

# Project Plan

The following is the plan and some goals we set up according to the evaluation metrics. Our plan is to divide tasks so we can work in parallel to enable the different modes of image recognition defined.

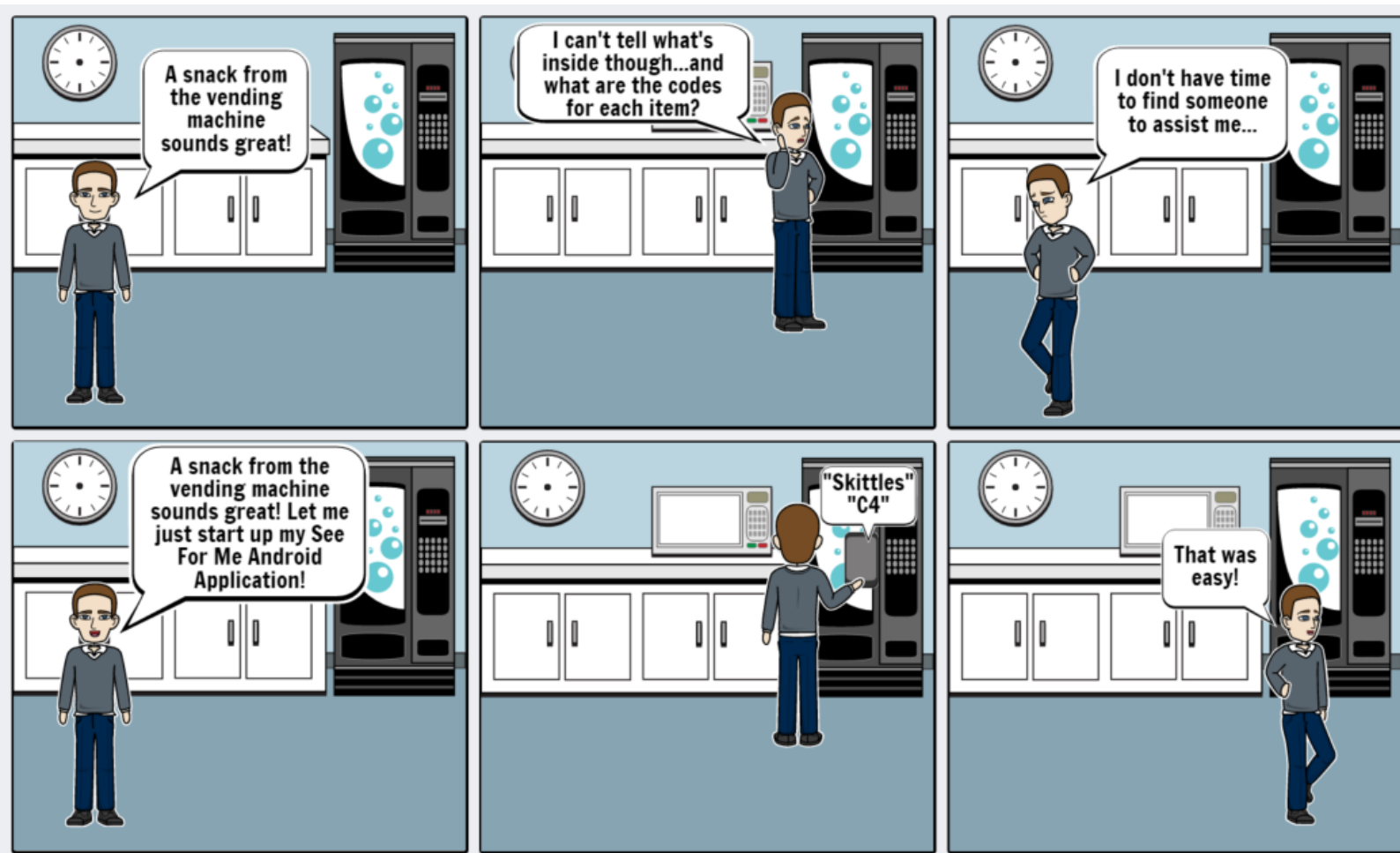
Right now we are able to make API calls to the Google Cloud Vision API and we are able to do text recognition, and object recognition. For the next step, Jane will be responsible for improve the accuracy and efficiency for text recognition. One of our goals is to reduce the time need for end users find the correct orientation, as the API cannot identify text printed vertically or upside down and straightness affects the quality of the OCR. In the evaluation, Jacob said he cannot always find the correct orientation of text that lacks large headers or other features that indicate correct orientation, so to account for this and the fact that it may not always be possible to take level images, we have this goal to make it easier to take an image that the OCR is able to run on. Another goal is to reduce the time needed for him to match related text, (for example, an item on a menu and its price). Our plan is to rearrange the text so that they appear in the correct order so that Jacob does not need to find the correct orientation by himself. Justin will be responsible for object recognition. Object recognition is able to show multiple labels, scoring multiple potential objects from the image taken. We have to determine how to decide how many labels to extract from a single image, and how the user may be able to indicate this. Keith will be responsible for color recognition. While a form of color recognition has been enabled, we have yet been able to create the user interface that translates an RGB value into an easily recognizable color name, and have to run user tests to determine what color ranges are necessary or desired. Justin will be responsible for object recognition. If we have time, we also want to combine text and object recognition if it is possible, so the user can remove the step of indicating whether they want text or object recognition, and reduce it to the single step of taking a photo.

