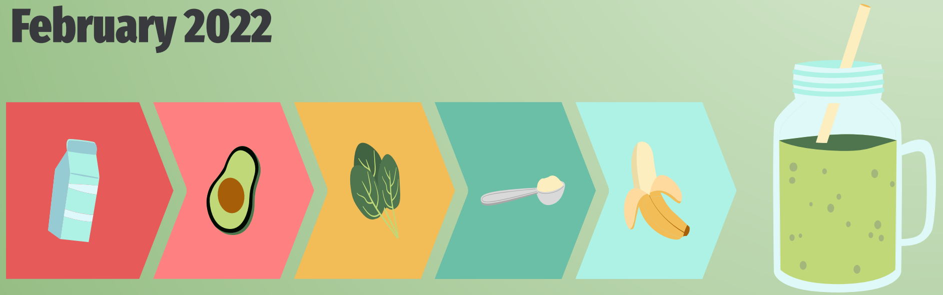
The Recipe Tree

What you want, what you need, we track your favorite recipes!

February 2022



Ben Allegrezza - Database Administrator
Dane Somdahl - Lead Developer
Jordan Talbot - Lead Developer
Jordon Paynter - SCRUM Master
Rick Kaucher - Repository Owner

The Recipe Tree Introduction

What is it?
Recipe Collector/Tracker
Random Recipe Generator

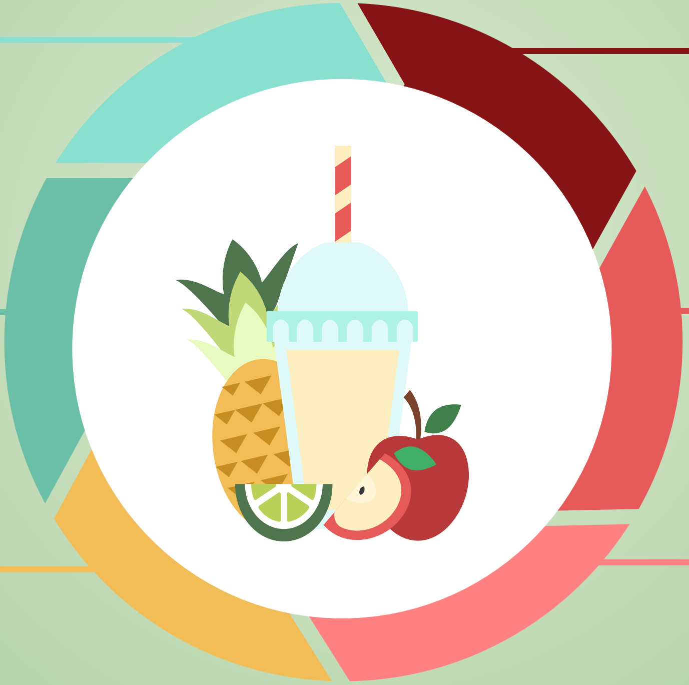
Users can...
Create
Read
Update
Delete

Why did we do it?
Food love,
we have major food love

Who did we do it for?
Other food lovers,
we are here for you

Planning Phase and The Database

- Plan, plan and plan some more.
- Data is King! Relationships matter!



01

User table that holds all user data

02

Recipe table that holds all recipe data

03

Ingredients table that holds all recipe ingredients and measuring units

04

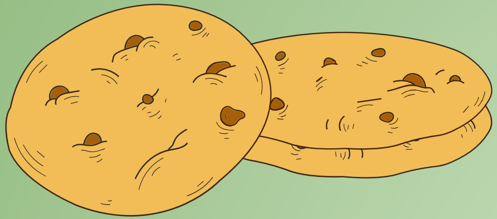
Category table that holds all recipe categories such as ethnicity, allergies, and diet

