

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus
Final Deliverable

Project Title:

Baby Feed v12.0

Team Members: Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam, Saily Hernandez

Product Owner(s): Dr. Cristina Palacios

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 *Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document presents the information necessary to gain a good understanding of the Baby Feed project and the work accomplished by the development team this term.

Baby Feed is a web application that provides a means by which parents, clinicians, and researchers can evaluate the nutritional value of an infant or toddler's diet and related health outcomes.

The basis for this evaluation is a score that is generated through the results of questionnaires that record the types and amounts of food that these children consume. This score correlates with certain health markers and allows parents and other concerned parties to determine the quality of their children's diet.

The most recent iteration of Baby Feed (v12.0) now calculates and displays the Diet Quality Index Score (DQIS). It also introduces changes to the User Interface (UI)—such as a loading bar and a home button—as well as numerous bug fixes and improvements to existing systems—such as the growth chart.

Table of Contents

Introduction.....	5
Current System.....	6
Purpose of New System	6
User Stories	7
Implemented User Stories	7
Pending User Stories	7
Project Plan	8
Hardware and Software Resources	8
Sprints Planning.....	9
Sprint 1.....	9
Sprint 2.....	10
Sprint 3.....	10
Sprint 4.....	11
Sprint 5.....	12
System Design	13
Architectural Patterns.....	14
Deployment Diagram.....	14
System Validation.....	15
Appendix.....	16
Appendix A - UML Diagrams	16
Appendix B - User Interface Design.....	24
Appendix C - Sprint Review Reports	25
Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents.....	29
References.....	42

INTRODUCTION

Infants require specialized diets to ensure proper nutrition and healthy development. By following appropriate dietary guidelines, many diseases can be prevented. Establishing good eating habits during the first two years of life is crucial for reducing the risk of chronic diseases like obesity and diabetes in adulthood.

However, accurately calculating infant nutrition can be challenging due to outdated and unreliable methods currently available. There is a lack of tools that incorporate the variety of new infant feeds on the market, posing difficulties for pediatricians and physicians in evaluating an infant's diet.

To address this issue, Dr. Cristina Palacios and her colleagues at the Florida International University's department focused on diabetes and nutrition developed an infant food frequency questionnaire in 2015. This validated questionnaire offers a reliable means of assessing an infant's nutrition accurately. It includes questions about the frequency and quantity of food intake, as well as the types of food consumed. The questionnaire covers a wide range of food groups, such as fruits, vegetables, grains, dairy products, and protein sources. By analyzing the responses, the questionnaire enables the calculation of an infant's nutrient intake and the evaluation of their diet.

Additionally, other tools have been developed to track the nutritional development of children, such as charts and tables that monitor the relationship between weight and height. These tools help ensure healthy development in children.

Current System

The current system utilizes a questionnaire submitted by parents and research participants to monitor the intake of different food types. This information is broken down into components that identify the nutritional profile and diet quality of the child, which is then relayed to users, admins, and clinics using the web application. This allows users to properly monitor the food that is being fed to their infants and helps direct improvements in the children's diets.

Parents also have access to a growth chart where they can input their child's age (in months), their gender, and their measurements (height and weight). Given these inputs, the data is then displayed on a chart that demonstrates the growth of the child and tells the parent if their infant is following a healthy growing pattern or not. A table displaying the history of all the data they've input for their child is presented below the growth chart.

Purpose of New System

The purpose of the new system is to provide site administrators with another bit of information from submitted questionnaires referred to as the Diet Quality Index Score (DQIS). The function of this score is to "grade" the nutritional profile of the results from a given food questionnaire, which is accomplished by considering factors that include the types, amounts, and frequency of foods consumed (among others). Values of these factors are compared against a table (refer to DQIS Table in Appendix D) to determine whether food consumption was adequate for the child.

Once a calculation has taken place, a score is displayed to the admin in a table sorted by food category. The points for each category are then added up and provide a DQIS for each child, with 60 being the max score. The DQIS can be exported to an excel sheet, which also contains all the questionnaire results in the system and effectively displays the same information from website in a compact form.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the Baby Feed project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

The following user stories were successfully implemented into Baby Feed 12.0:

- 1- As a user, I would like to see previous data of my child that I may have saved.
- 2- As a user, I would like to already have an established default child value.
- 3- As a user, I would like it so that I can't enter data above the 98th percentile and below the 2nd percentile.
- 4- As a user, I would like to have a button to go back to the parent portal home on the Growths Chart page.
- 5- As a user, I want to see a loading status wheel on every page.
- 6- As a user, I would like the title of the parent portal home to be changed to "Welcome to the Parent/Caregiver Portal."
- 7- As a user, I want the "Other" tab to be called "Educational Resources from Experts".
- 8- As a user, I would like to be able to change a participant's password.
- 9- As a user, I want to be able to save and add my child's data onto a chart page.
- 10- As a user, I want the breast milk calculations to be fixed so that I can see accurate numbers.
- 11- As a user, I want a new results table called "DQIS" shown only in the admin portal.
- 12- As a user, I would like the table DQIS to pop up with 2 columns: one with food groups and a total at the end, and another with the corresponding DQIS scores.
- 13- As a user, I want to be able to have the DQIS data exported with the other food results.

Pending User Stories

As a user, I would like for the English and Spanish translations to be accurate.

As a user, I want to see the recorded times a caregiver enters the educational resource page.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- Visual Studio Code
- IntelliJ IDEA
- Spring-Boot - v.2.0
- MongoDB – v.4
- MongoDB Compass Community
- Studio 3T
- Java – JDK 11
- Maven – v.3.8.1
- Angular – v.7.3
- Git
- Node JS – v.12
- Postman
- Microsoft Azure DevOps
- Microsoft Azure Web Portal

Sprints Planning

Sprint 1

Sprint Planning Meeting Minutes 5/15/2023

Project: Babyfeed Web-Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam

Start time: 8:30 pm

End time: 9:00 pm

After discussion, the velocity of the team was estimated to be 1 week.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

User Story:

1. Get familiarized with the different features of Baby Feed
 - a. Create new users for clinics and for parents (intervention portals: CLINICIAN for healthcare providers assigning the users and PARENTAL for parents)
 - b. Create new users for research site and participants (control portals: RESEARCH for the researcher assigning the users and PARTICIPANTS for parents in the control group; they will only input data but will not see results).
 - c. Complete questionnaire (infant FFQ)
 - d. Complete tracking
 - e. See results
 - f. See recommendations
 - g. Add data for weight and length and see the chart.
2. Login to MongoDB, Azure, GitHub (text PO for access request)

The team members indicated their willingness to work on the following user stories.

- Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam
 - User Story 1
 - User Story 2

Sprint 2

Sprint Planning Meeting Minutes 5/29/2023

Project: Babyfeed Web-Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam

Start time: 8:30 pm

End time: 9:00 pm

After discussion, the velocity of the team was estimated to be 22 hours

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

User Story:

1. Implement collection of data for child's growth
 - a. Establish a default child value (1 per user account)
 - b. Add percentile for too high or too low (<2nd percentile or 98th percentile)
 - c. Fix issues with chart (not showing/saving data)
2. Other tasks if there is time:
 - a. Back button to go home
 - b. Add loading status bar everywhere
 - c. Change the Baby Feed Front Page from "Parental Portal Welcome to the Parent Portal" to "Welcome to the Parent/Caregiver Portal" only.

The team members indicated their willingness to work on the following user stories.

- Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam
 - User Story 1
 - User Story 2

Sprint 3

Sprint Planning Meeting Minutes 6/12/2023

Project: Babyfeed Web-Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam

Start time: 8:30 pm

End time: 9:00 pm

After discussion, the velocity of the team was estimated to be 18 hours across the 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

User Story:

1. Verify growth chart percentiles in code (2nd and 98th)
2. Fix the new user and password issues; not saving password so cannot login to the participant portal (control group)
 - a. Add a section to input infant's age, weight and length (no chart just data input table)
3. Fixing breast milk volume calculations
4. Start with implementing Diet Quality index Score (DQIS) for results page

The team members indicated their willingness to work on the following user stories.

- Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam
 - User Story 1
 - User Story 2
 - User Story 3
 - User Story 4

Sprint 4

Sprint Planning Meeting Minutes 6/26/2023

Project: Babyfeed Web-Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam

Start time: 8:30 pm

End time: 9:00 pm

After discussion, the velocity of the team was estimated to be 20 hours across the 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

User Story 1: Work on the DQISS

- Show results of the DQIS only in the Admin portal as a separate item
- Name it as DQIS
- Show pop up as a table with 2 columns: one with the food groups with total at the end and another with the corresponding score (DQIS)
- Download excel: same columns as the other results, each food group with its DQIS, and total DQIS at the end

The team members indicated their willingness to work on the following user stories.

- Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam

- Story 1: Implement the Diet Quality Index Score (DQIS) for admin users

Sprint 5

Sprint Planning Meeting Minutes 7/10/2023

Project: Babyfeed Web-Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam, Saily Hernandez

Start time: 8:30 pm

End time: 9:00 pm

After discussion, the velocity of the team was estimated to be 20 hours across the 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

User Story 1: Finishing touches

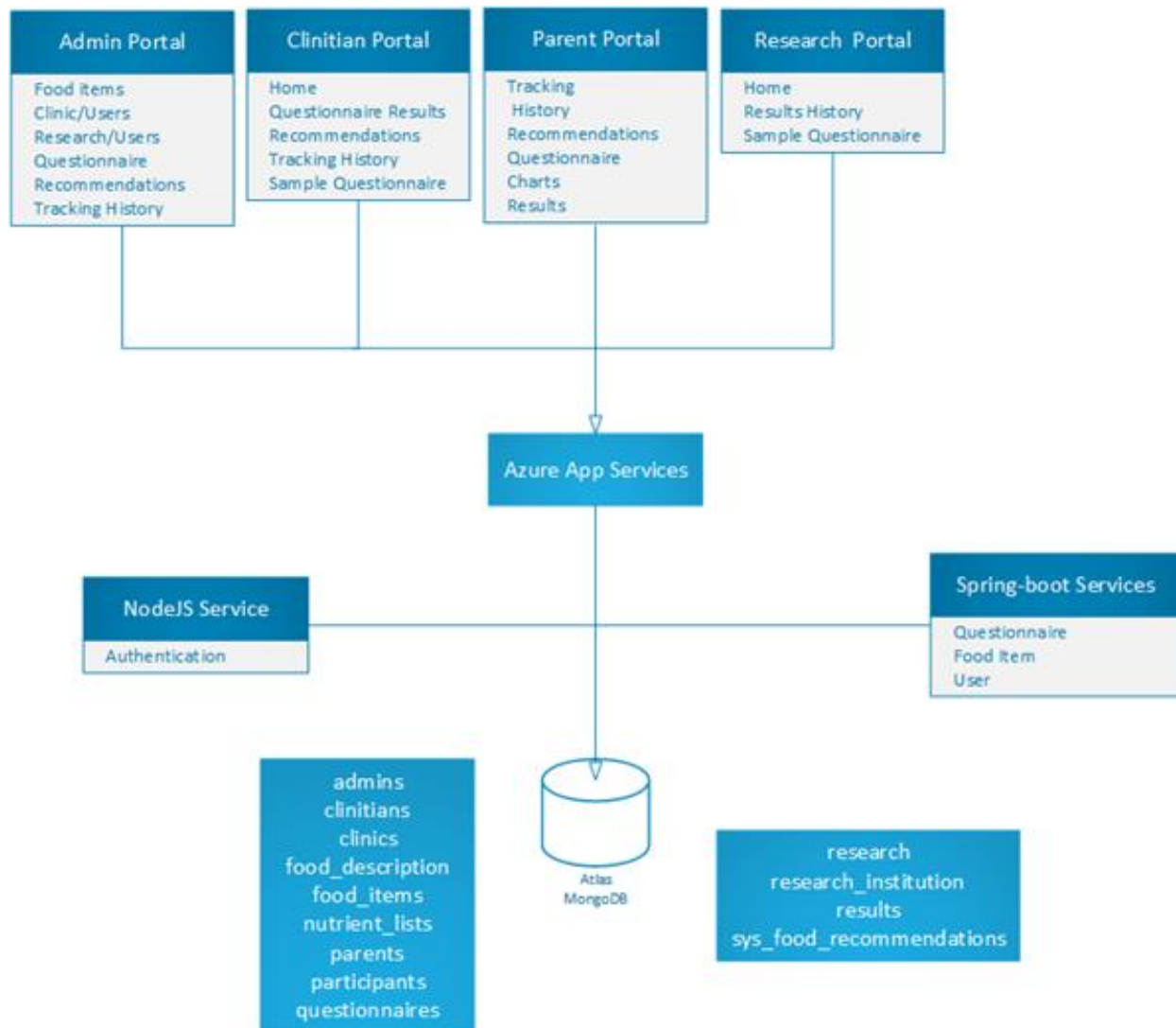
- Download excel: same columns as the other results, each food group with its DQIS, and total DQIS at the end
- Spinning wheel everywhere (loader)
- Start pushing different components into production

The team members indicated their willingness to work on the following user stories.

- Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam, Saily Hernandez
 - User Story 1: Finishing touches

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.



Architectural Patterns

The architectural patterns used for this project are:

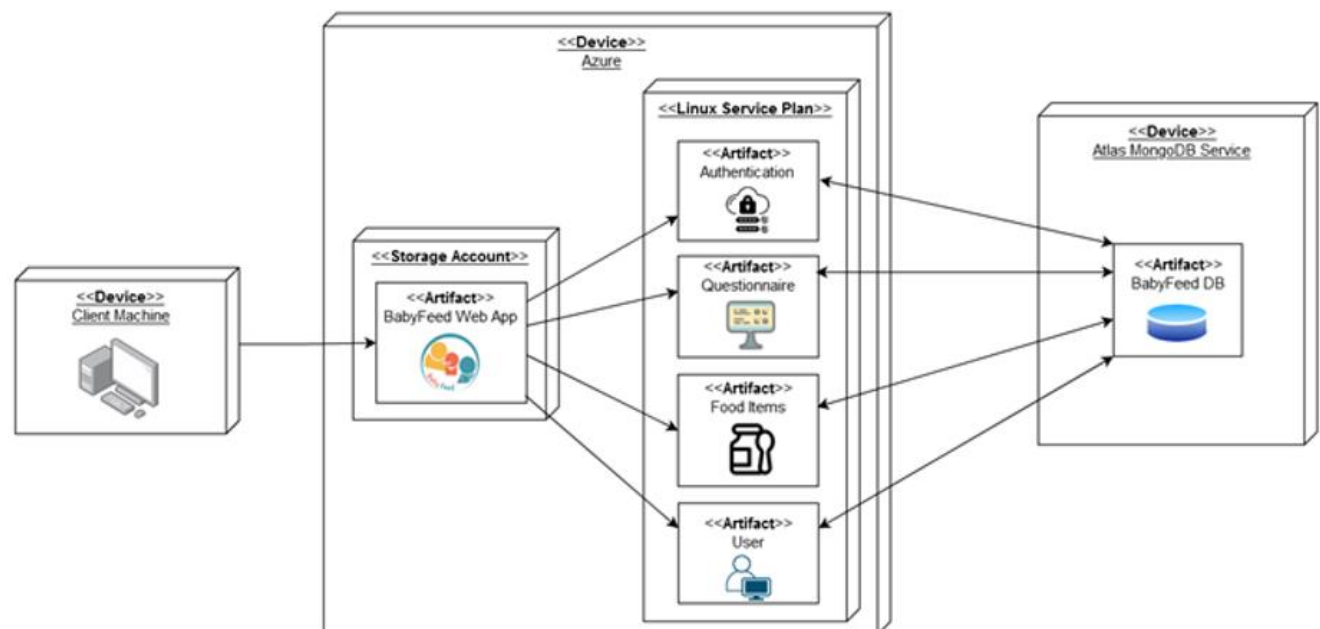
Client-Server Architecture

The functions of our system are split between the online web portal and the Node.JS authentication service. With this type of architecture, general functionality of the project can be available to all clients without being implemented by all of the different services offered.

Microservices Architecture

There are multiple different services that are offered with Baby Feed; specifically, there are multiple portals to be used: Admin, Parent, Researcher, and Clinician. This type of architecture allows for smaller and faster deployments of code as well as allowing different services to be unaffected by the issues of another.

Deployment Diagram



SYSTEM VALIDATION

To validate the effectiveness and reliability of the changes implemented to Baby Feed, extensive testing and experimentation of the application were performed. Since the primary development feature for this term was the implementation of the DQIS, most of the testing was geared towards finding faults in that feature's performance.

Numerous questionnaires were filled out by various parties—including the development team, the product owner, and a PhD student—and in a manner that tested for several possible scenarios, including edge cases meant to test the limits of the system's ability to warn users of outlier values.

Additionally, tests were performed on secondary features like UI changes as well as revisions to older features like the growth chart. UI changes, such as the implementation of loading bars, were tested in different environments to ensure they would work reliably regardless of the access point. Compatibility was also tested in this manner. For example, the site was explored and utilized with different browsers to ensure that the images in the questionnaires loaded properly regardless of how users accessed the site.

