

팀명: 상상하조

팀원: 성영은, 이성수, 정하솔, 조유빈

기간: 2023. 12. 01. ~ 12. 22.

프로젝트 완료 보고서

프론트엔드 응용 - 앱 기획 및 제작

1. 프로젝트 개요 (Overview)

React 라이브러리에 Javascript언어를 이용하여 구상한 앱의 디자인과 기능을 구현하고, MongoDB로 데이터베이스 저장 기능을 더함.

2. 기획의도 & 목표

- React를 학습하고 활용하여 코딩 능력 향상
- 사용자편의성을 고려하여 통일감 있게 제작한 UI디자인
- 여러 매체를 이용하여 팀원들 간의 소통능력 향상

3. 레퍼런스 앱

FREEZY, 냉싸부

4. 각 팀원의 담당 업무

이성수: 팀장 / 기능구현

성영은: 기록 / 기능구현

정하솔: Github 관리 / 디자인 총괄

조유빈: 디렉토리 관리 / 기능, 디

5. 작업 순서

- | | |
|------------------------------------|-------------------------------|
| 1) 기획할 앱 주제, 팀 명, 프로젝트 목표 선정 | 8) 각 컴포넌트 기본기능 작업 |
| 2) 기획할 앱 주제 선정 | 9) 컴포넌트 라우팅 작업 |
| 3) 업무 분배 | 10) 부수 기능 작업 후 로직 연결 |
| 4) 사용자 시나리오, work-flow, use-case구상 | 11) CSS작업 |
| 5) 요구사항 분석, 디렉토리 구조 설계, 코딩 가이드 작성 | 12) 디버깅작업 및 최종점검 |
| 6) 디자인 가이드 작성, 프로토타입 제작 | 13) 발표 및 피드백 |
| 7) 디자인 시안 작업, 이미지 소스 작업 | 14) 피드백을 기반으로 자료수정 및 완료보고서 작성 |

(날짜 별 작업과정: https://github.com/you-been/team_project_2)

6. 개발 환경

• 프로젝트 관리 Tool

- GitHub ([singsinggo \(github.com\)](https://github.com))
- Figma (디자인 시안 공유, 프로토 타입, 워크플로우 등 제작)
- Notion (팀 작업일지)
- Google spread sheet(요구 분석, 스타일 가이드 등 제작)