05

Favorite recipe table that holds user's favorite recipes

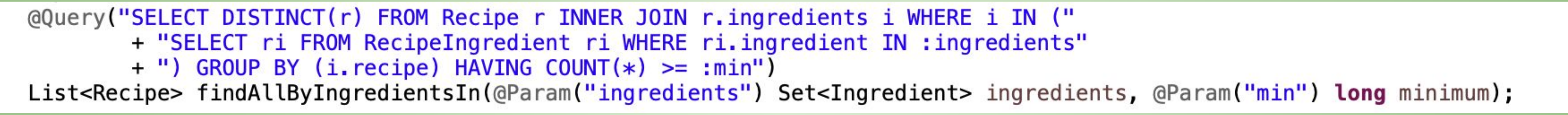
06

Recipe review table that holds user's recipe reviews



Recipe Search Development

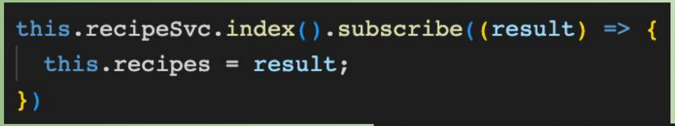
1. First revision was a backend query



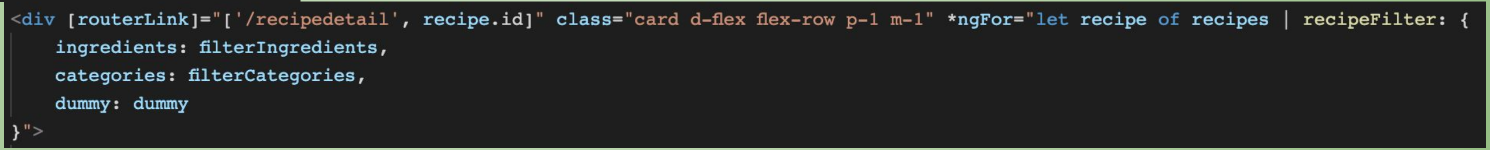
```
@Query("SELECT DISTINCT(r) FROM Recipe r INNER JOIN r.ingredients i WHERE i IN ("
        + "SELECT ri FROM RecipeIngredient ri WHERE ri.ingredient IN :ingredients"
        + ") GROUP BY (i.recipe) HAVING COUNT(*) >= :min")
List<Recipe> findAllByIngredientsIn(@Param("ingredients") Set<Ingredient> ingredients, @Param("min") long minimum);
```

- Complicated
- Fragile

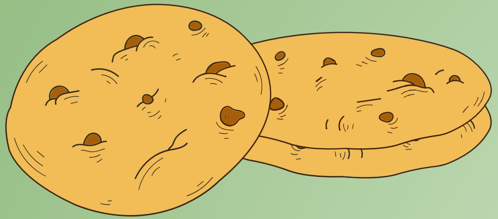
2. Second revision was a frontend query and filtering the results client side



```
this.recipeSvc.index().subscribe((result) => {
  this.recipes = result;
})
```



```
<div [routerLink]="['/recipedetail', recipe.id]" class="card d-flex flex-row p-1 m-1" *ngFor="let recipe of recipes | recipeFilter: {
    ingredients: filterIngredients,
    categories: filterCategories,
    dummy: dummy
}">
```



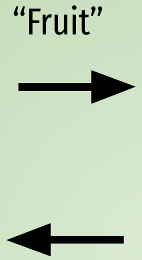
Recipe Search Filter

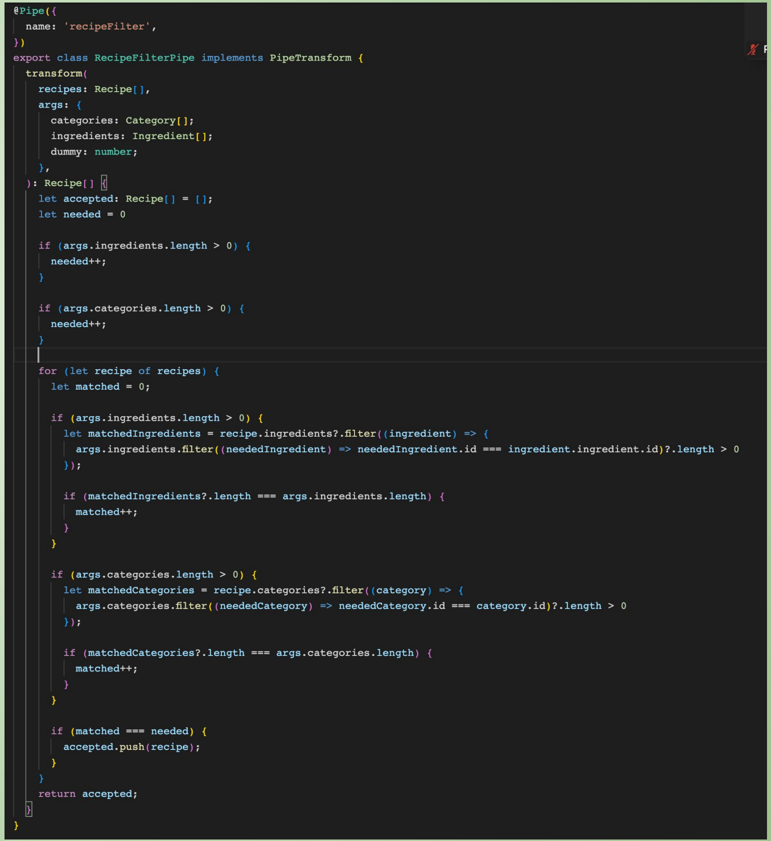
Pros

- Faster Testing/Iterations
- Finer Control of results
- Not as much confusing syntax
- Load moved to frontend

