

App Development Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that teaches your classmates about any topic you both find interesting. Along the way you'll learn how to use many of the features of App Lab as well as skills that will help you when building more apps throughout this class.

You will submit

- Your final app
- This completed project-planning guide

App Requirements

- Uses at least three screens
- Includes examples of images, audio, and text
- A clear and easy to navigate user interface
- Clearly communicates information about your topic
- Code is cleanly written and free of errors

Steps

- Collaborate with your partner to pick a topic you are both interested in
- Interview classmates to identify what they already know about the topic
- Design your app's user interface using this planning guide
- Design and program your app in App Lab
- Collect feedback from your classmates and update your app
- Share your final app with the class

Investigate and Reflect Phase

Step 1. Brainstorm Topic Ideas: Your app can teach your classmates about any topic you and your partner agree on. Your topics could be a hobby, something you've always been interested in, a piece of your personal history, or just something you think your classmates should learn more about.

Write down three ideas for a topic that you brainstorm individually.

Idea 1:

Idea 2:

Idea 3:

Step 2. Choose One Topic: Now talk through your ideas with your partner. Together pick a topic both of you are interested in teaching your classmates about. Explain in a few sentences what would be covered. For example, if your topic is basketball, you would write a few sentences explaining that you would cover the rules and the origin of the sport.

Our Topic:

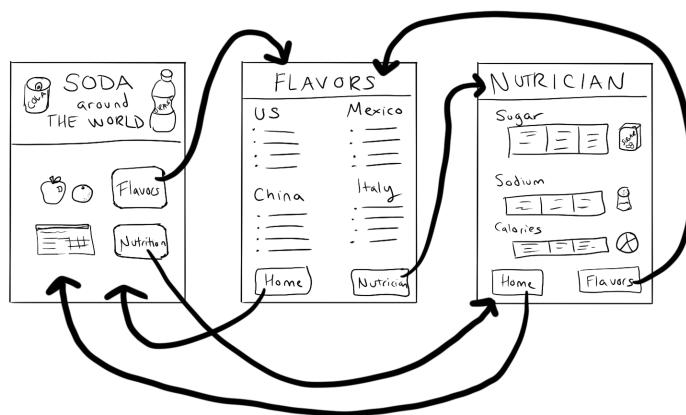
Step 3. Interview Your Classmates: To design your app you'll need to understand your users. For this project your user is your classmates, and you'll need to understand what they already know about your topic.