• 개발 Tool

- Visual Studio Code
- Adobe Illustrator

• 웹 구현

- React, JavaScript

• 플러그인 / 패키지

- @fullcalendar/react
- @fullcalendar/inreractio
- @fullcalendar/daygrid

- React-router-dom
- uuid

7. 프로젝트 내용

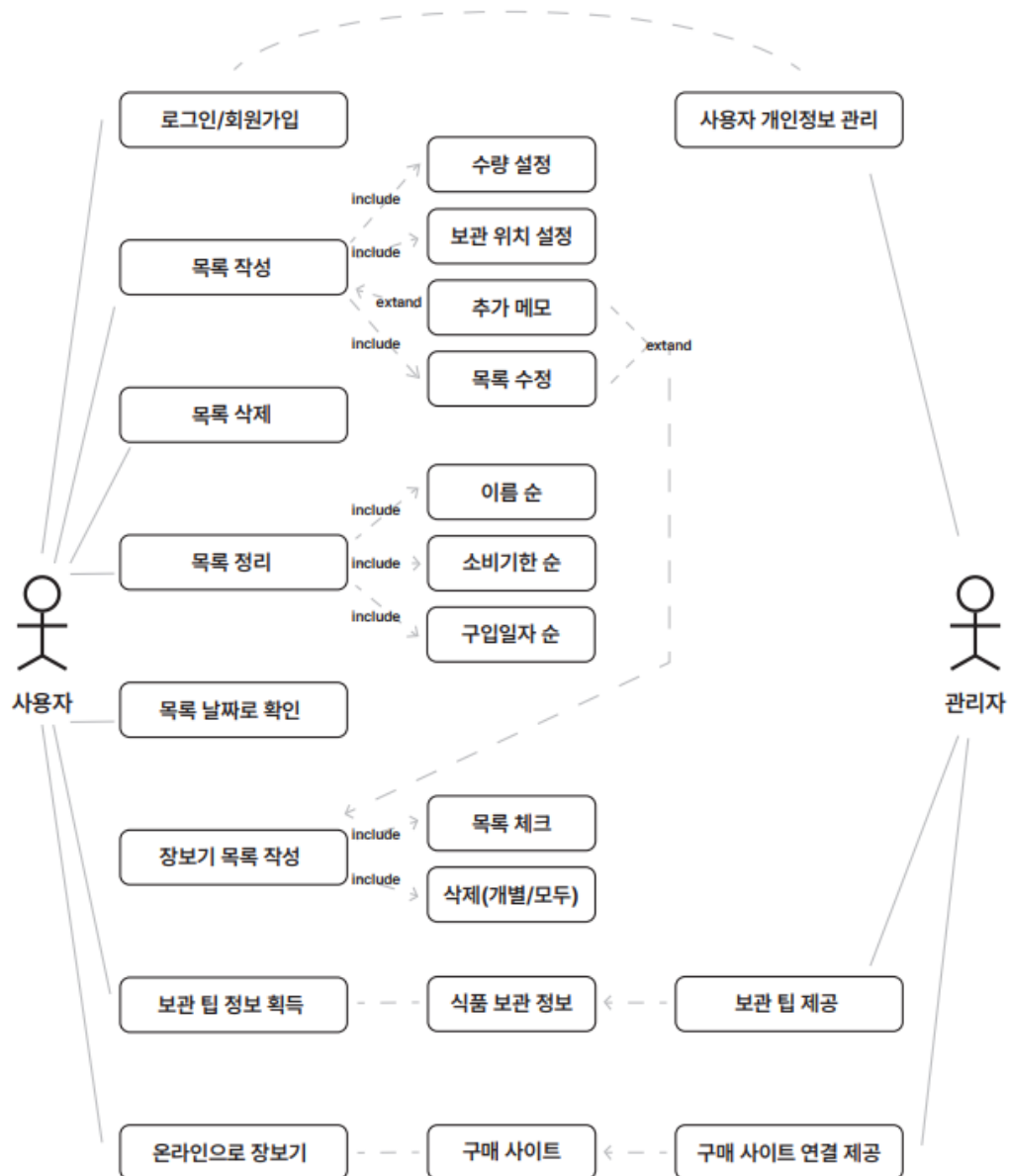
• 요구사항 분석

요구사항 분석 - 기능적		
서비스	필요기능	
회원관리	로그인	아이디와 패스워드를 입력하여 로그인
	회원가입	질처에 맞게 회원가입을 할 수 있도록 제공 (아이디, 비밀번호, 이메일)
	로그아웃	로그인 후 로그아웃
	아이디 중복체크	다른 사용자와 아이디가 중복되지 않게 함
내 냉장고	냉장고 등록	사용할 냉장고 등록
	목록 등록	사용자가 가지고 있는 식재료 추가
		식료품의 이름, 수량, 보관위치 소비기한 설정, 카테고리 분류, 메모 등의 기능이 필요 식료품명, 카테고리, 보관위치 소비기한이나 보관일자 등은 필수 작성항목으로 함
	목록 삭제	모두 사용한 식재료를 삭제하는 기능
	목록 수정	기록을 잘못 작성했을 시 수정이 가능한 기능
	소비기한이 다른 동일한 식료품의 추가	동일한 재료지만 구입날짜가 다르거나 소비기한이 다른 경우 따로 관리할 수 있게 해야 함
	식료품 검색	검색을 통하여 원하는 식료품을 쉽게 찾을 수 있음
	목록 정렬	가나다순, 소비기한순으로 선택지를 만들어 사용자가 원하는 방식으로 정리
	보관 위치 분류	목록 등록 시 선택한 보관 위치로 각각의 카테고리 목록 분류(냉장/냉동/실온)
	날짜별 식료품 확인	사용자가 등록한 식재료의 구입날짜를 캘린더로 확인
	소비기한 임박 알림	사용자에게 소비기한이 임박한 식료품에 대하여 리스트 색상 등 변화를 통해 알려줌
온라인 구매	외부 사이트 링크 연결	외부 온라인 홈페이지 링크를 연결하여 식료품의 정보를 확인하거나 주문할 수 있도록 제공함
장보기 리스트	장보기 목록 작성	사용자가 자유롭게 등록/삭제 가능한 쇼핑리스트를 작성함