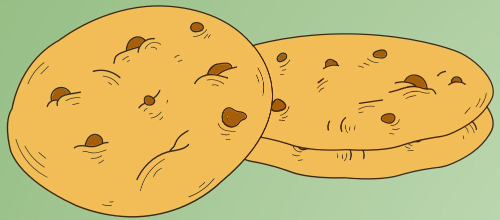
Cons

- More code





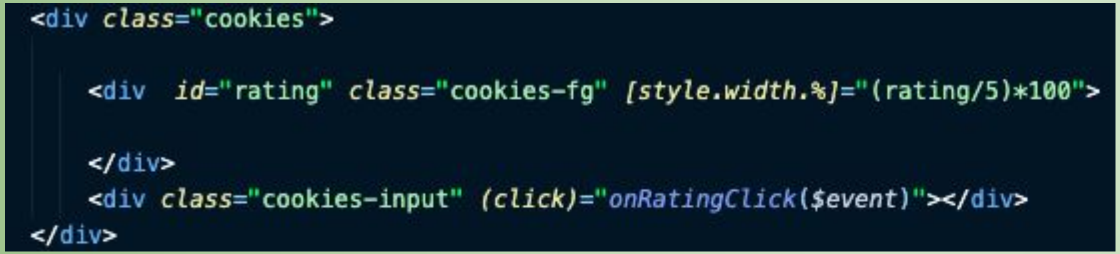
```
@Pipe({
  name: 'recipeFilter',
})
export class RecipeFilterPipe implements PipeTransform {
  transform(
    recipes: Recipe[],
    args: {
      categories: Category[];
      ingredients: Ingredient[];
      dummy: number;
    },
  ): Recipe[] {
    let accepted: Recipe[] = [];
    let needed = 0

    if (args.ingredients.length > 0) {
      needed++;
    }

    if (args.categories.length > 0) {
      needed++;
    }

    for (let recipe of recipes) {
      let matched = 0;

      if (args.ingredients.length > 0) {
        let matchedIngredients = recipe.ingredients?.filter((ingredient) => {
          args.ingredients.filter((neededIngredient) => neededIngredient.id === ingredient.ingredient.id)?.length > 0
        });

        if (matchedIngredients?.length === args.ingredients.length) {
          matched++;
        }
      }

      if (args.categories.length > 0) {
        let matchedCategories = recipe.categories?.filter((category) => {
          args.categories.filter((neededCategory) => neededCategory.id === category.id)?.length > 0
        });

        if (matchedCategories?.length === args.categories.length) {
          matched++;
        }
      }

      if (matched === needed) {
        accepted.push(recipe);
      }
    }
    return accepted;
  }
}
```