When we are able accomplish the modes of image recognition, we will put greater emphasis on refining and simplifying the process for the user. This first step is confirming that we are able to meet the core functionality, which involves being able to make the API calls and receive a reasonable response. From here we will be able to test multiple iterations of how to simplify the input process, and ensuring that the input and output synchronizes well with Talkback. Justin will be responsible for this part. We estimate we will be able to accomplish these initial tasks within the next couple weeks, and then emphasis can be placed on user testing and design iteration.

For further details on the project plan, see the development path.

# Storyboards

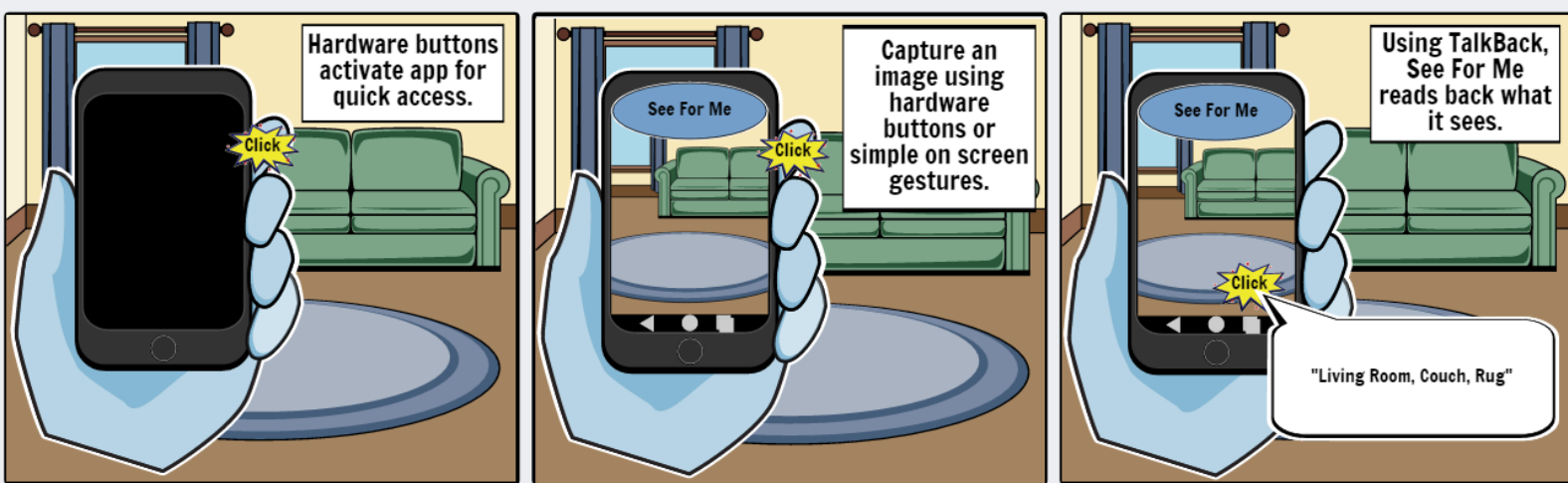
## Scenario 1: Text recognition



## Scenario 2: Color/object recognition



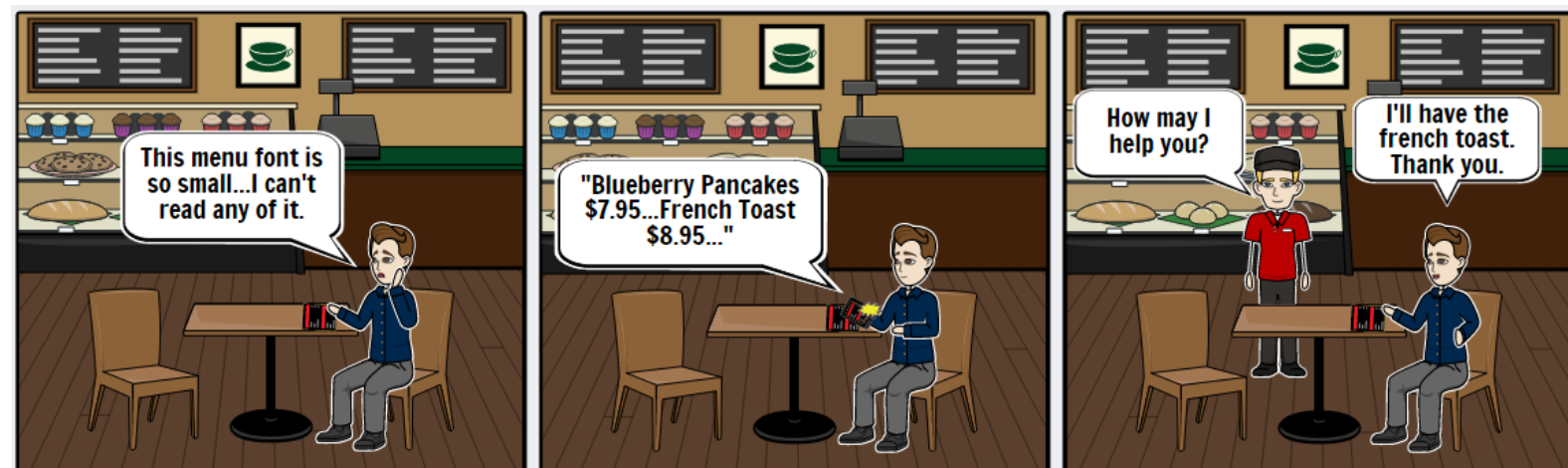
### Scenario 3: object recognition



### Scenario 4: Color recognition



### Scenario 5: Text recognition (OCR)



# Development Path

## 1. Camera:

Since we already set the camera up, right now we can focus on how to minimize the input: It would be pretty ideal if no buttons need to be used to activate the camera: We can overwrite the volume button function: for example, by holding volume down button for a while to take a picture. (Here is an example of listen to volume button events: <http://stackoverflow.com/questions/9162705/android-how-to-listen-for-volume-button-events>)

If we have to use buttons, the interface need to be nicely designed and tested, so that the buttons are easy to memory and easy to find (Users can always find the button accurately without any help from others) and clearly labeled. Maybe the button can be on the edge of the screen so it is easier for them to find out.

Time estimate: 10 hours

Member who's responsible for this: Keith

Projection: reduce the time need for taking picture to 2 seconds.

## 2. Color recognition

Not sure what to do next since Keith already get this part done very well!

It would be nice if the app is able to detect all substantial colors of any object with any type of surface in normal lighting (using a threshold, perhaps? how many pixels is "substantial" here). If there is more than one color in the picture, can make a list of all colors in certain order if possible.

Consistency: the result of multiple color detection of a same object should be the same (maybe 9 out of 10?)

Time estimate: N/A

Member who's responsible for this: Keith has already accomplished much of this

## 3. Text Recognition

We need to add a new text recognition button on the first page and use it to activate the camera. We can probably use the same camera for color recognition and only need to make the following changes:

