

profile

ProfileService

- profileRepository: ProfileRepository

+ add(name: String): void

+ remove(name: String): void

+ getBy(name: String): Optional<Profile>

+ getAll(): Collection<Profile>

+ setOpinionAbout(name: String, food: Food, opinion: Opinion): void

+ addToAvailable(name: String, food: Food): void

+ removeOpinionAbout(name: String, food: Food): void

+ markUnavailable(name: String, food: Food): void

recipe

MatchingService

- repository: RecipeRepository

+ getMatchingRecipeFor(id: UUID, profile: Profile): Collection<Recipe>

- compare(recipe: Recipe, comparing: Recipe, profile: Profile): double

- getOpinionScore(comapring: Recipe, profile: Profile): double

- getIntersectionOverUnion(recipe: Recipe, comparing: Recipe): double

- caontainsDealbreaker(comparing: Recipe, profile: Profile): boolean

RecipeService

- repository: RecipeRepository

+ refreshFrom(source: Datasource): void

+ getAll(): Collection<Recipe>

<<interface>>

Datasource

+ getRecipes(): Collection<Recipe>