요구사항 분석 - 비기능적	
서비스	필요기능
보안 및 개인정보보호	로그인 및 사용자 정보 처리 시, 보안 프로토콜을 적용하여 개인정보를 안전하게 보호함
사용자 친화성	직관적이고 사용자 친화적인 인터페이스를 제공하여 쉽게 기능을 이용할 수 있도록 함 오류 발생 시 오류메시지를 작성하여 사용자가 상태를 인식할 수 있도록 제공함
성능 및 신뢰성	빠르고 안정적인 서비스 제공을 위해 성능 및 신뢰성에 대한 최적화
다양한 플랫폼 지원	다양한 기기와 환경에서 서비스를 이용할 수 있도록 모바일 및 데스크톱 플랫폼에 대한 효율적인 지원을 제공함
확장성	새로운 기능 및 업데이트를 효율적으로 추가할 수 있도록 시스템의 확장성을 고려함
시스템 모니터링 및 로깅	시스템의 상태와 성능을 지속적으로 모니터링
품질 테스트	소프트웨어 개발 단계에서 지속적인 테스트를 통해 시스템의 품질을 확보하고 버그를 최소화함

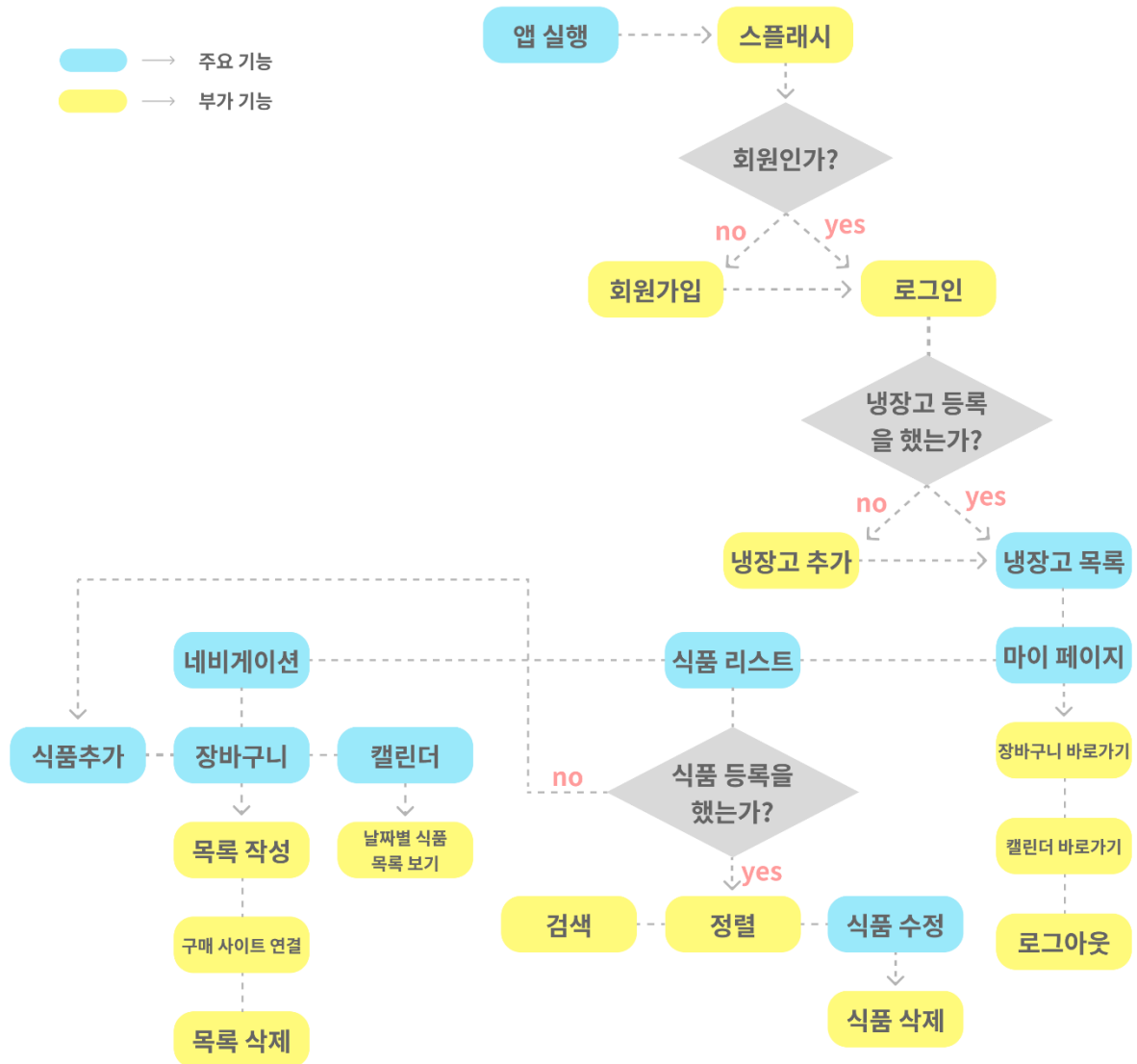
- 유스 케이스

- 사용자가 앱을 사용하고 관리자가 앱을 관리할 때 필요한 기능의 구조를 작성



