



Hackathon Project Planning Guide

Project Description

For this project you will work with a partner. Together you will create an app that uses a dataset. There are two roles in this project: designer and programmer. You are both responsible for the overall project, but you will focus on different parts of the project at any given time. On the last day of the project, you will individually complete a written response that will help you practice for the Create Performance Task.

You will submit

- Your final app
- This completed project-planning guide
- A written response

App Requirements

- At least three screens
 - All screens can be easily navigated to through the user interface
- A dataset used in a meaningful way towards the programs purpose
- At least one list is traversed using: map, reduce, or filter (indicate which in a comment) in a meaningful way towards the program's purpose
- Programming constructs: variable(s), function(s), conditional(s), list(s), loop(s)
- All functions include comments that explain their purpose and how they work
- All element IDs have meaningful names
- No errors in the code

Steps

- Collaborate with your partner to pick a dataset
- Decide on an app to build which traverses a list from the dataset
- Create a paper prototype
- **Designer:**
 - Create the screens
 - Support the programmer with pair programming as needed
- **Programmer:**
 - Program the app
 - Support the designer as needed.

Investigate Phase

Step 1. Choose a Dataset: Open a project on Code Studio and look through the different datasets available. Choose one that looks interesting to both you and your partner.

Dataset: _____.

What column(s) of data will you use in your app?

-
-
-

Step 2. Brainstorm an App: Consider the columns of data that you are using. How will your app traverse this data? Circle one and explain below.

- **Filter** (most common option): use the list from one column to determine information that will be filtered from a list created by another column
 - Example: dogHeight filters dogNames, so only the names of small dogs are added to the filtered list
- **Map:** Add or change each item in a list
 - Example: map a list of numbers pulled from a column using Math.round - now each number is rounded
- **Reduce:** Reduce the data in a list to a single number
 - Example: find the smallest number in a list

Your app will use (circle):