For older features that needed some minor revisions, such as the growth chart, testing revolved around ensuring that the system rejected outlier values and properly documented other changes. As with the DQIS, edge cases were used to test the limits of the system and other experiments were performed to ensure that values were recorded accurately and displayed properly. The system met expectations in all tests that were conducted.

APPENDIX

Appendix A - UML Diagrams

Use Case Diagrams

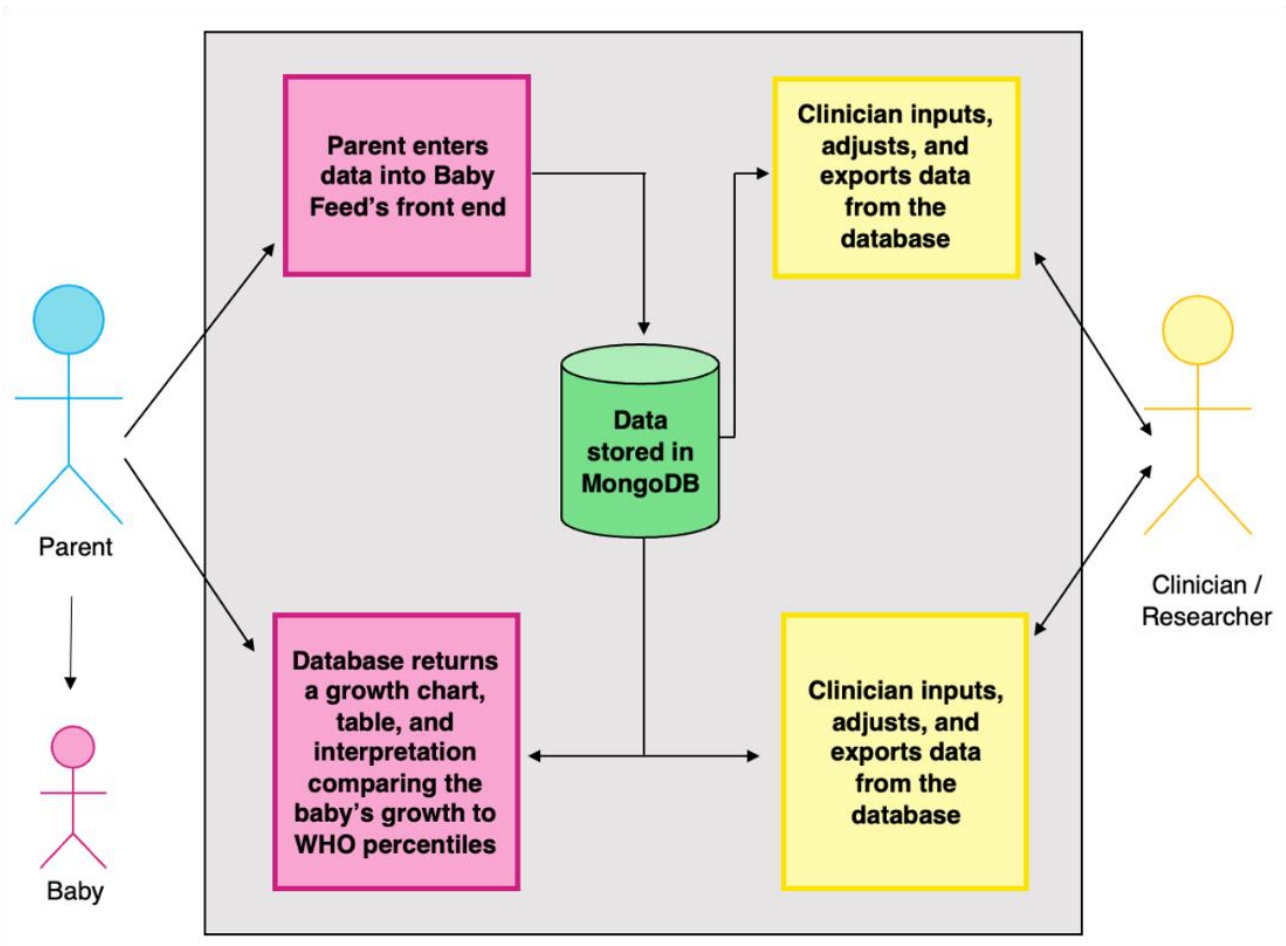


Figure 1 Use case diagram for the menu in the charts tab for the parent portal.

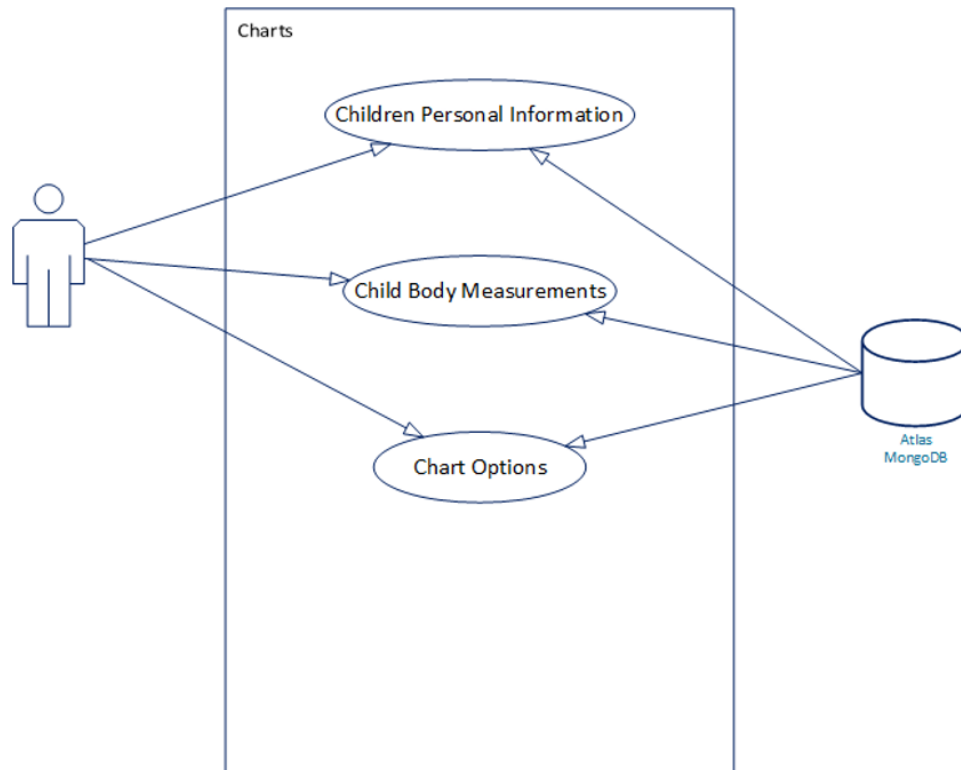


Figure 2 Use case diagram for the children input menu in the Charts tab.

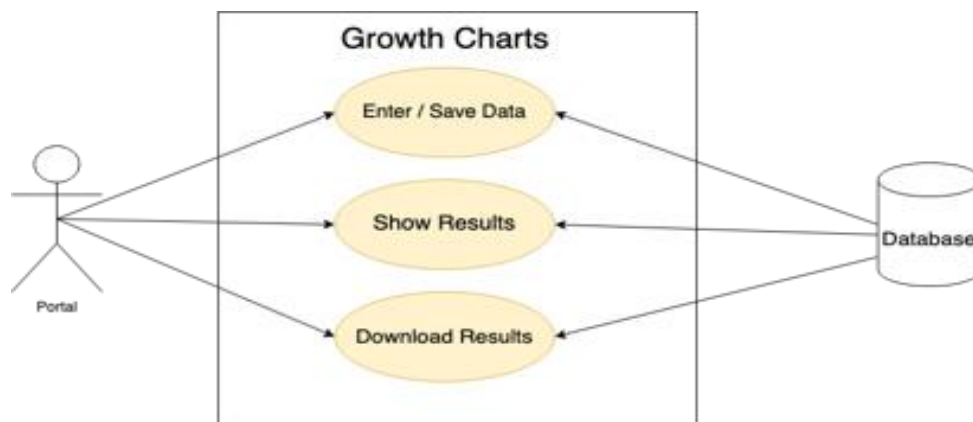


Figure 3 Use case diagram to save the input data in the Charts tab.