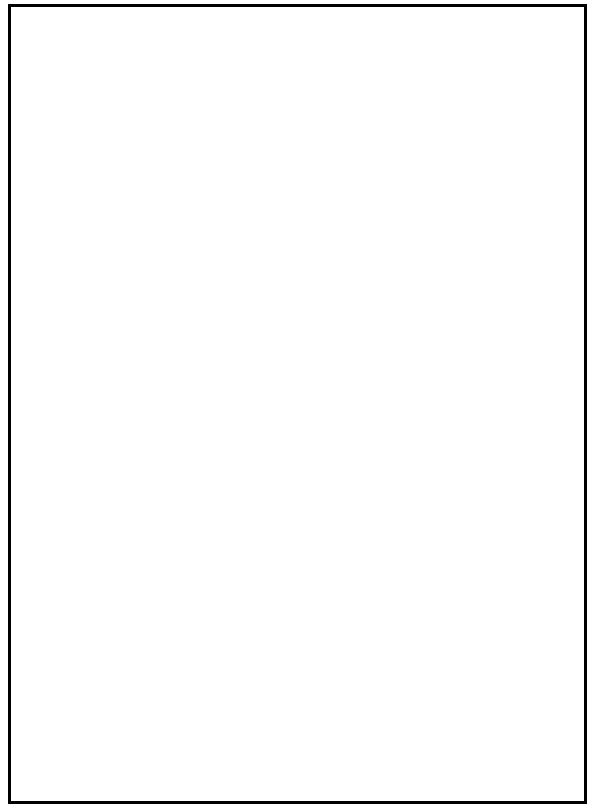
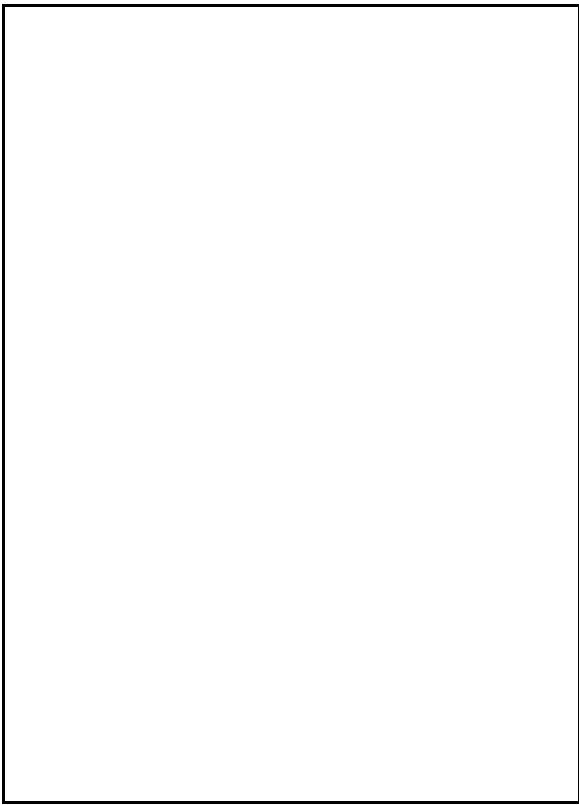
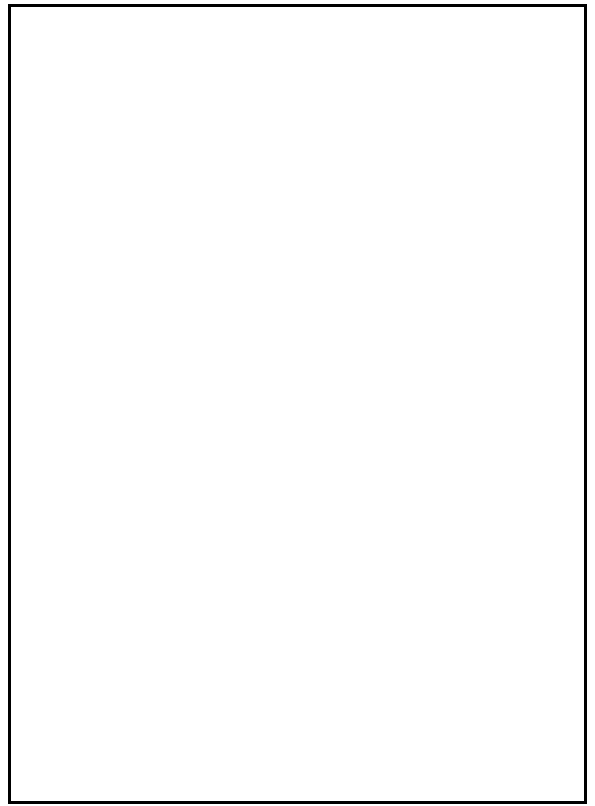
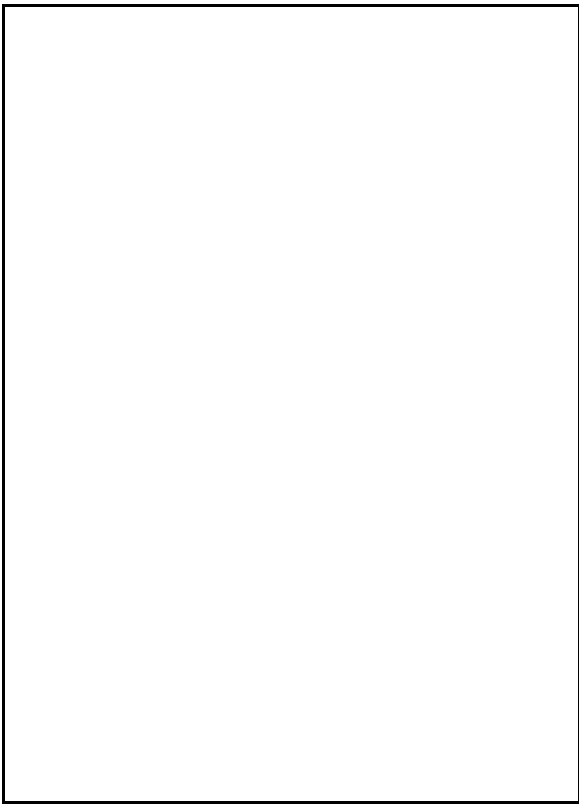
Find two classmates and talk to them about your topic for a couple minutes. Then fill in this table

Name	What do they already know about your topic?	What do they need or want to learn about your topic?

