Sprint 8
(Endgame Edition)

Healthy Belly

Document Updates

- Updated our user manual (we also included a barebones tutorial on the navigation page of the application)
- Installation Guide
- Management Plan
  - Gantt Chart
  - Burndown Chart

Gantt Chart

Healthy Belly Gantt Chart

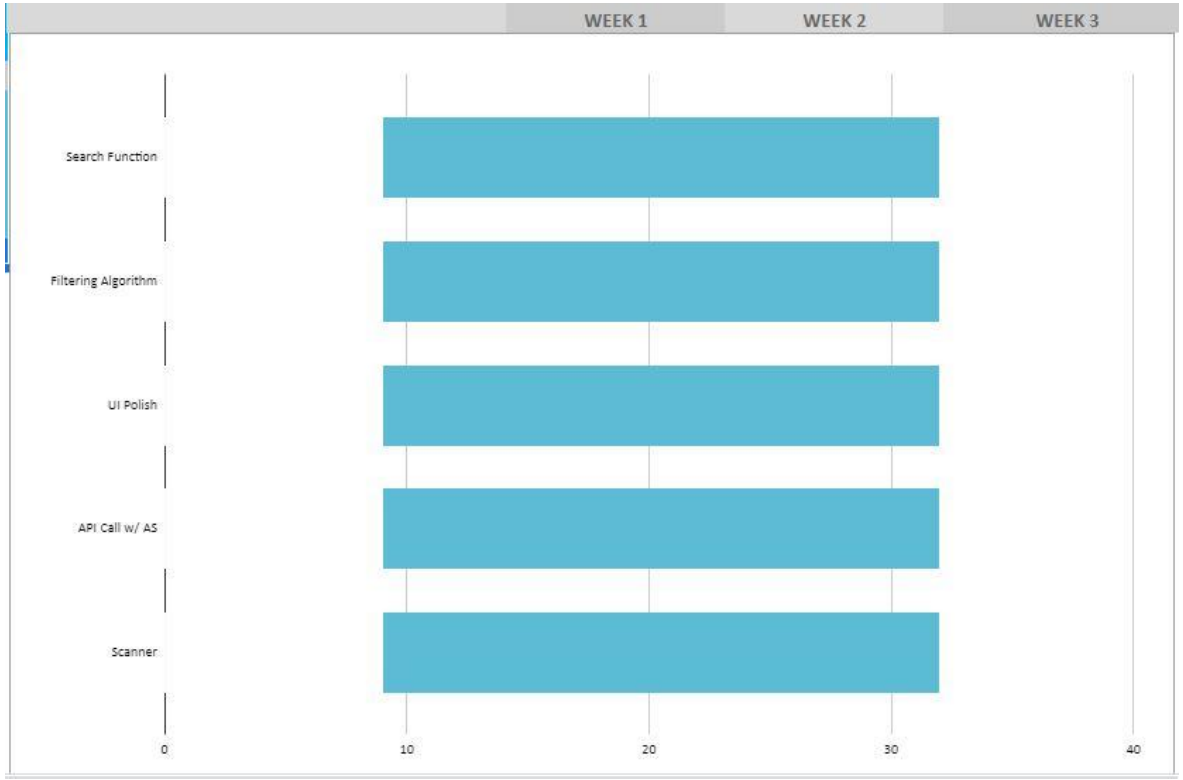
Roster: [M]arty, [Mat]eo, [J]osue, [H]assan