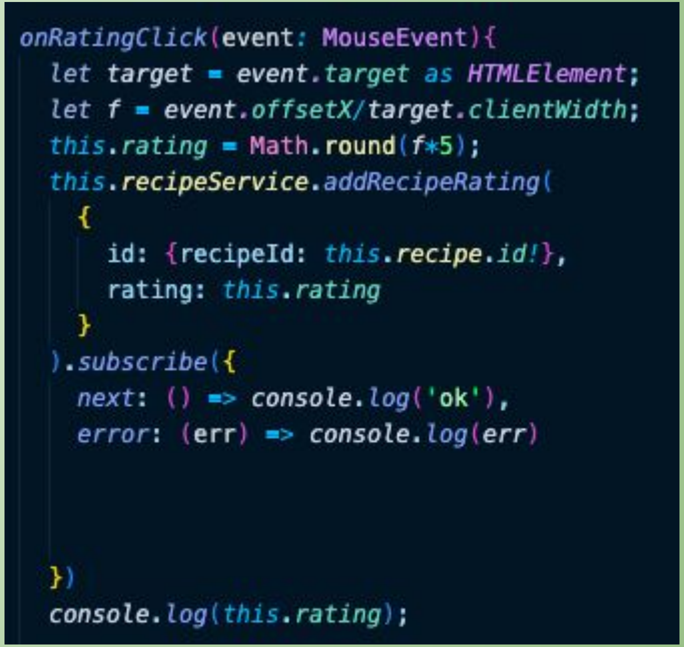
Recipe Rating HTML & Type Script

Image is in the css, not in the html

Foreground that changes with rating

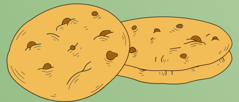


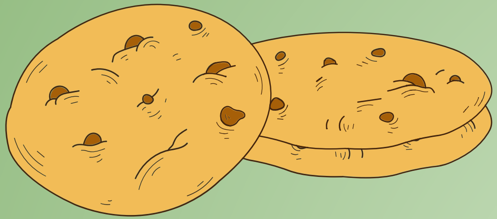
```
<div class="cookies">

    <div  id="rating" class="cookies-fg" [style.width.%]="(rating/5)*100">

    </div>
    <div class="cookies-input" (click)="onRatingClick($event)"></div>
</div>
```



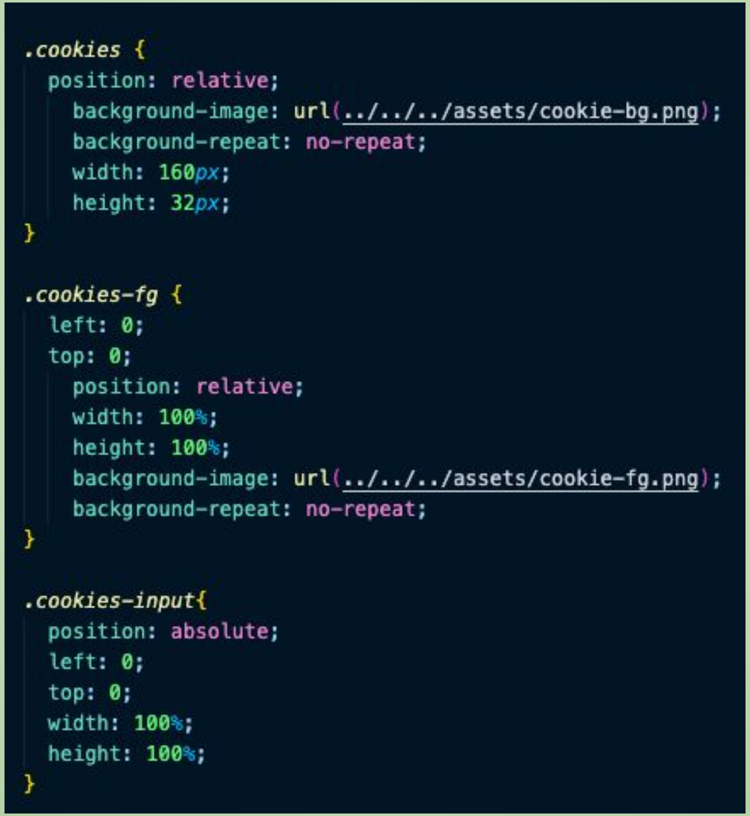
```
onRatingClick(event: MouseEvent){
  let target = event.target as HTMLElement;
  let f = event.offsetX/target.clientWidth;
  this.rating = Math.round(f*5);
  this.recipeService.addRecipeRating(
    {
      id: {recipeId: this.recipe.id!},
      rating: this.rating
    }
  ).subscribe({
    next: () => console.log('ok'),
    error: (err) => console.log(err)



  })
  console.log(this.rating);
```

When you click on a cookie it creates a new rating and saves it to your profile





Recipe Rating CSS

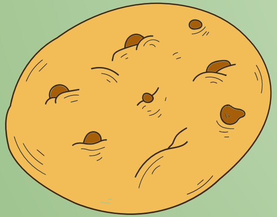


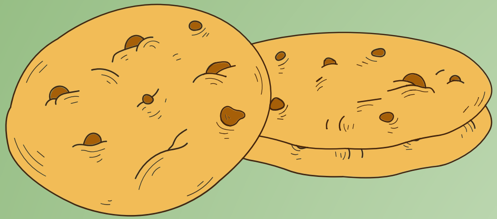
```
.cookies {
  position: relative;
    background-image: url(../../../assets/cookie-bg.png);
    background-repeat: no-repeat;
    width: 160px;
    height: 32px;
}

.cookies-fg {
  left: 0;
  top: 0;
    position: relative;
    width: 100%;
    height: 100%;
    background-image: url(../../../assets/cookie-fg.png);
    background-repeat: no-repeat;
}

.cookies-input{
  position: absolute;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
}
```