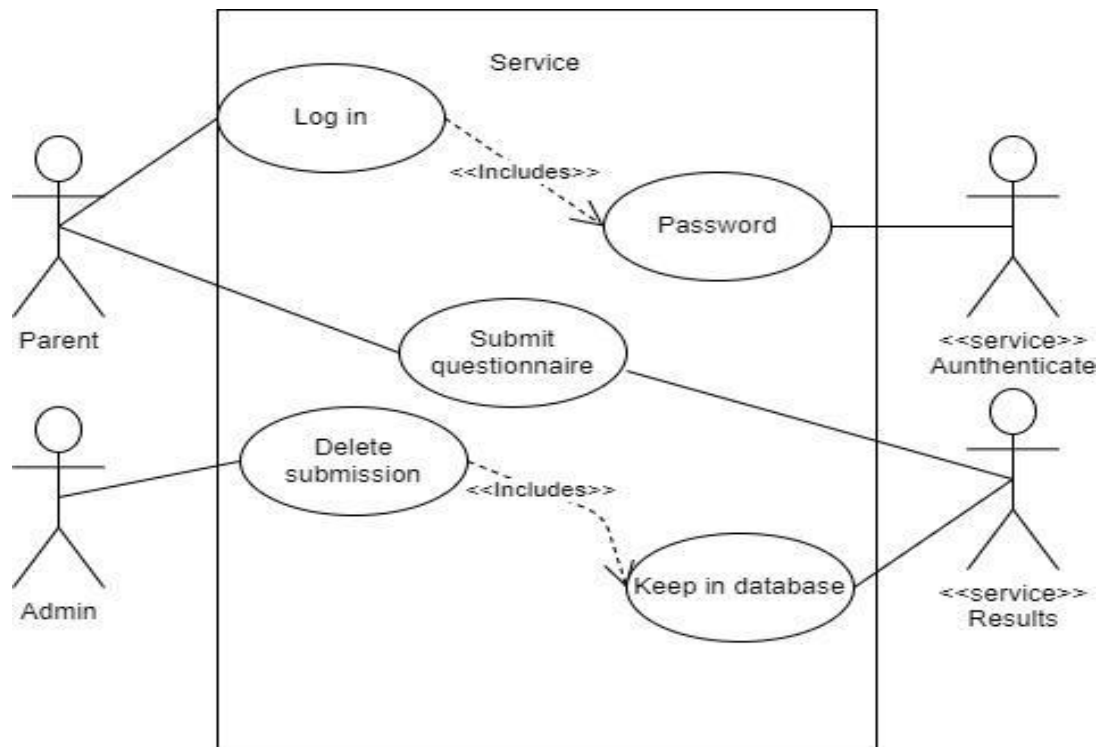


Figure 4 Use case diagram for parent login and admin portal deletion of a questionnaire.

Sequence Diagrams

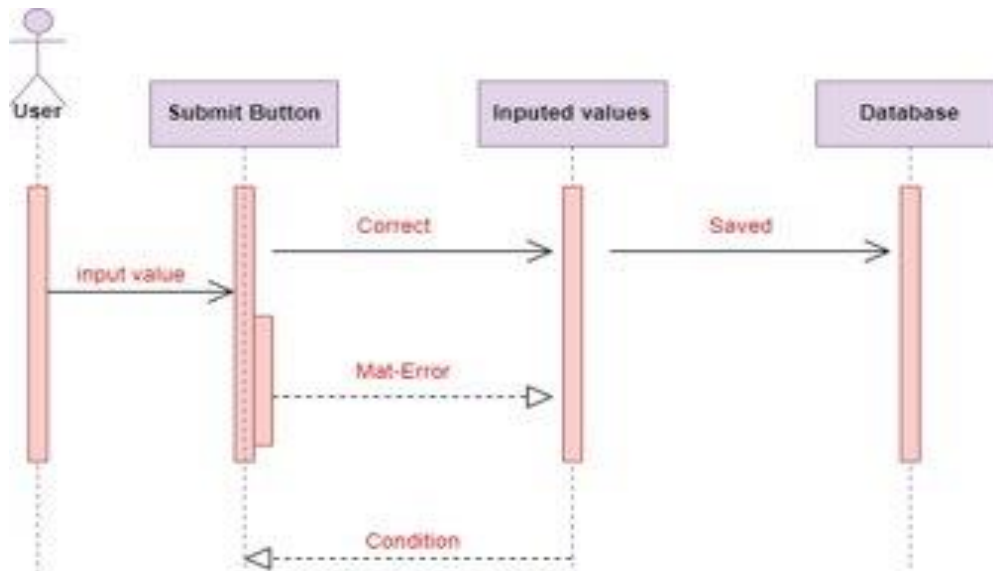


Figure 1 Sequence diagram of pressing the submit button for a questionnaire.

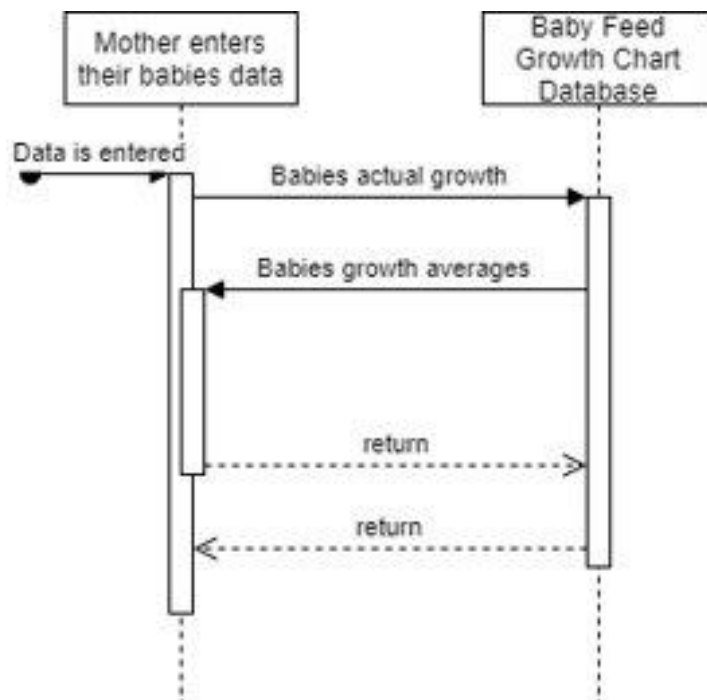


Figure 2 Sequence diagram of inputting data into the menu in the Charts tab.

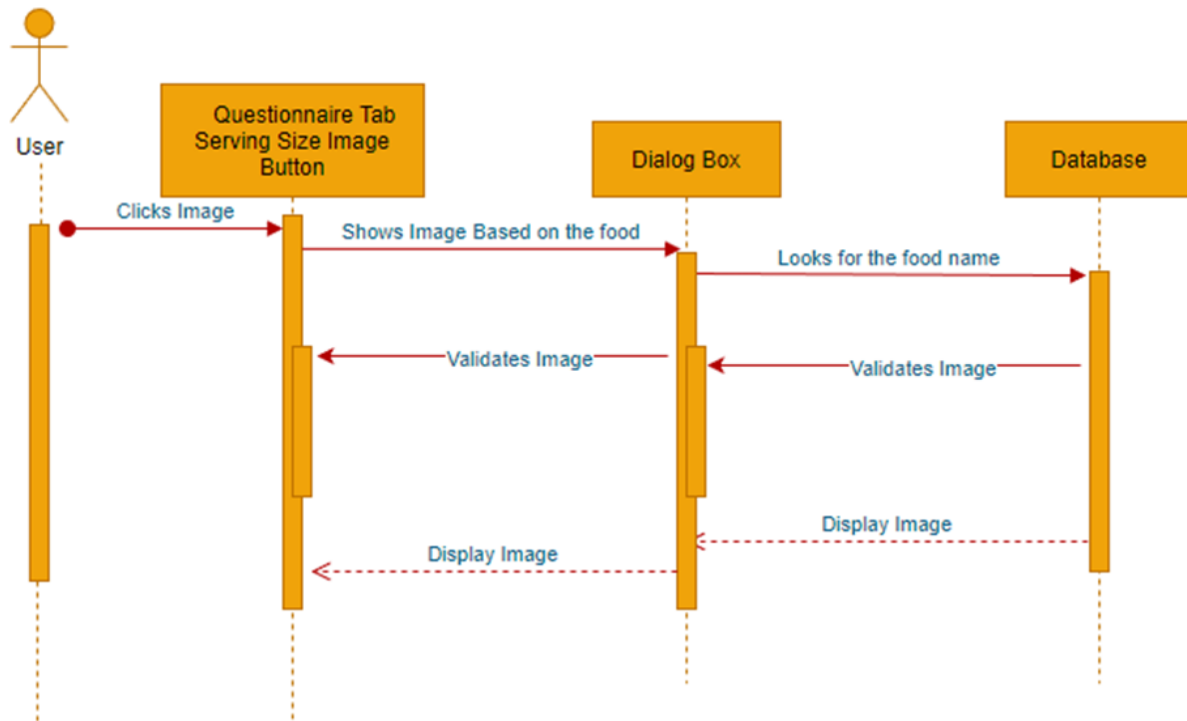


Figure 3 Sequence diagram when clicking the serving size button in a questionnaire.