---

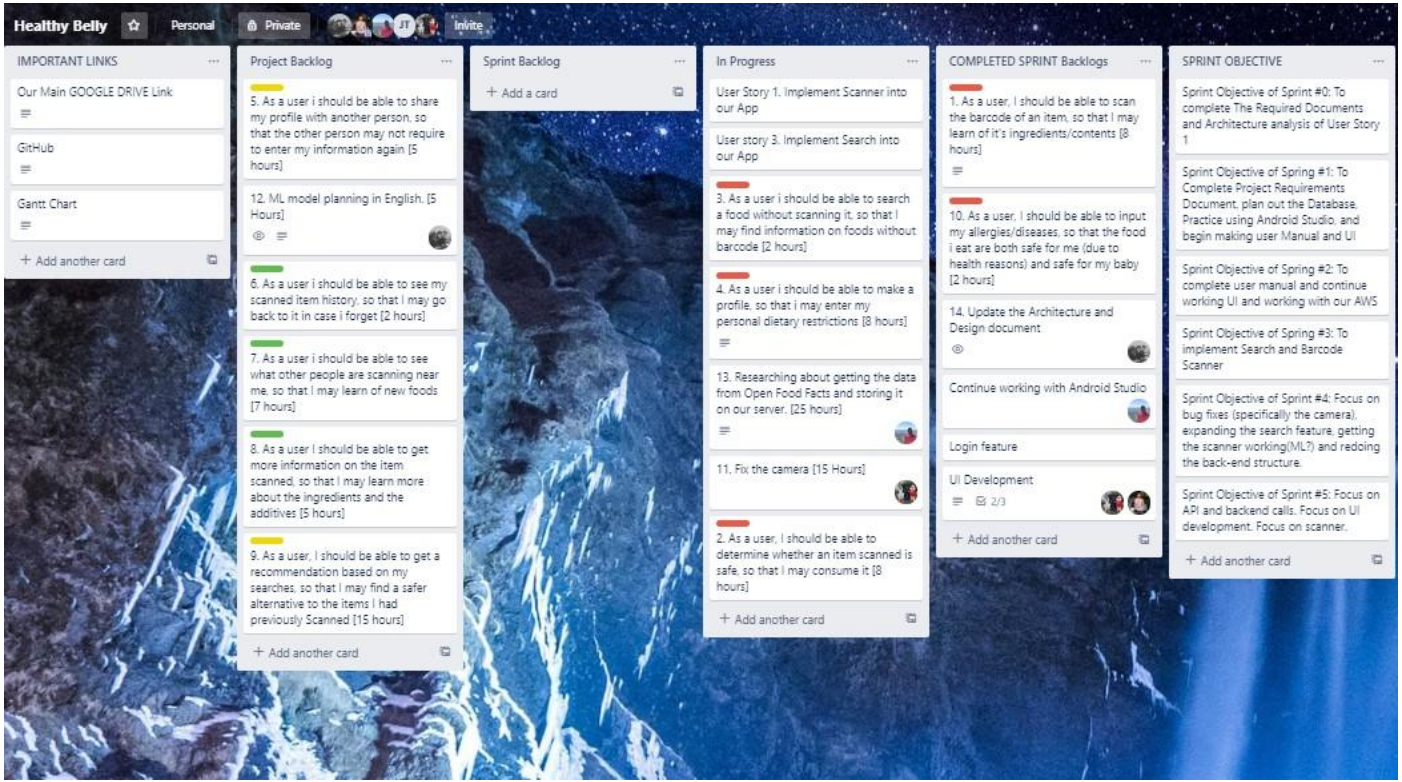
* = an automatically calculated cell

TASK NAME	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
Search Function	4/9	9	5/1	23	23	0	Mat, H	100%
Filtering Algorithm	4/9	9	5/1	23	23	0	Mat, H	100%
UI Polish	4/9	9	5/1	23	23	0	Mat, J	100%
API Call w/ AS	4/9	9	5/1	23	23	0	H	100%
Scanner	4/9	9	5/1	23	23	0	M	100%