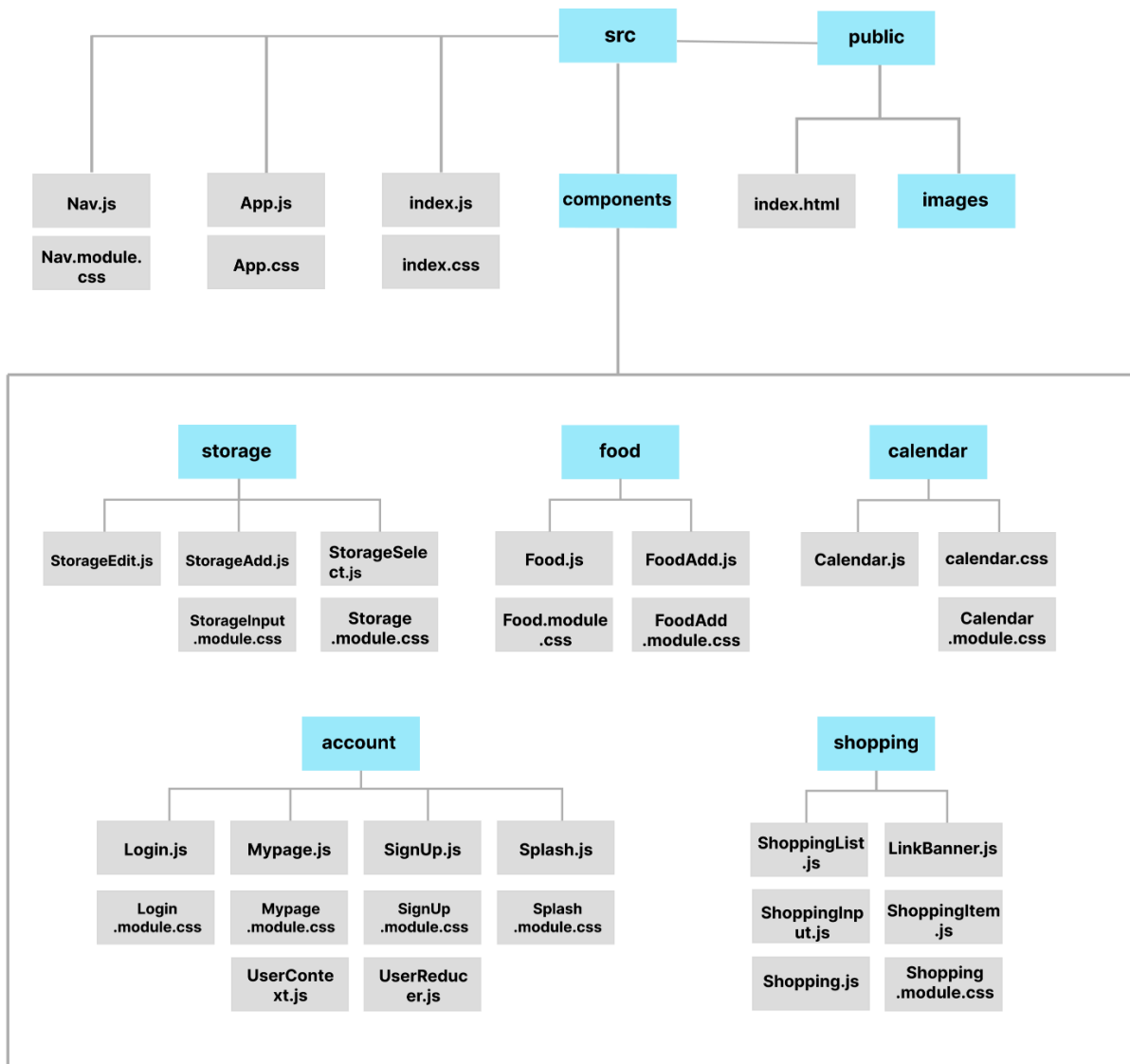
• 워크 플로우

- 사용자가 앱을 실행하고 사용하는 동안의 알고리즘을 구상하여 작성



- 디렉토리 구조

- 이미지는 최상위 폴더인 public에 두어 절대경로로 연결
- components 폴더 내부에 기능을 기준으로 생성한 하위 폴더별로 컴포넌트들과 css모듈을 함께 분류



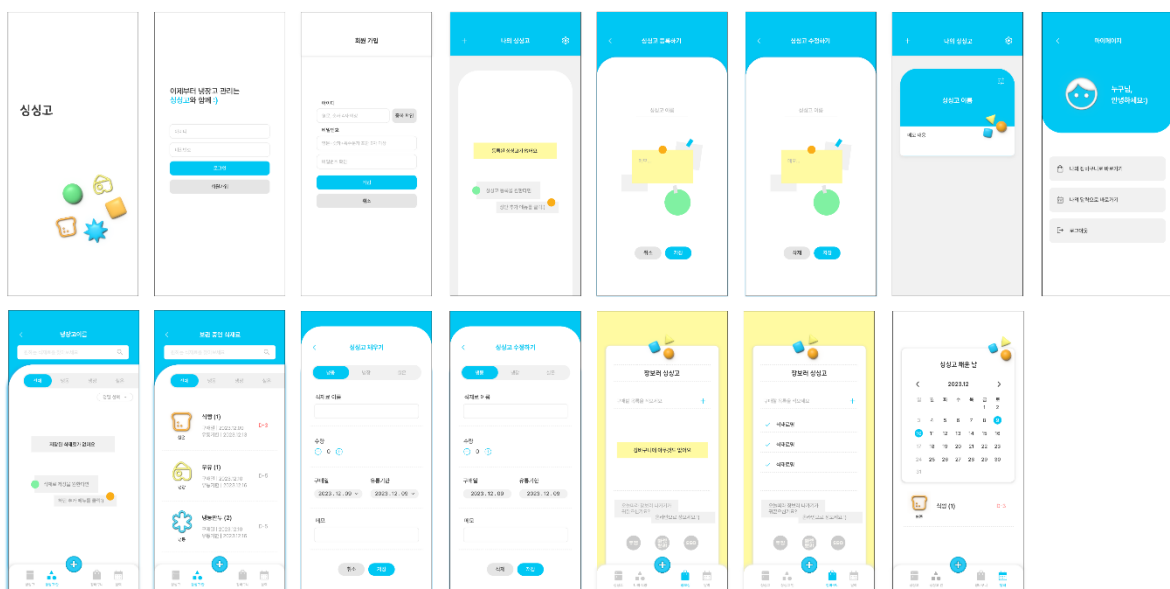
• 프로토타입 (초기)

