Project Document

This document is meant to help you

* Document ideas for your Project
* Record your weekly progress
* Reflect on the process and experience

The audience of this doc is people who want to read more about your project. You can assume they understand the basic project you started with, but don’t assume they’ll have access to your project or will be able to talk to you personally. Include everything somebody needs to know to understand your project and how you will make it happen.

When implementing a new feature, Google engineers write design docs similar to this one, for the same reasons: to organize their plan, to break down the work, to record what they did, and to document the final product.

Part 1: What do you want to do?

What is the entire list of user groups your team has considered?

There are no bad ideas during brainstorming. List them all here.

What user group/idea has your team eventually decided on?

Describe the problem that your project will address in a few sentences. What’s your [elevator pitch](https://en.wikipedia.org/wiki/Elevator_pitch)? Are you close to a point where you can create a user statement?

What will the final user experience be?

From a user perspective, what will change to your app between now and Week 10? What will the user be able to do? How will people use your feature? Include any mocks (sketches showing how your feature will work) you come up with here.

What technical tools + skills are required to realize your vision?

From a technical perspective, what will you change to your app between now and Week 10? What tools/APIs will you have to learn? What code changes are necessary? What data models will you need? Be specific.

What smaller pieces can you break your project down into?

Try to break your idea down into smaller steps. What is the “hello world” version of your idea? What is the [minimum viable product](https://en.wikipedia.org/wiki/Minimum_viable_product) version of your feature? What is your stretch goal?

Who will do what?

Be specific, and make sure that everybody agrees on the plan before the end of your meeting. It’s okay if the plan shifts after you get started, but at this point you should have a roadmap. Keep this section updated, if responsibilities or timeline change, record those changes here.

When will it get done?

Try to set milestones: what should you have done in one week? Two weeks? Keep in mind that completing a phase might include additional steps such as testing and code reviews. Try to factor in your timeline events such as holidays or exams that might slow you down.

Part 2: What are you doing?

Fill this part out as you go. Keep a weekly log of everything that each person does. Link to pull requests. How did the plan change? What ended up being easier or harder than you expected? Where did you get stuck? What were you surprised by?

Part 3: How did it go?

Have each person fill this part out after you’re done.

How did the project go? What worked well? What didn’t work well? How did the demo week go? Did users uncover any bugs or strange behavior? It’s okay if you didn’t finish everything you planned! What would you do differently? What would you do if you had more time?