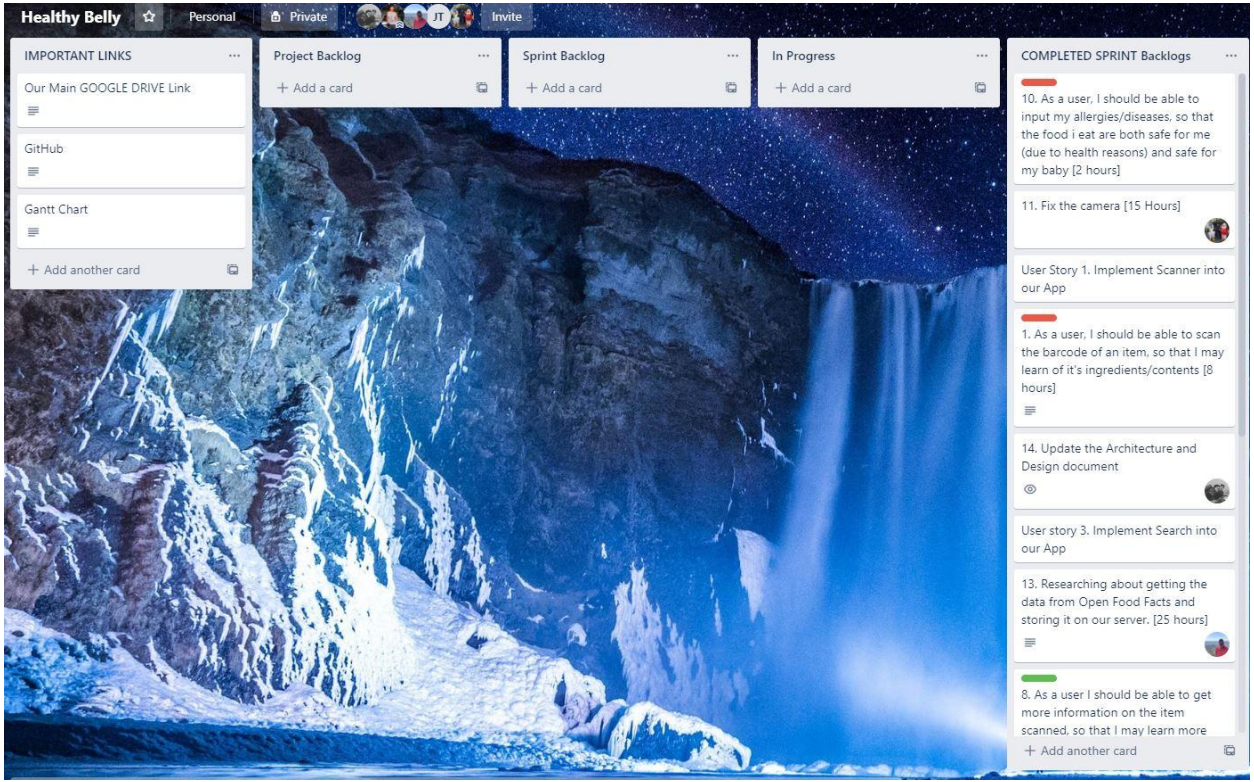
Gantt Chart


WEEK 1
WEEK 2
WEEK 3
Search Function
Filtering Algorithm
UI Polish
API Call w/ AS
Scanner
0
10
20
30
40

Sprint Board Start

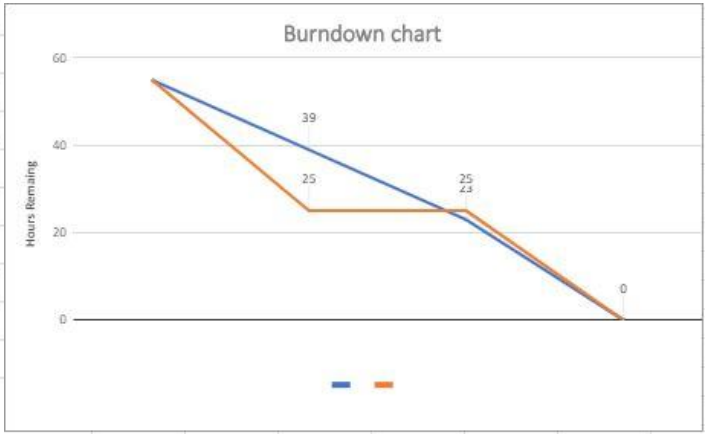


Sprint Board End



Burndown Chart in Theory

Task	Planned Hours	Week 1	Week 2	Week 3	Total Hours
Search Function	15	5	5	5	15
Filtering Algorithm	5	0	5	0	5
UI Polish	5	1	1	3	5
API Call w/ AS	10	5	0	5	10
Scanner Functionality	20	5	5	10	20
			Hours Completed :		55
Actual Hours Remaining	55	39	23	0	
Estimated Remaining Hours	55	25	25	0	



In Practice

- We knew that we had a long road ahead of us, but we also knew we would all be busy with other classes so originally the planned user stories/hours were 55.
- We actually achieved not only that, but we went above and beyond that goal and spent some sleepless nights to get the best product we could out there for you to try.
- We also had to add extra hours a few days ago due to some problems with the API updating just a few days after submitting our code. We will talk about that more later.