Image of cookies

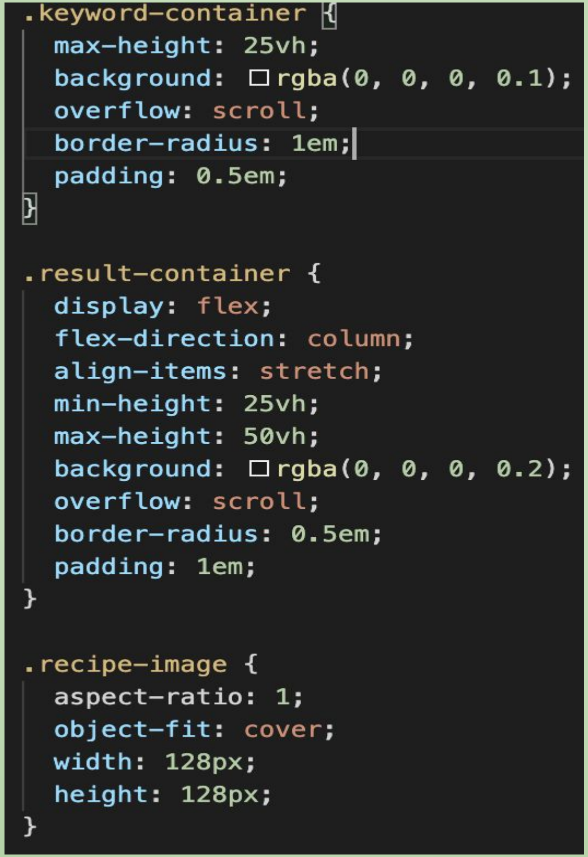
Foreground changes size based on rating

Clickable rating input field





Profile Page CSS

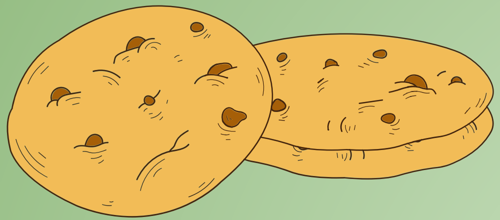


```
.keyword-container {
  max-height: 25vh;
  background: □rgba(0, 0, 0, 0.1);
  overflow: scroll;
  border-radius: 1em;
  padding: 0.5em;
}

.result-container {
  display: flex;
  flex-direction: column;
  align-items: stretch;
  min-height: 25vh;
  max-height: 50vh;
  background: □rgba(0, 0, 0, 0.2);
  overflow: scroll;
  border-radius: 0.5em;
  padding: 1em;
}

.recipe-image {
  aspect-ratio: 1;
  object-fit: cover;
  width: 128px;
  height: 128px;
}
```

