket structure of our *bar* data and create a Lab Notebook entry with a proposal for how we could mimic it using our *foo* data

## Template

<Date> - <Authors> - <Topic(s)>  
<Questions/Discussion/Decisions>