User Stories and Epics

- User stories implemented
  - Users can manually input a barcode and get feedback
  - Users can scan an item’s barcode and get feedback
  - Users can create a profile
  - Users can add allergens to their profile
  - Users can edit their allergen list
  - Users can Log into the app
  - Users can see information about the team
  - Users can see their profile
- Epic Stories
  - A User can create a profile unique to themselves
  - Users can scan or type in a barcode that will give them results unique to their profiles

User Stories in the Final Build

- Planned: 8
- Achieved:8

Some of the features that you may remember from last semester had to be scrapped due to the downsizing of our team. We are happy to report however that we were able to complete the features that were the MUST HAVES of our application.

Testing

- We initially planned to do Black Box Testing, but with the news that pregnant people were added to the high risk group of COVID-19 and in general not being able to go observe their interactions with the app we as a team decided against it.
- Most of our testing was in the form of White Box Testing.
  - The way we developed test case was based on what we believed would be the most common things users would do.
  - This was mostly done as the project was being developed so we didn't physical records of the testing.

API and Servers

- We used Firebase for user authentication.
- We used Firebase's Firestore to store the user's allergens, and in the future possibly other information we might need from the user.
- For the API we used OpenFoodFacts.

FireBase + Firestore

- As mentioned before, Firebase is used for user authentication.
- Firestore on the other hand stores all the user's information, such as their username and allergen list. It is set up so that we can add more attributes to the users.

- The biggest issue we faced, which we will get into more later, is that Android Studio's emulators for debugging had problems sending and retrieving information from FireBase and FireStore.

OpenFoodFacts Facts

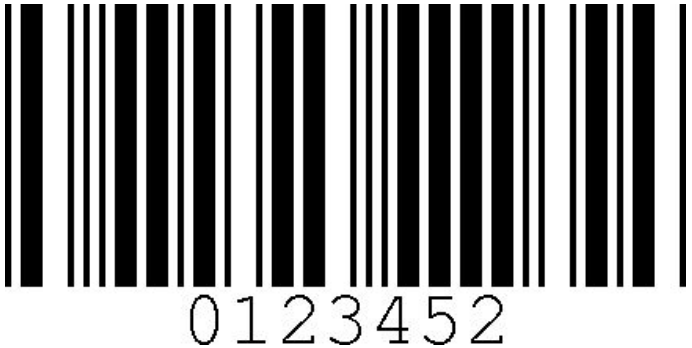
- Utilized OpenFoodFacts API to get information regarding foods that were either Scanned or Searched by manual input
- RESTful API: Json
- Issue:
  - Updated OpenFoodFacts database
  - Returned JSON responses had different keys
- Fix:
  - Changed our parser to work with the updated JSON response

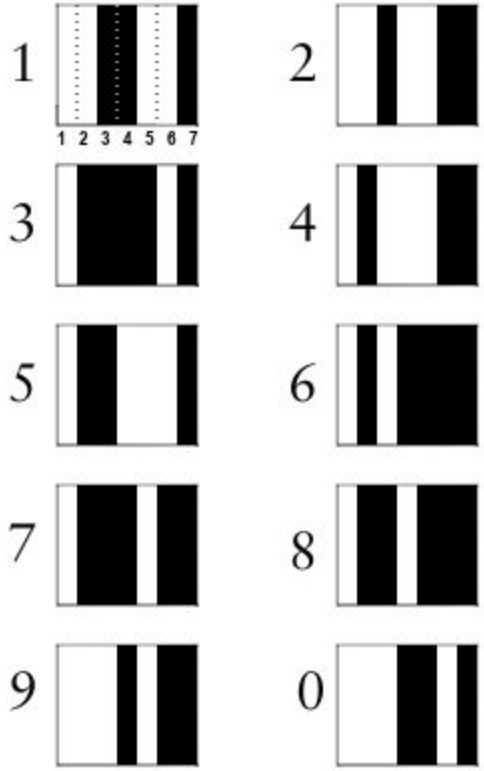
Search

- How does it work?
  - It prompts the user to type in a barcode number
  - Validates input by only letting the user enter a number
  - Uses the barcode number to send a GET request to OpenFoodFacts
- Barcode numbers:
  - When inputting the numbers, users have to input all the numbers, including the first and last number in smaller font
  - Some numbers to test:
    - 052000338768
    - 041900076795

Scanner

- A Quick Explanation On Barcodes
  - Each digit in the product number is given the same amount of horizontal space
  - The first part of a barcode tells you the country where it was issued
  - The next part is the manufacturer of the product
  - The final part of the barcode identifies the product itself





Scanner (Cont.)