Class Diagrams

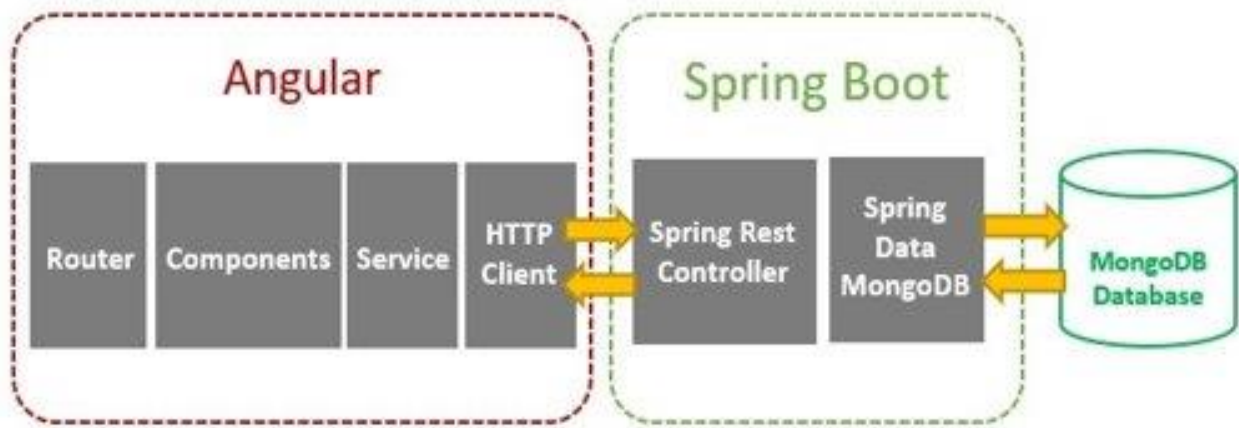


Figure 1 Class Diagram of connection between the Charts tab and MongoDB.

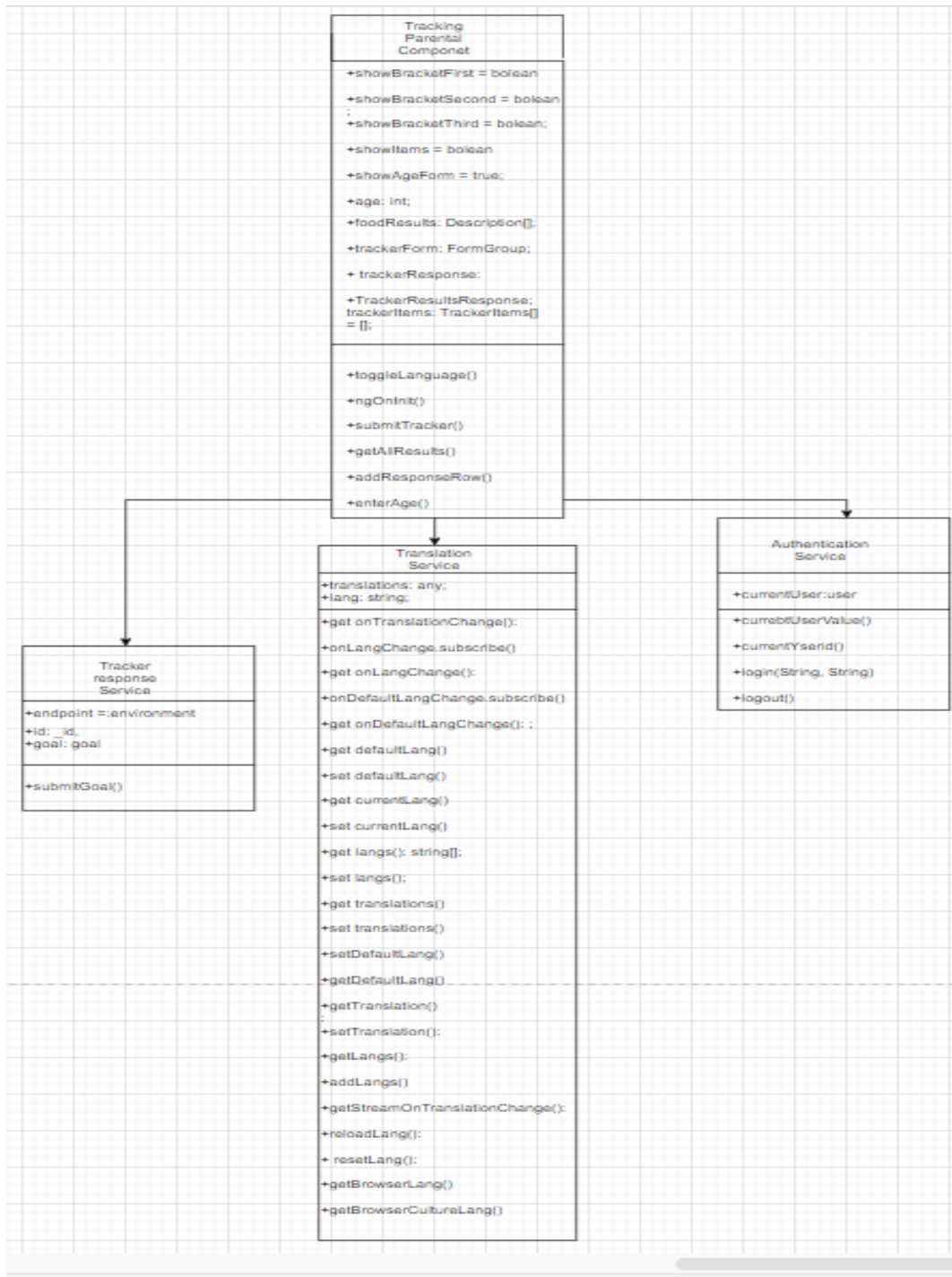


Figure 2 Class diagram of Parent portal tracking tab.