MAP

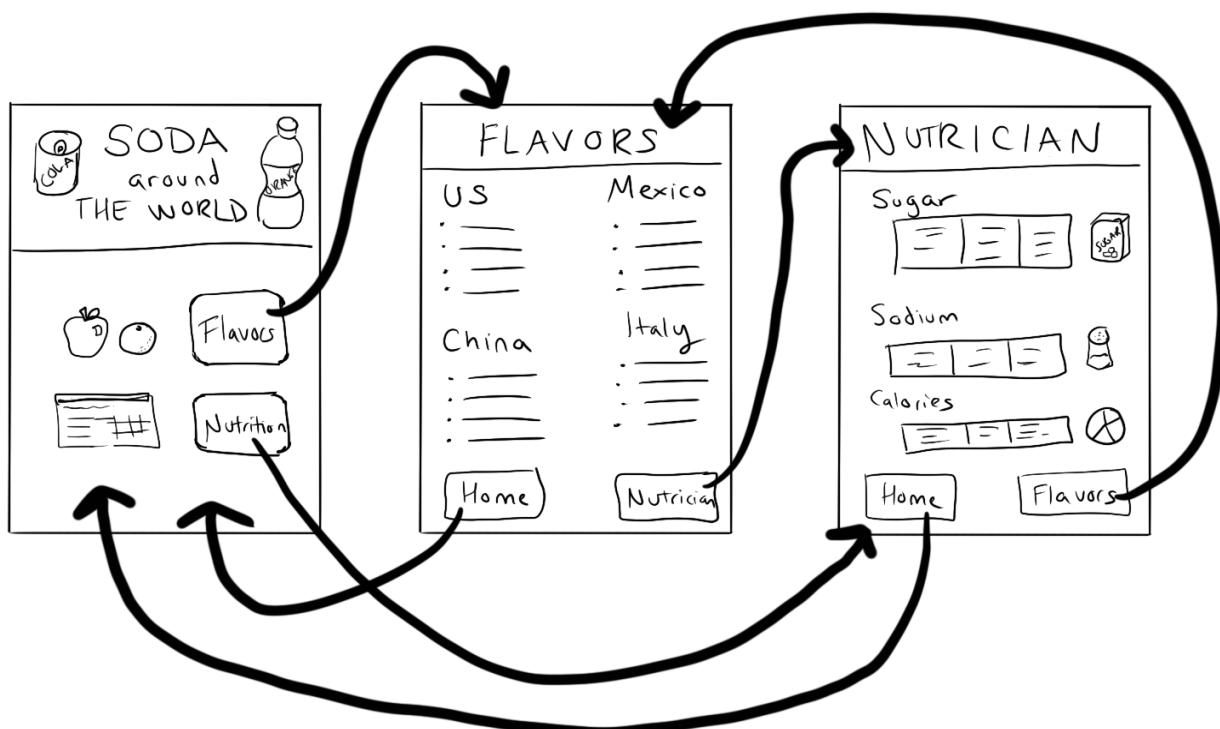
REDUCE

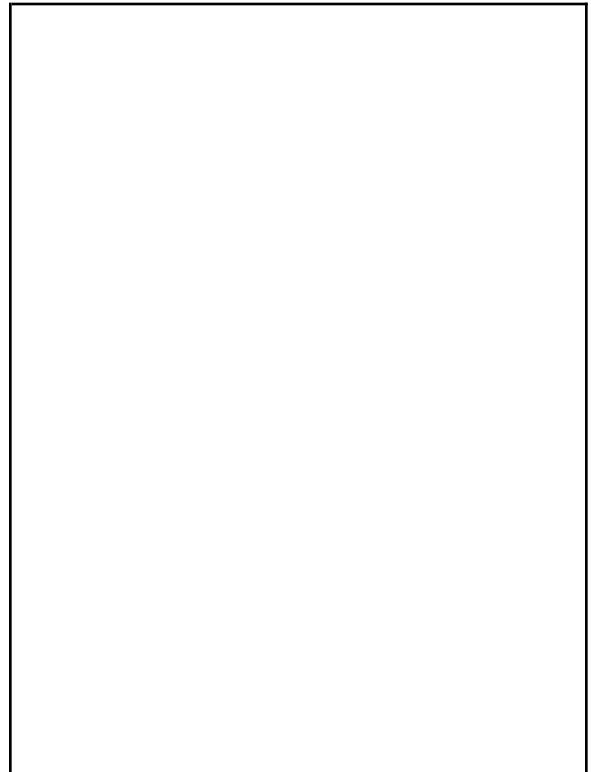
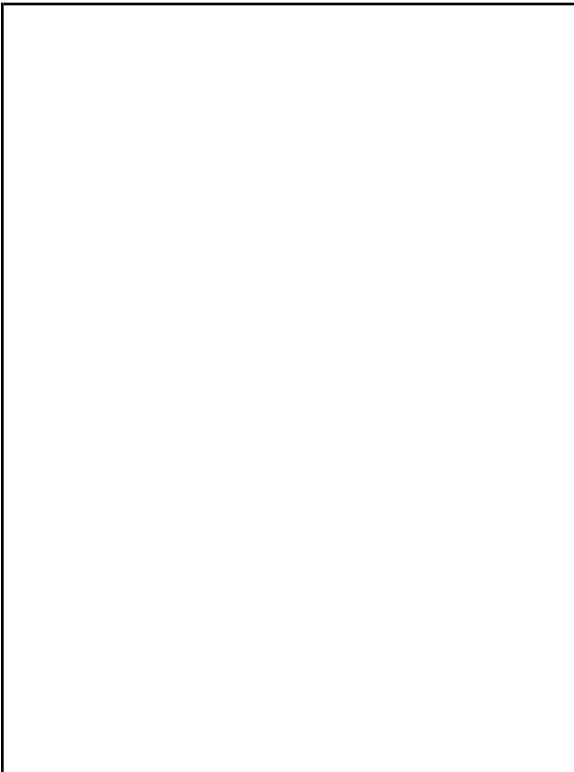
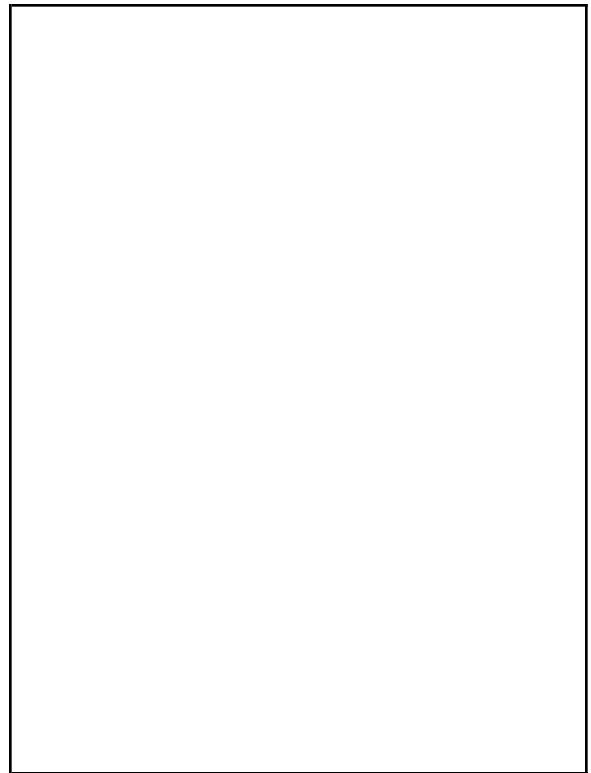
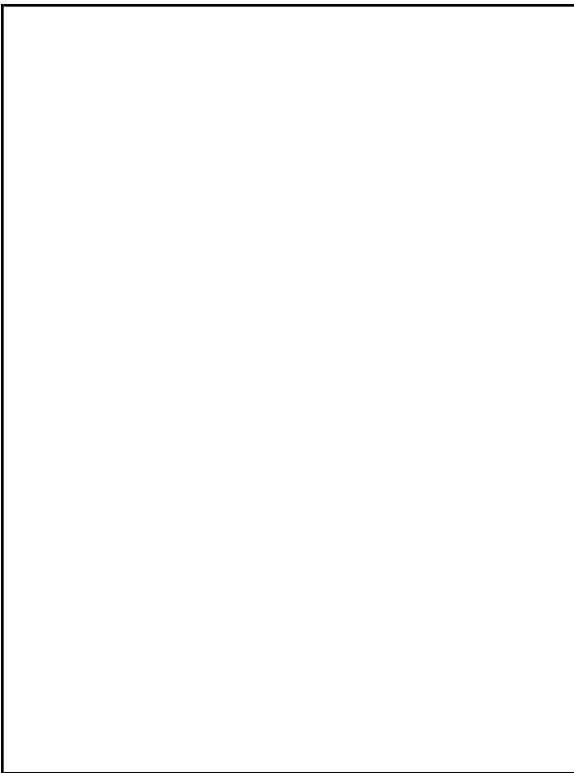
FILTER