- How does the Scanner work?
  - The app activates video preview and captures images continuously
  - Scanner performs some processing as to lower the amount of data and make specifically barcodes stand out better. In the simplest case it will convert the image to grayscale and threshold at a certain level to create a 1-bit-per-pixel (“black-n-white”) image
  - After this the software looks for the outer edges of a barcode
  - Once found, it aligns and frames the Barcode
  - It then breaks down pixels to align with Barcode patterns
  - When that’s done it tries to interpret the barcode and convert it to usually human-readable data

Scanner (Cont.)

- A Shift from Google Vision API to Image Recognition
  - A requirement that was added after Sprint 2
- Third Party Libraries: ZBar and MaterialBarcodeScanner
  - Why these failed?
    - Outdated and Depreciated
    - Incompatible with New Generations of Phones: Specifically Pixels
- What we did:
  - We completed the circle and return to our original plans

Video Demo


m@gmail.com
Clipboard
Password
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
!#1
English (US)

APK

You can find our app, the user manual and the installation guide:

Here: rb.gy/erk0tj

Here: https://drive.google.com/drive/folders/19XayCe1Y9ARG02jUUegAQTqGA8aKzAMU

And Here:

Example of Our Research

- Mulethi
- Licorice
- Bisphenol A (BPA)
- Ephedra
- Angelica
- Kava
- Yohimbe
- Black/Blue Cohosh
- Dong Quai
- Borage Oil
- Pennyroyal
- Mugwort
- Unpasteurized juice
- MSG (Monosodium glutamate)
- Quinoline Yellow (E104)
- Cochineal (E120)
- Indigo Carmine (E132)
- Green S (E142)
- Ponceau 4R (E124)
- Allura Red AC (E129)
- Erythrosine (E127)
- Patent Blue V (E131)
- Tartrazine (E102)
- Phthalates
- Perfluoroalkyl Chemicals
- Perchlorate
- Nitrates/Nitrites
- ... and MANY MORE

Allergen List

- We referenced the Food Allergy Research and Education website for all of our research into what ingredients different allergies were triggered by.
- We compiled a list and went above and beyond and looked into synonyms and different spellings of some of these ingredients to ensure the most accurate list we could given our time.
- The allergies listed in our app are what are commonly known as the Big 8:
    - Milk | Eggs | Peanuts | Tree Nuts | Fish | Shellfish | Wheat | Soy

Sprint Goal And Retrospective

- Sprint Goal:
  - Complete any and all remaining User Stories in the Backlog
  - Have a Functioning Application
  - Test and Debug the Application Consistently at Every Stage
  - Complete Projects and study for Finals for every other Class we are taking
  - As mentioned 55 user stories planned and we went above and beyond that.

- Sprint Retrospective:
  - Prepare for a Summer in Quarantine
  - Continue working on Healthy Belly, fixing bugs and adding more features (For Employment Inquiries, please contact Mr. Bharoocha)
- https://forms.gle/mrdjjrALXaFr1Fzf9

End of the Project Retrospective

- Android Studio was uncharted territory for most of us.
- Many versions of our app were made and scrapped and we had to start from scratch, but now with some new knowledge.
- Many features we had in mind in the inception of the project had to be cut and we are sad that we didn’t get to implement or work on them.
- We are glad that we have something that we believe was significant for the time we and manpower we had to dedicate to this project.
- We taught each other a lot and learned a lot from each other as well.

EndNote

SHOUT OUT TO ALL THE TEAMS FOR COMPLETING THEIR PROJECTS FROM LAST SEMESTER!

Our Journey Together Ends