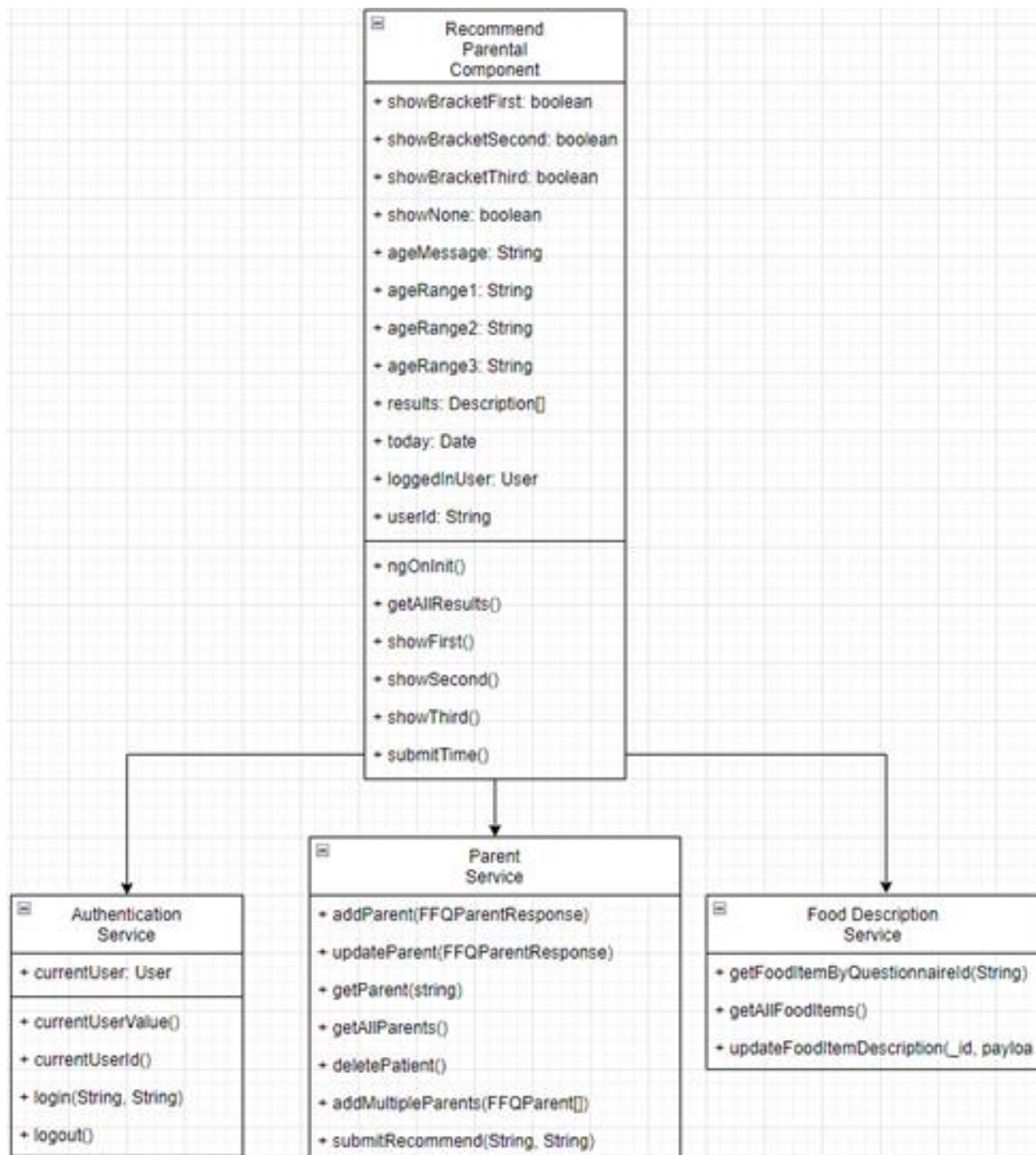


Figure 3 Class diagram of Recommended page.

Appendix B - User Interface Design

Figure 1 Picture of Questionnaire Results page in Admin Portal.



Figure 2 Picture of DQIS results in Admin Portal.

DQIS

Food Item	Calculated Points
Milk	10.0
Whole Grains	2.50
Refined Grains	2.50
Proteins	0.00
Vegetables	0.00
Fruits	2.50
100% Juices	5.00
Sugary Beverages	5.00
Sweets	5.00
Salty Snacks	5.00
Total Points	37.5

Ok

Appendix C - Sprint Review Reports

Sprint 1 Review Meeting Minutes

Project: Babyfeed Web Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam

Start time: 8:30 pm

End time: 9:00 pm

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

User Story:

1. Get familiarized with the different features of Baby Feed
2. Login to MongoDB, Azure, Github (text PO for access request)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

User Story

- Create new users for clinics and for parents (intervention portals: CLINICIAN for healthcare providers assigning the users and PARENTAL for parents)
- Create new users for research site and participants (control portals: RESEARCH for the researcher assigning the users and PARTICIPANTS for parents in the control group; they will only input data but will not see results).
- Complete questionnaire (infant FFQ)
- Complete tracking
- See results
- See recommendations
- Add data for weight and length and see the chart.

Sprint 2 Review Meeting Minutes

Project: Babyfeed Web Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, Saily Hernandez, James Lam

Start time: 8:30 pm

End time: 9:00 pm

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

User Story:

1. Adding the child collection in Mongo and incorporating it into the code [High Priority]
2. Button to go home from chart page [Low Priority]
3. Loading Status bar everywhere [Low Priority]

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

N/A

Sprint 3 Review Meeting Minutes

Project: Babyfeed Web Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam

Start time: 7:30 pm

End time: 8:00 pm

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

User Story:

1. Verify growth chart percentiles in code
2. Fixing breast milk volume calculations
3. Change "Other" Tab to "Educational Resources from Experts"

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. Added Researcher user portal to put infants age and weight (currently reads wrong user information)
2. Calculate the Diet Quality Index Score (DQIS)

Sprint 4 Review Meeting Minutes

Project: Babyfeed Web Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam, Saily Hernandez

Start time: 10:30 pm

End time: 11:00 pm

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

User Story:

1. Show results of the DQIS only in the Admin portal as a separate item
2. Name it as DQIS
3. Show pop up as a table with 2 columns: one with the food groups with total at the end and another with the corresponding score (DQIS)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. Download excel: same columns as the other results, each food group with its DQIS, and total DQIS at the end

Review 5 Meeting Minutes

Project: Babyfeed Web Application

Attendees: Jose Altamirano, Michael Ugarte, Katie Chavez, James Lam, Saily Hernandez

Start time: 10:30 pm

End time: 11:00 pm

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

User Story:

1. Download excel: same columns as the other results, each food group with its DQIS, and total DQIS at the end
2. Loading bar on all pages
3. Push all code to production.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- There are no future sprints planned for this development team (term has ended).

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Installation Manual

Due to compilation issues on Windows and MacOS, the development for this project is done in Ubuntu. More specifically, it is necessary to use Ubuntu 20.04 due to compatibility issues with software products (MongoDB, to be precise) that are critical to development. As such, for those with Windows and Mac devices it is recommended to use a VM to set up and work on the project over the course of the semester.

First, download the Ubuntu 20.04 desktop image from the Ubuntu website. Next, download a Virtual Machine in which to run the development environment. Previous development teams have used free options like VMware Workstation Player and Oracle VM VirtualBox to great effect. Once a VM has been set up with Ubuntu 20.04, carefully follow these provided instructions for installing the software suite used for development (please be aware that installing a different version of the used dependencies might give you unexpected errors or issues):

Install Visual Studio Code

- | | |
|--|----------------------------------|
| 1. open the terminal: | (ctrl + alt + t) |
| 2. install visual studio code: | sudo snap install --classic code |
| 3. to run vs code on the terminal run: | code |

Install Mongo DB

- | |
|---|
| 1. open the terminal: ctrl + alt + t |
| 2. run the following: |
| a. sudo apt update |
| b. sudo apt-get install gnupg |
| c. wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc sudo apt-key add - |
| d. echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list |
| e. sudo apt update |
| f. sudo apt-get install -y mongodb-org |

Install JDK 11

1. update the system: `sudo apt update`
2. install default jdk: `sudo apt install default-jdk`
3. to verify jdk version: `java -version`

Note: At the moment of this guide, the default-jdk for Ubuntu 20.04 is jdk 11. You need to install jdk version 11. A different version can give you unexpected errors

Check the status of MongoDB

1. run the following:
 - a. check the status of MongoDB: `sudo systemctl status mongod`
 - b. stop the process on the terminal: `ctrl + c`
 - c. clear the screen on the terminal: `clear`

```
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: https://docs.mongodb.org/manual
```

Note: MongoDB is inactive, so you will need to activate it.

Activate Mongo DB

1. run the following:
 - a. `sudo systemctl enable mongod`

Start Mongo DB

1. run the following:
 - a. start MongoDB: `sudo systemctl start mongod`
 - b. check the status of MongoDB: `sudo systemctl status mongod`
 - c. stop the process on the terminal: `ctrl + c`
 - d. clear the screen on the terminal: `clear`

```
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-07-26 10:14:25 PDT; 14s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 7346 (mongod)
  Memory: 161.9M
    CGroup: /system.slice/mongod.service
            └─7346 /usr/bin/mongod --config /etc/mongod.conf
```

Install MongoDB Compass

MongoDB Compass is a tool that allows one to connect to a local database and make modifications to the database using a graphical user interface.

1. run the following:
 - a. `sudo apt install /home/val0dia/Downloads/mongodb-compass_1.32.5_amd64.deb`

Install Apache Maven 3.8.1

First, download the Apache Maven version 3.8.1 using the following link: [Apache Maven 3.8.1](#). A different version can cause unexpected errors and compilation issues.

1. unzip the file to /opt:
`sudo tar xf /home/val0dia/Downloads/apache-maven-3.8.1-bin.tar.gz -C /opt`
Note: You need to update the path for your file accordingly.

Set up environment variable for Apache Maven 3.8.1

1. create a file named maven.sh.
2. copy inside maven.sh the following:

```
export JAVA_HOME=/usr/lib/jvm/default-java
export M2_HOME=/opt/apache-maven-3.8.1
export MAVEN_HOME=/opt/apache-maven-3.8.1
export PATH=${M2_HOME}/bin:${PATH}
```
3. copy maven.sh file to the /etc/profile.d:

```
sudo cp /home/val0dia/Desktop/maven.sh /etc/profile.d/maven.sh
```
4. make the script executable:

```
sudo chmod 777 /etc/profile.d/maven.sh
```
5. load the environment:

```
source /etc/profile.d/maven.sh
```
6. check version:

```
mvn -version
```

```
val0dia@ubuntu:~$ mvn -version
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: /opt/apache-maven-3.8.1
Java version: 11.0.15, vendor: Private Build, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-41-generic", arch: "amd64", family: "unix"
```

Note: You need to update the path for your file accordingly.

Install Nodejs 12

Note: installing a different version will cause compilation issues.