기능을 중심으로 작성한 초기 프로토타입



• 프로토타입 (완성)

수정을 거쳐 정해진 기능에 디자인을 입혀 작성한 프로토타입



- CSS variable 가이드

폰트		
로그인/회원가입	로그인 타이틀	SemiBold(1.9rem)
	로그인, 회원가입 인풋	Medium(1.2rem)
	로그인 버튼	
	아이디, 비번 txt	Medium(1.25rem)
냉장고 등록 (추가/수정)	추가/수정- 메모 인풋	Regular(1.25rem)
	추가/수정-이름 등록 인풋	
	저장/삭제 버튼	Medium(1rem)
냉장고 목록	냉장고 이름	Bold(1.6rem)
	메모 내용	Regular(1.25rem)
식재료 목록 (보관 중인 식재료)	검색창	
	카테고리 선택	
	정렬	Regular(1.25rem)
	식재료 이름	SemiBold(1.5rem)
	구매일/소비기한	Regular(1.25rem)
	디데이	Medium(1.3rem)
	보관 위치	SemiBold(1.1rem)
	카테고리 선택	Regular(1.25rem)
식재료 등록 (상상고 채우기 /수정하기)	이름, 수량, 구매일, 유통기한, 메모	Medium(1.4rem)
	구매일, 유통기한 날짜선택	
	저장/삭제 버튼	Medium(1rem)
쇼핑리스트 (장보러 상상고)	추가 인풋	Medium(1.25rem)
	추가된 목록	Medium(1rem)
캘린더	년/월	SemiBold(1.4rem)
	일/요일	Medium(1.25rem)
	식재료 이름	SemiBold(1.5rem)
	디데이	Medium(1rem)
	보관 위치	SemiBold(1.1rem)
하단 메뉴	메뉴 이름	Medium(0.9rem)
공통	저장된 목록이 없다고 표시되는 박스 부분 (냉장고 목록/식재료 목록 /쇼핑리스트)	SemiBold(1.25rem)
	각 페이지 타이틀 (로그인,쇼핑리스트, 캘린더 제외)	Bold(1.6rem)

웹 폰트 연결
1. 프리텐다드 (본문)
@font-face {
font-family: 'Pretendard-Regular';
src: url('https://cdn.jsdelivr.net/gh/Project-Noonnu/noonfonts_2107@1.1/Pretendard-Regular.woff') format('woff');
font-weight: 400;
font-style: normal;
}
2. 소요단풍체 (로고, 제목)
@font-face {
font-family: 'SOYOMapleBoldTTF';
src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/noonfonts_2310@1.0/SOYOMapleBoldTTF.woff2') format('woff2');
font-weight: 700;
font-style: normal;
}

색상	
메인	#00C8F4
포인트1	#FFFB3A
포인트2	#FBAE17
포인트3	#6AE589
디데이 임박	#FF8080
그레이1	#E6E6E6
그레이2	#B3B3B3
폰트컬러	#3d3d3d

• UI 디자인

실제 많이 사용되고 있는 여러 앱들의 UI를 참고하여 현재 트렌드에 맞추어 디자인

- 키워드

생기 있는. 싱그러운. 맑은. 즐거운.