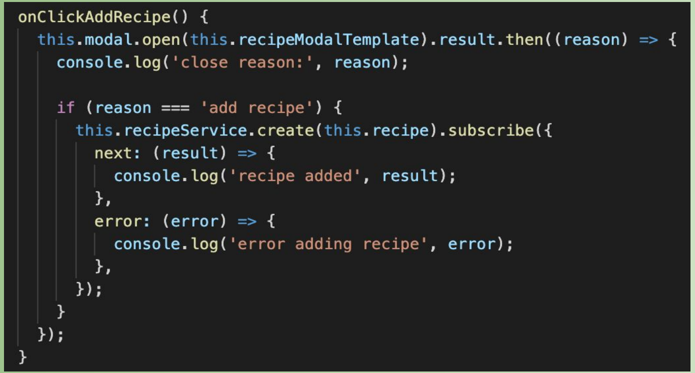
Populates keyword tags to search from

Populates recipe search results

Image of the recipe



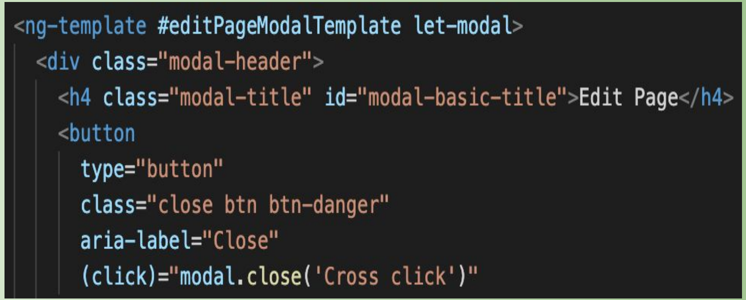
Profile Page HTML & Typescript



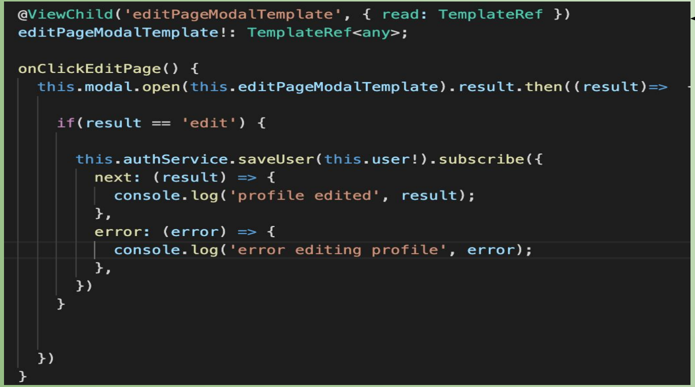
```
onClickAddRecipe() {
  this.modal.open(this.recipeModalTemplate).result.then((reason) => {
    console.log('close reason:', reason);

    if (reason === 'add recipe') {
      this.recipeService.create(this.recipe).subscribe({
        next: (result) => {
          console.log('recipe added', result);
        },
        error: (error) => {
          console.log('error adding recipe', error);
        },
      });
    }
  });
}
```

Implementation of the modal template in TS.

Modal template in HTML for Edit Page function



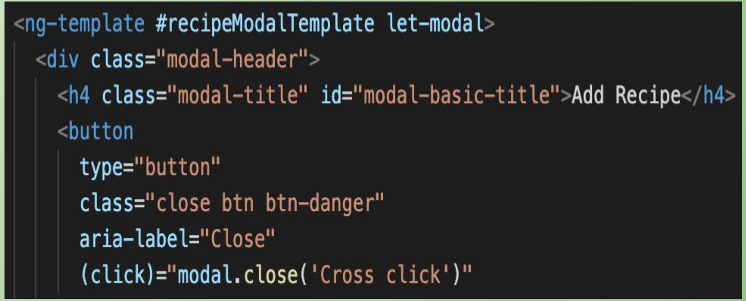
```
<ng-template #editPageModalTemplate let-modal>
  <div class="modal-header">
    <h4 class="modal-title" id="modal-basic-title">Edit Page</h4>
    <button
      type="button"
      class="close btn btn-danger"
      aria-label="Close"
      (click)="modal.close('Cross click')"
```



```
@ViewChild('editPageModalTemplate', { read: TemplateRef })
editPageModalTemplate!: TemplateRef<any>;

onClickEditPage() {
  this.modal.open(this.editPageModalTemplate).result.then((result)=>

    if(result == 'edit') {

      this.authService.saveUser(this.user!).subscribe({
        next: (result) => {
          console.log('profile edited', result);
        },
        error: (error) => {
          console.log('error editing profile', error);
        },
      })
    }


  })
}
```

@viewChild() makes the add recipe form pop-up on click.

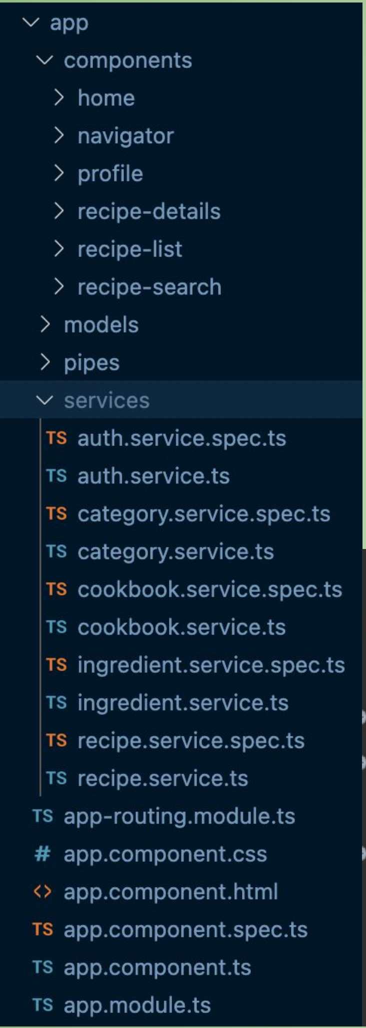
Modal template in HTML for Add Recipe function



```
<ng-template #recipeModalTemplate let-modal>
  <div class="modal-header">
    <h4 class="modal-title" id="modal-basic-title">Add Recipe</h4>
    <button
      type="button"
      class="close btn btn-danger"
      aria-label="Close"
      (click)="modal.close('Cross click')"
```