Design Phase

Step 4. Create a Program Specification: Based on your research you identified **requirements** for what your app must teach your classmates. On the next page you should draw a **specification** that shows how your app will actually run to meet those requirements. This means you should include all the buttons, text, and images that the user will be able to use. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.





Building Phase

Step 5. Start Building Your App: Work with your partner to build your app. Along the way make sure you:

- Use the program specification you drew as a starting point, but it's OK to update as you go.
- Use pair programming
- Use your debugging skills to check that your app is working

Before you begin to code, fill out the chart below for any Event Handlers in your program:

Element ID	Action	What happens?
"dogButton"	"click"	<i>A picture of a dog appears The background of the screen changes to green</i>

--	--	--

Use the chart to guide you in adding programming statements to your program.

Testing Phase

Step 6. Testing & Feedback: You will need to test your app to make sure it works as expected. To do that find at least two classmates to use your app.

- Ask students to read through your program specification and requirements
- Ask them to use the app and test out the different behaviors included in your specification
- Write down anything you noticed them finding confusing or broken
- Ask them to share anything they recommend improving

Name	Things that could be improved based on watching them use the app	Improvements this person recommends

Step 7. Pick Improvements: Pick at least one improvement you plan to make to your app based on feedback you collected from your classmate.

Improvement 1:

Improvement 2 (Optional):

Step 8: Complete Your App: Finish your app!

Reflection

Question 1: Provide a written response that:

- describes the overall purpose of the program
- describes the functionality of your app
- describes the input and outputs of your app

(Approx 150 words)

Question 2: This project was created using a development process that required you to incorporate the ideas of your partner and feedback from your classmates. Provide a written response that describes one part of your app that was improved through input from EITHER your partner or feedback you received from classmates. Include:

- Who specifically provided the idea or recommendation
- What their idea or recommendation was
- The specific change you made to your app's user interface or functionality in response to the recommendation
- How you believe this change improved your app

(Approx 150 words)

Rubric

Category	Extensive Evidence	Convincing Evidence	Limited Evidence	No Evidence
User Interface Screens	User interface includes at least three screens	User interface includes two screens.	User interface is on a single screen.	The screen is blank.
User Interface Navigation	The user can easily navigate between all screens.	The user can easily navigate between most screens.	The user can easily navigate between some screens.	The user cannot navigate between screens.
User Interface Elements	The app includes at least one example each of: <ul style="list-style-type: none"> - Text - Image - and audio. 	The app includes at least one example of two of the following: <ul style="list-style-type: none"> - Text - Image - And audio 	The app includes at least one example of one of the following: <ul style="list-style-type: none"> - Text - Image - And audio 	The app includes no text, images, or audio.
Code	Code runs without errors	Code runs with a few errors.	Code does not run or has a lot of errors.	Code is blank.
Element IDs	Screen elements all use meaningful IDs.	Screen elements mostly use meaningful IDs	Some screen elements use meaningful IDs	Screen elements do not use meaningful IDs.
App Topic	Topic is clearly communicated and explained.	Topic is somewhat clearly communicated and explained.	Topic is not communicated well.	App appears to be a random collection of elements with no clear topic.
App Development Planning Guide:	Planning guide is fully completed.	Planning guide is mostly completed.	Planning guide has a few parts completed..	Planning guide is empty.
Written Response 1:	Response accurately describes the purpose, functionality, and inputs/outputs of the app.	Response mostly describes the purpose, functionality, and inputs/outputs of the app.	Response is not complete, but does describe the purpose, functionality, or inputs/outputs of the app.	Response does not address the prompt in any way or is blank.
Written Response 2:.	Response fully describes an idea or recommendation provided by a partner / peer and how it improved the app.	Response mostly describes an idea or recommendation provided by a partner / peer and how it improved the app.	Response is not complete, but does describe some of the work with a partner.	Response does not address the prompt in any way or is blank.