- 컨셉

둥근 형태를 메인으로 사용하여 시각적으로 편안하고 귀여운 컨셉

- 일러스트

냉장고에 메모를 붙이는 마그넷 형태에서 착안한 아이콘 스타일의 일러스트

- UX

사용자가 가이드를 보지 않고도 앱을 사용하기에 어려움이 없도록 직관적이고 단순

한 UI를 사용하여 사용자 편의성을 최대화

버튼 위치는 사용자가 인식하기 용이하게 배치하고 버튼 사이즈, 입력 폼 사이즈 등은 작동하기 편리한 사이즈로 선정

적당한 여백을 주어 앱을 시각적으로 편안하게 볼 수 있도록 함

테두리는 실선이 아닌 그림자로 자연스럽게 처리하여 딱딱한 느낌을 주지 않도록 함

• 간략한 코딩 가이드

- 클래스 네임은 쉽게 알아볼 수 있고 누구나 쉽게 납득 가능한 이름으로 작성한다. 네 단어 이상의 사용을 지양하고 카멜 표기법으로 작성함
- Const 선언 방식을 사용함
- 백틱이 필요한 경우를 제외, 따옴표는 큰 따옴표를 사용함
- 이벤트 핸들러 함수를 사용함
- Prettier를 사용하여 코드를 정리하여 작성함
- 설명이 필요한 부분은 주석을 통해 작성함
- 화살표 함수를 사용하여 작성함
- 들여쓰기는 space 2로 통일함
- 파일명은 각 페이지의 기능을 기준으로 작성함
- 코드에 대한 설명이 필요한 부분은 주석으로 작성함
- css는 모듈로 작성하며 공통부분은 app.css에 전역으로 작성함
- 미디어 사이즈는 844x390으로 고정함
-

• 주요 기능 및 명세

1) 공통

데이터는 localStorage로 저장하고 저장된 데이터를 불러오는 방식으로 관리 (추후 mongodb 사용 예정)

React Hook인 'useState'와 'useEffect' 사용하여 상태관리

React-router-dom의 'useNavigate'와 'link'로 페이지 경로 이동

폼 작성 모달창이나 곳곳의 버튼 기능 이벤트는 이벤트 핸들러로 설정

2) 로그인

아이디 중복확인: JS의 if문을 사용하여 구현

비밀번호 조합 조건 설정: 정규표현식 사용하여 구현

3) 냉장고 선택

등록한 냉장고 별 데이터 분류는 아직 구현하지 못하였지만 mongodb 사용하여 구현 예정

4) 식료품 리스트

카테고리/ 검색: JS의 filter 메서드 사용하여 구현

정렬: Js localCompare함수 Sort함수 사용하여 구현

5) 장바구니

추가: uuid 아이디 생성 패키지를 이용하여 아이디를 부여하여 사용자의 입력내용을 추가하고 배열로 반환된 내용을 JS의 map으로 리스트를 보여줌