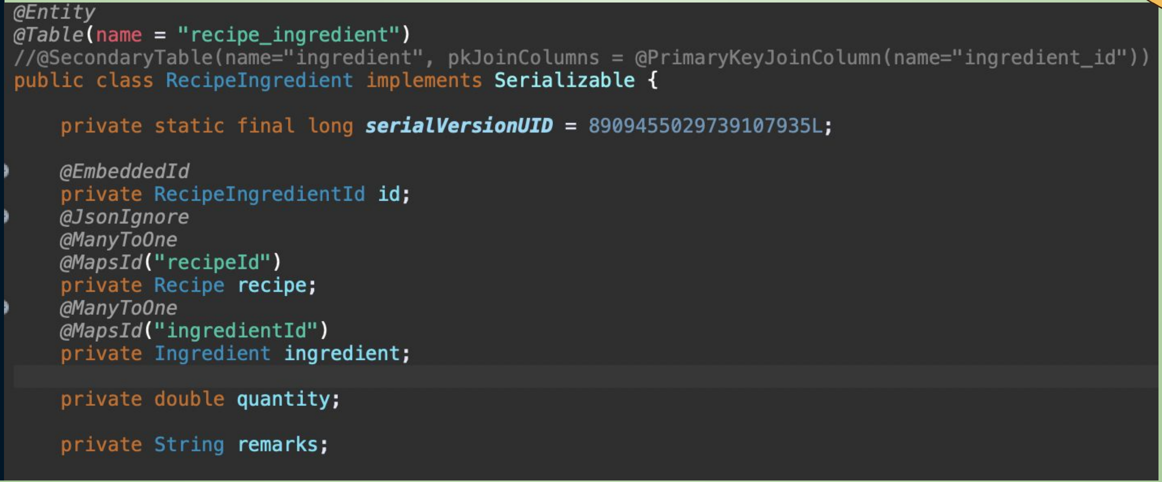
Challenging Recipes to Still Cook

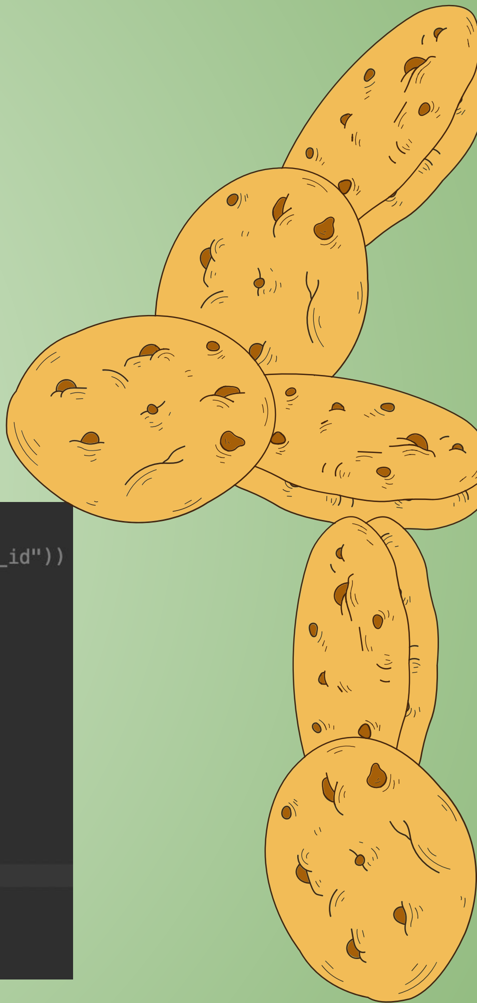
- We wanted to implement users being able to modify their own Cookbooks.
- Modify Diet Plan(s)
- Leave comments on Recipes
- Additional modifications to users profile page