1. run the following:
 - a.

```
sudo apt install curl
```
 - b.

```
curl -sL https://deb.nodesource.com/setup\_12.x -o /tmp/nodesource_setup.sh
```
 - c.

```
sudo bash /tmp/nodesource_setup.sh
```
 - d.

```
sudo apt install nodejs
```
 - e.

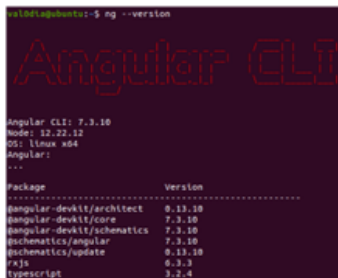
```
node -v
```

```
val0dia@ubuntu:~$ node -v
v12.22.12
```

Install Angular 7.3

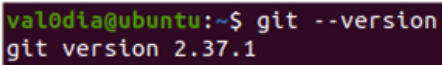
Note: installing a different version will cause compilation issues.

1. run the following:
 - a. `sudo npm install -g @angular/cli@7.3`
 - b. `ng --version`



```
val0dia@ubuntu:~$ ng --version
Angular CLI
Angular CLI: 7.3.10
Node: 12.22.12
OS: linux x64
Angular:
...
Package                         Version
-----                         -
angular-devkit/architect        0.13.10
angular-devkit/core             7.3.10
angular-devkit/schematics       7.3.10
schematics/angular             7.3.10
schematics/update               0.13.10
rxjs                            6.5.3
typescript                      3.7.4
```

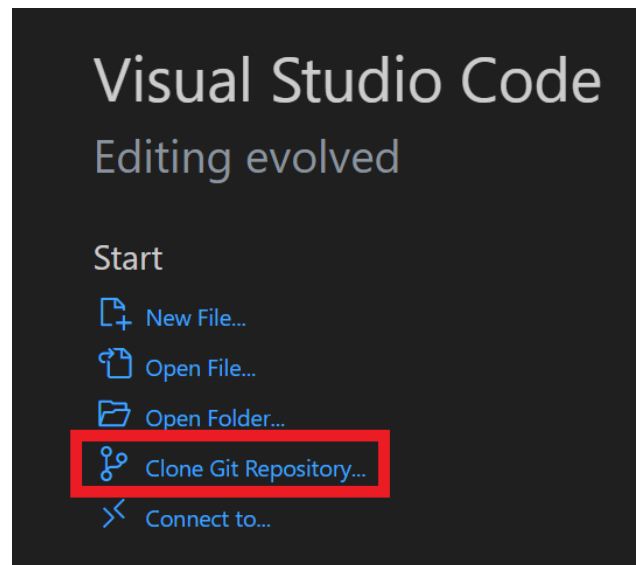
Set Up GitHub

1. run the following:
 - a. `sudo add-apt-repository ppa:git-core/ppa`
 - b. `sudo apt update`
 - c. `sudo apt install git`
 - d. `git --version`
- 
- ```
val0dia@ubuntu:~$ git --version
git version 2.37.1
```
2. config your GitHub:
    - a. `git config --global user.name "your_user_name"`
    - b. `git config --global user.email "your_email@example.com"`
    - c. `git config --list`

Note: To configure your GitHub account from Visual Studio Code click on the following click: [Using Git with Visual Studio Code](#)

### Set Up Project in Visual Studio Code

Now that GitHub account access has been established, clone the Baby Feed repository.



After cloning Baby Feed, follow these instructions to set up each component repository:

#### ffq-authentication-service

1. Right click on [ffq-authentication-service](#) folder and click on [Open in integrated Terminal](#)
2. run on the terminal: `npm install`
3. stop the process: `ctrl + c`

Note: To run [ffq-authentication-service](#) repeat step 1 and run on the terminal: `npm start`

### ffq-food-item-service

4. Right click on [ffq-food-item-service](#) folder and click on [Open in integrated Terminal](#)
5. run on the terminal:  
`mvn spring-boot:run -Dspring-boot.run.arguments="mongo.fooditems.load=true"`
6. stop the process: `ctrl + c`

Note: To run [ffq-food-item-service](#) repeat step 4 and run on the terminal: `mvn spring-boot:run`

### ffq-user-service

7. Right click on [ffq-user-service](#) folder and click on [Open in integrated Terminal](#)
8. run on the terminal:  
`mvn spring-boot:run -Dspring-boot.run.arguments="mongo.users.load=true"`
9. stop the process: `ctrl + c`

Note: To run [ffq-user-service](#) repeat step 7 and run on the terminal: `mvn spring-boot:run`

### ffq-questionnaire-web

4. Right click on [ffq-questionnaire-web](#) folder and click on [Open in integrated Terminal](#)
5. run on the terminal: `npm install`
6. stop the process: `ctrl + c`
7. to run [ffq-questionnaire-web](#) repeat step 7 and run on the terminal: `ng serve --open`
8. open the browser and type: `localhost: 4200`

Note: Instead of typing `ng serve --open` in step 10 you can type `npm start` and open the browser and go to [localhost: 4200](#)

### ffq-questionnaire-service

7. Right click on [ffq-questionnaire-service](#) folder and click on [Open in integrated Terminal](#)
8. run on the terminal:  
`mvn spring-boot:run -Dspring-boot.run.arguments="mongo.questionnaires.load=true"`
9. stop the process: `ctrl + c`

Note: To run [ffq-questionnaire-service](#) repeat step 10 and run on the terminal: `mvn spring-boot:run`

### Set Up MongoDB Compass

Open MongoDB Compass, skip the first dialog box displayed, and accept the default options in the second dialog box. Click on “connect to load the database.” Some of the databases may not load the data automatically, so it may be necessary to load the files manually. Clicking on the ffq-database database should display the following collections:

1. Admin user collection: admins
2. Clinician user collection: clinicians
3. Clinic collection: clinics
4. Parent user collection: parents
5. Food description collection: food\_description
6. Food item collection: food\_items
7. Nutrients lists collection: nutrient\_lists
8. Participants collection: participants
9. Questionnaire collection: questionnaires
10. Research institution collection: research\_institution
11. Researchers collection: researchers
12. Results collection: results
13. Sys Food Recommendations collection: sys\_food\_recommendations
14. Sys Nutrients Recommendations collection: sys\_nutrients\_recommendations

If any of these collections are missing, click on the plus sign next to ffq\_database to create a collection. Type the name of the collection and click on the “create collection” button. After creating a collection, select the collection and click on “import data.” Select the input file as JSON and select the appropriate file. The locations of the files in the repo are as follows:

1. admins: ffq-user-service/src/main/resources/AdminPayload.json
2. clinicians: ffq-user-service/src/main/resources/ClinicianPayLoad.json
3. clinics: ffq-user-service/src/main/resources/ClinicPayLoad.json
4. parents: ffq-user-service/src/main/resources/ParentPayLoad.json
5. participants: ffq-user-service/src/main/resources/ResearcherParticipantPayload.json

6. research\_institution: ffq-user-service/src/main/resources/ResearchInstitutionPayload.json
7. researchers: ffq-user-service/src/main/resources/ResearchPayload.json
8. food\_description: ffq-food-item-service/src/main/resources/FoodDescriptionPayload.json
9. food\_items: ffq-food-item-service/src/main/resources/FoodItemPayload.json
10. nutrient\_lists: ffq-food-item-service/src/main/resources/SysNutrientsRecommendationsPayload.json
11. questionnaires: ffq-food-item-service/src/main/resources/ResearchResultPayload.json
12. results: ffq-food-item-service/src/main/resources/ResearchResultPayload.json
13. sys\_food\_recommendations: ffq-food-item-service/src/main/resources/SysFoodRecommendationsPayload.json
14. sys\_nutrients\_recommendations: ffq-food-item-service/src/main/resources/SysNutrientsRecommendationsPayload.json

### Importing Collection Payloads from Cloud Database (optional)

The payloads that are saved in the codebase are outdated and don't have the current changes for users and information from the cloud server. If a development team wants to work with the updated information, it is possible to install the payloads directly from the cloud server and import them into a local host.

Step 1: Go into database access in MongoDB Atlas and pick a user to login with on the Compass in the Virtual Machine.

Step 2: Login to the cloud database on your compass.

Step 3: Go to each collection and download the data as a text file that can be imported into your local database.

Step 4: Connect to the local database and delete all the data from each collection and import the new text files into each collection.