삭제: JS filter를 사용하여 특정 id의 내용을 보여주지 않는 방식으로 삭제

6) 캘린더

캘린더 플러그인 (full calendar)을 설치하고 사용하여 날짜 선택 시 그 날짜에 해당하는 리스트 내용을 보여줌

• 문제점 및 해결 방안

- 1) 보관중인 식재료 리스트를 새로고침 하면 등록했던 싱싱고와 식품들 데이터가 초기화 되는 문제 발생

➔ 해결 : 로컬 스토리지를 이용하여 로컬에 데이터를 저장하여 데이터가 삭제되지 않도록함

- 2) 캘린더 기능 구현 시 캘린더를 통해 리스트를 불러오는 방법에 대한 고민과 거듭된 시행착오를 거침

➔ 해결 : full-calendar 플러그인을 사용하여 플러그인이 가지고 있는 날짜에 이벤트를 띄우는 기능을 통해 구현

- 3) 아이템을 선택한 후 삭제 버튼을 누르면 선택한 아이템이 삭제되어야 하는데 랜덤의 아이템이 삭제됨

➔ uuid를 통해 아이템마다 고유의 id값을 가지게 한 후 삭제 함수에서 선택한 아이템의

id를 찾아 실행할 수 있도록 코드를 수정함

4) 디데이 기능 중 디데이 임박 시 폰트 색상이 붉은색 계열로 변경되는 기능을 구현할 계획이었음

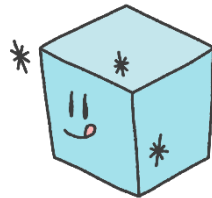
➔ If 함수를 사용하여 구현하려 했지만 적용을 결국 시키지 못함

➔ css 우선순위도 확인해 보고 console을 통해 받아오는 데이터도 확인해 보는 등 여러 고민을 하였지만 아직 해결하지 못한 아쉬움이 있음

➔ 좀 더 여러 경우를 고민하여 보완 예정

5) 초기 대자인에서 컬러대비가 약하고 일러스트, 버튼, 색상 등의 통일감이 없다는 피드백을 받음

➔ 기존에 선정했던 색상들의 채도를 높이고, 일러스트, 아이콘 등을 컨셉에 맞게 수정



초기 일러스트



현재 일러스트

8. 프로젝트 발표

- 청중 질의 내용

(1) mongoDB를 이용하여 데이터를 저장하였는지?

➔ Local storage에 저장하는 방식으로 작업

(2) 캘린더 모양은 직접 디자인했는지?

➔ 플러그인이 가지고 있는 기본 디자인에 css로 컨셉에 맞게 디자인을 추가하는 방식으로 함

(3) 아이디, 비밀번호 조합 조건 구현방법?

➔ 정규표현식을 사용하여 조건 설정함

- 피드백

(1) '기능들을 어떤 로직을 통해 구현했는지'의 내용 부족

- (2) 문제점과 해결방안에서 추상적인 표현보다는 구체적인 표현을 이용한 작성 필요
- (3) 냉장고 추가부분이나 로그인 부분의 데이터 관리를 추후 mongoDB를 이용하여 보완 요망
- (4) 발표시간이 규정시간인 15분보다 짧았음

9. 프로젝트 완료 리뷰

아쉬운 점

mongoDB를 이용하여 데이터를 저장하는 작업도 하고싶었지만 능력과 시간의 부족으로 구현하지 못함

백엔드쪽 서버관리부분도 좀 더 개인학습이 필요한 부분이라 팀 작업에 넣지 못한 아쉬움이 남음

잘한 점

팀원들과의 소통을 적극적으로 참여함

팀원들의 코드를 합치며 오류를 수정함

배운 점과 의견

디자인과 코드작성에 있어서 여러 팀원들이 동시에 작업하는 것이 쉬운 일은 아니었다. 각자가 원하는 방향과 컨셉도 조금씩 다를 수밖에 없다고 생각한다. 또 기능적인 코딩 작업을 하면서 팀원들 간에 수정한 코드가 중복되거나 프로토타입 구성 시 작업했던 부분과 다르게 만들어진 부분도 있었다. 전체 앱의 로직과 데이터가 연결되어 있는 작업이라 각자가 작성한 코드를 합치는 것도 쉽지 않았다. 하지만 팀원들의 소통으로 문제되는 부분을 하나하나 해결하고 수정하다 보니 오류가 하나씩 해결되고 기능이 정상적으로 작성되는 결과를 볼 수 있었다. 이렇게 문제를 함께 풀어가면서 각자가 느끼는 성취감이 팀원들을 이끄는 원동력이었다고 생각한다. 디자인적으로 창작하는 어려움도, 기능구현 면에서도 학습이 부족하여 느낀 한계도 있었지만 배운 점이 더 많은 프로젝트였다고 생각한다.