Explain in more detail:

Design Phase

Step 3. Create a Paper Prototype: On the next page you should draw a **prototype** that shows how your app will actually run. This means you should include all the buttons, text, and images that the user will be able to use and see. 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.





Step 4. Prepare to Build Your App:

Before you begin to code, fill out the chart below. Add more rows (or use a scrap piece of paper) as needed.

Variable(s)		
Name	Type	What it's storing
dogSize	list	A list of dog sizes pulled from the "Max Height" column
Conditional(s)		
Boolean Expression (pseudocode is ok)	Notes	
<i>if dog size is greater than 15, append to a list</i>	<i>Inside a for loop that traverses the dogSize list</i>	
Loop(s)		
For Loop (pseudocode is ok)	Notes	
<i>for(var i=0; i<dogSize.length; i++)</i>	<i>Traverses dogSize list</i>	

onEvent(s)		
Element ID	Action	What happens?
“dogButton”	“click”	<i>A picture of a dog appears The background of the screen changes to green</i>
function(s)		
Name	Purpose	How it works
updateScreen	<i>updates what appears on the screen after the user selects a dog</i>	<i>Filters the lists and displays the images and names of dogs on the screen.</i>

Build Phase

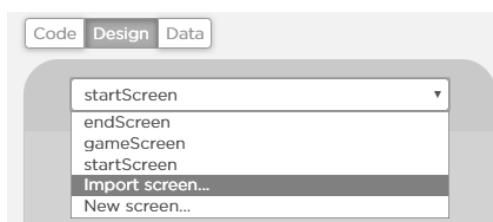
Step 5: Test Screens: Build a quick version of the screens which includes all elements with their proper element IDs. Do not worry about design at all. This is purely to allow the programmer the ability to test their code as they go along.

Step 6: Build

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

Designers: Use the chart to guide you in adding screen elements to your program. You can work on a separate computer from your partner.

- When you have finished screens, your partner should delete the test screens and then import the finished screens.
- Note: If screens are not deleted before importing the project you will get an error message because element IDs will be the same. To avoid this, add a blank screen, delete the old project screens, import the finished screens, and delete the blank screen. Make sure to set your home screen to be the default.



Scoring Guidelines

Written Response (individual) - In the Written Response the student...	
Task	Point
Describes the purpose of the app	/1
Includes a program segment that shows how the initial data has been stored in a list	/1
Includes a program segment that shows how the data in this list is being processed	/1
Identifies the name of the list being processed in this response	/1
Identifies what the data contained in this list is representing in the program	/1
Explains how the list manages complexity by explaining how this program code would be written differently without using this list.	/1
Includes a program segment that shows a function definition	/1
Describes what the function does and how it contributes to the program overall	/1
Overall Project (both partners)	
Program uses a dataset in a meaningful way towards the program's purpose	/1
At least one list is traversed using: map, reduce, or filter (indicate which in comment) in a meaningful way towards the program's purpose	/1
Program contains the following programming constructs that are used in meaningful ways towards the program's purpose	
• variable(s)	/1
• function(s)	/1
• conditional(s)	/1
• list(s)	/1
• loop(s)	/1
The app has at least three different screens	
• screen #1	/1
• screen #2	/1
• screen #3	/1
All screens can be easily navigated to through the user interface	/1
All functions include comments that explain purpose and how they work.	/1
No errors in the code	/1
All element IDs have meaningful names	/1
Total	/22