[https://groups.google.com/forum/?utm\\_medium=email&utm\\_source=footer#!searchin/cloud-vision-trusted-testers/locale/cloud-vision-trusted-testers/mLAKyEEK2nk/Pzu2XojzCgAJ](https://groups.google.com/forum/?utm_medium=email&utm_source=footer#!searchin/cloud-vision-trusted-testers/locale/cloud-vision-trusted-testers/mLAKyEEK2nk/Pzu2XojzCgAJ)

Convert the final image to binary code use “base 64”, then make API call with the content of the output file.

<http://developer.android.com/reference/android/util/Base64.html>

Parse the response Json file to get the Entity textual description, .

([http://www.tutorialspoint.com/android/android\\_json\\_parser.htm](http://www.tutorialspoint.com/android/android_json_parser.htm))

All sentences in the description are separated by new line characters “\n”, we need to parse it so that Talkback can work well.

Display the parsed text on the screen using “Toast”, then Talkback will read it out-loud. (<http://developer.android.com/guide/topics/ui/notifiers/toasts.html>)

The reason that vertical printed text are not detectable is because it tends to set the wrong “locale” value (the language code). Check both the “locale” value and the confidence, if the locale value is not English (“en”) and the confidence is low, it is probably in the wrong orientation. If the locale value is English but the confidence is low, we want to suggest the user take another picture.

Member who’s responsible for this: Jane, John

Time estimate: 1 week

Projection: We aim to reduce the time take to perform text detection by 5 seconds.

#### **4. Object Recognition**

Process is very similar to Text Recognition except flag for object

User press down the object recognition button -> activate the camera -> take a picture -> send to cloud for processing -> extract the content from the Json file it sends back (may send back more than one, how should we decide which one to use)-> display it with toast -> Talkback read it out-loud.

Time estimate: 1 week

Member who’s responsible for this: Justin

Projection: reduce the time need for doing object recognition (such as selecting items from vending machine) from one minute and ten seconds to 30 seconds.

#### **5. Output should layover nicely with Talkback:**

All buttons should be clearly labeled, so that talkback can read it out-loud.

Volume button can be used for both volume control and taking picture (if possible).

Is it possible to interact with Talkback: for example, pause, replay, etc

Time estimate: 1 week: Review, user testing, and other layout and design fixes

Member who’s responsible for this: Justin, John