## Azure Pipelines and Pushing to Production

1. Commit all changes to the main branches.
2. Ensure that the changes are properly deployed through the Azure pipelines (2 checks).
  - o Log into Azure DevOps with the PO account.
  - o Click on **Pipelines** (1<sup>st</sup> check), then **Releases**.
  - o Check that the relevant pipelines are deploying (2<sup>nd</sup> check).
    - Getting the pipelines to deploy properly may require some troubleshooting. Plan your deployment schedule accordingly.

|                                       |                                                                                       |
|---------------------------------------|---------------------------------------------------------------------------------------|
| ✓ babyfeed.ffq-food-item-service      | #20230719.2 • summer 2023 semi-full commit<br>Manually triggered for  master          |
| ✓ babyfeed.ffq-user-service           | #20230719.2 • summer 2023 semi-full commit<br>Manually triggered for  master          |
| ✓ babyfeed.ffq-questionnaire-web      | #20230719.1 • summer 2023 semi-full commit<br>Individual CI for  master               |
| ✓ babyfeed.ffquestionnaire-service    | #20230409.2 • Merge pull request #6 from jwils133/master<br>Individual CI for  master |
| ✓ babyfeed.ffq-authentication-service | #20230409.1 • Adding node_modules<br>PR automated for  9                              |

*Figure 1 Pipelines (1st check)*

| ffq-questionnaire-web                      |                        |         |  |
|--------------------------------------------|------------------------|---------|--|
| Releases Deployments Analytics             |                        |         |  |
| Releases                                   | Created                | Stages  |  |
| <b>Release-226</b><br>20230719....  master | 7/18/2023, 8:17:04 PM  | Stage 1 |  |
| <b>Release-225</b><br>20230421....  master | 4/20/2023, 11:06:25 PM | Stage 1 |  |

*Figure 2 Releases (2nd check)*

## **Shortcomings / Wishlist**

As it stands, Baby Feed performs its core function well enough. As with any software product, however, there is always room for improvement. The following is a non-exhaustive list of improvements this dev team would like to see in future iterations of the project:

1. **A full implementation of the “child value” user story.**

In the second sprint of this term, the dev team was tasked with ensuring that user accounts had a default string value (“child”) assigned to the name category of every child. The solution that was implemented during this sprint was useful for testing purposes but a permanent solution is required for proper use.

2. **A finished implementation of the translation feature.**

Having all relevant website features available in Spanish would greatly enhance accessibility, especially when considering the demographics of South Florida. Care should be taken to ensure that translations are free of spelling and syntax errors.

3. **Additional UI changes to improve the user experience.**

- i. Implementing a home button in the Participants portal (like the one implemented in the Parents portal this term).
- ii. Implementing some kind solution that would reduce the amount of scrolling users have to do during questionnaires. Although one has technically already been implemented, there are a few bugs associated with its use.

## **DQIS Table**

**Table 1** - Diet Quality Index Score (DQIS) for infants and toddlers.

| <b>Component</b>                                 | <b>Score</b> | <b>0 - &lt; 6 months</b>        | <b>6- &lt; 12 months<br/>(Portions<sup>a</sup>)</b> | <b>12-36<br/>months<br/>(Portions<sup>a</sup>)</b> |
|--------------------------------------------------|--------------|---------------------------------|-----------------------------------------------------|----------------------------------------------------|
| Milk                                             |              |                                 |                                                     |                                                    |
| Breast milk<br>(exclusive) <sup>b</sup>          | 15           | Yes                             | Yes                                                 | Yes                                                |
| Breast milk<br>(partial) <sup>b</sup>            | 10*/5        | Yes, *0-5.9 months              | Yes                                                 | Yes                                                |
| Formula/Cow's<br>Milk/Other milks (fl<br>oz/day) | 5            | Yes                             | 0.1-28                                              | 0.1-18                                             |
|                                                  | 2.5          | n/a                             | 28.1-35.0                                           | 18.1-24.0                                          |
|                                                  | 0            | n/a                             | 0 or >35*                                           | 0 or >24.0*                                        |
| Grains                                           |              |                                 |                                                     |                                                    |
| Whole Grains<br>(oz/day) <sup>c</sup>            | 5*/2.5       | Not consumed, *0-<br>5.9 months | 0.1-3.5                                             | 0.1- 5.5                                           |
|                                                  | 1.25         | n/a                             | 3.6-8.0                                             | 5.6-8.0                                            |
|                                                  | 0            | Any intake                      | 0 or >8.0                                           | 0 or >8.0                                          |
| Refined Grains<br>(oz/day) <sup>c</sup>          | 5*/2.5       | Not consumed, *0-<br>5.9 months | 0-1.5                                               | 0-1.8                                              |
|                                                  | 1.25         | n/a                             | 1.6-3.5                                             | 1.9-4.2                                            |
|                                                  | 0            | Any intake                      | >3.5                                                | >4.2                                               |
| Proteins (oz/day) <sup>d</sup>                   | 5            | Not consumed, 0<br>intake       | 0.1-6.0                                             | 0.1-3.0                                            |
|                                                  | 2.5          | n/a                             | 6.1-10.0                                            | 3.1-6.0                                            |
|                                                  | 0            | Any intake                      | 0 or >10                                            | 0 or >6.0                                          |
| Vegetables (oz/day)                              | 5            | Not consumed                    | ≥2.0                                                | ≥8.0                                               |
|                                                  | 2.5          | n/a                             | 0.1-1.9                                             | 0.1-7.9                                            |
|                                                  | 0            | Any intake                      | 0                                                   | 0                                                  |
| Fruits (oz/day)                                  | 5            | Not consumed, 0<br>intake       | ≥2.0                                                | ≥8.0                                               |
|                                                  | 2.5          | n/a                             | 0.1-1.9                                             | 0.1-7.9                                            |
|                                                  | 0            | Any intake                      | 0                                                   | 0                                                  |

|                                    |     |                        |         |         |
|------------------------------------|-----|------------------------|---------|---------|
| 100% Juices (fl oz/day)            | 5   | Not consumed, 0 intake | 0       | 0-4.0   |
|                                    | 2.5 | n/a                    | 0.1-6.0 | 4.1-6.0 |
|                                    | 0   | Any intake             | >6.0    | >6.0    |
| Sugary Beverages (fl oz/day)       | 5   | Not consumed, 0 intake | 0       | 0       |
|                                    | 2.5 | n/a                    | 0.1-4   | 0.1-4   |
|                                    | 0   | Any intake             | >4      | >4      |
| Sweets <sup>e</sup> (oz/day)       | 5   | Not consumed, 0 intake | 0       | 0       |
|                                    | 2.5 | n/a                    | 0.1-1.0 | 0.1-1.0 |
|                                    | 0   | Any intake             | >1.0    | >1.0    |
| Salty snacks <sup>f</sup> (oz/day) | 5   | Not consumed, 0 intake | 0       | 0       |
|                                    | 2.5 | n/a                    | 0.1-1.0 | 0.1-1.0 |
|                                    | 0   | Any intake             | >1.0    | >1.0    |

\*0 if not consuming breastmilk or any other milk source

## CLOSING REMARKS

The Summer 2023 development team is proud of the work they managed to produce in such a short period of time. Work from previous dev teams was refined and expanded upon. UI changes that improve the user experience were enacted. Numerous bugs were fixed. The implementation of an entirely new construct for Baby Feed, the DQIS, was undertaken and thoroughly tested to ensure reliability. It was a success on all fronts.

Working on Baby Feed was beneficial in other ways too. It provided the team with experience developing a real-world product, using technologies and frameworks that many on the team were initially unfamiliar with. It also provided the team with an opportunity to collaborate and gain experience with Agile methodologies. All in all, it was a great learning experience for everyone involved and we're incredibly thankful for having been a part of such a wonderful project.

## REFERENCES

- Angular Docs. (2019). Retrieved from <https://angular.io/docs>
- Azure Documentation. (2020). Retrieved from <https://docs.microsoft.com/en-us/azure/?product=featured>
- Centers for Disease Control and Prevention. (2010, September 9). Growth charts - who child growth standards. Centers for Disease Control and Prevention. Retrieved June 7, 2023, from [https://www.cdc.gov/growthcharts/who\\_charts.htm](https://www.cdc.gov/growthcharts/who_charts.htm)
- Docker Documentation. (2019). Retrieved from <https://docs.docker.com/>
- Infant and Toddler Nutrition | CDC. (n.d.). Retrieved May 29, 2023, from <https://www.cdc.gov/nutrition/infantandtoddlernutrition/index.html>
- Mongodb Documentation. (2020). Retrieved from <https://docs.mongodb.com/>
- Poor Nutrition | CDC. (n.d.). Retrieved May 29, 2023, from <https://www.cdc.gov/chronicdisease/resources/publications/factsheets/nutrition.htm>
- Websystiqueadmin. (2017, September 30). Spring Boot Rest API Example. Retrieved May 29, 2023, from <http://websystique.com/springboot/springbootrestapiexample/>