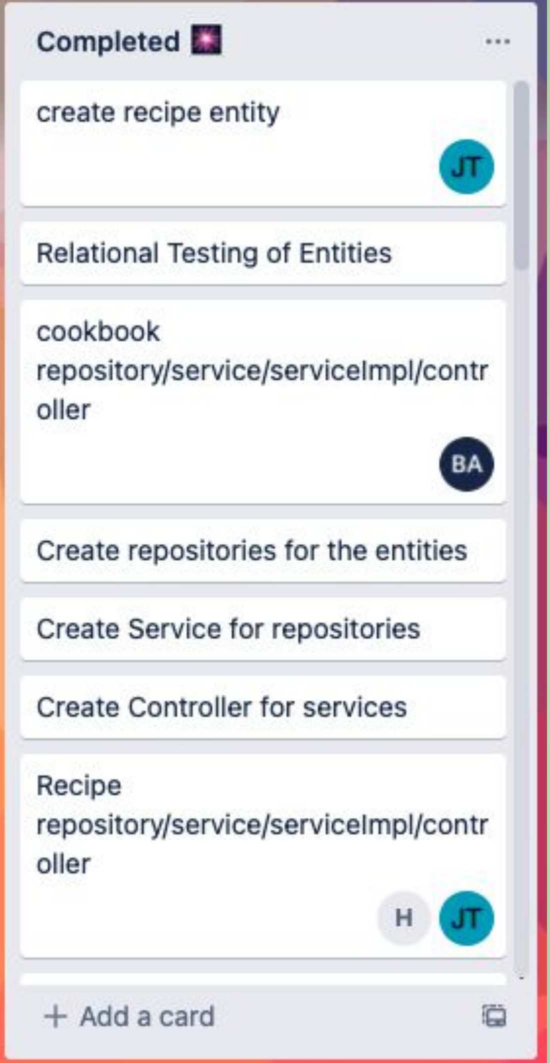

```
app
  components
    home
    navigator
    profile
    recipe-details
    recipe-list
    recipe-search
  models
  pipes
  services
    auth.service.spec.ts
    auth.service.ts
    category.service.spec.ts
    category.service.ts
    cookbook.service.spec.ts
    cookbook.service.ts
    ingredient.service.spec.ts
    ingredient.service.ts
    recipe.service.spec.ts
    recipe.service.ts
  app-routing.module.ts
  app.component.css
  app.component.html
  app.component.spec.ts
  app.component.ts
  app.module.ts
```

Challenges Over Well Done

- Team Size
- Composite Keys
- Deciding Whether front-end or back-end
- Communication
- Pair Programming


```java
@Entity
@Table(name = "recipe_ingredient")
//@SecondaryTable(name="ingredient", pkJoinColumns = @PrimaryKeyJoinColumn(name="ingredient_id"))
public class RecipeIngredient implements Serializable {

    private static final long serialVersionUID = 8909455029739107935L;

    @EmbeddedId
    private RecipeIngredientId id;
    @JsonIgnore
    @ManyToOne
    @MapsId("recipeId")
    private Recipe recipe;
    @ManyToOne
    @MapsId("ingredientId")
    private Ingredient ingredient;

    private double quantity;

    private String remarks;
```





Completed

create recipe entity

JT

Relational Testing of Entities

cookbook repository/service/serviceImpl/controller

BA

Create repositories for the entities

Create Service for repositories

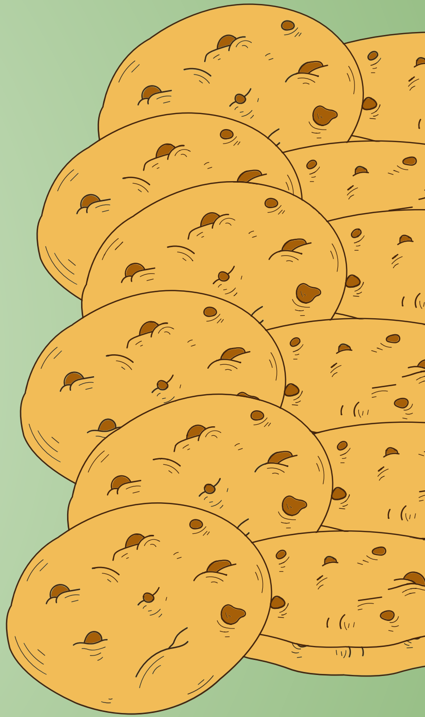
Create Controller for services

Recipe repository/service/serviceImpl/controller

H JT

Conclusion

- Meet Minimum viable product Requirements
- Worked out Solutions as a Team
- Helped each other
- Ready to tackle the next task



Questions

- What Is your Favorite Recipe?
- What do you prefer working on front end or back end?
- What Technology did you enjoy most?
- What Technology did you learn the most from the project?
- What would you do different now, if you started over.
- How did the team plan out this project?
- What advice would you give to yourself starting this course?