
Feed Me!

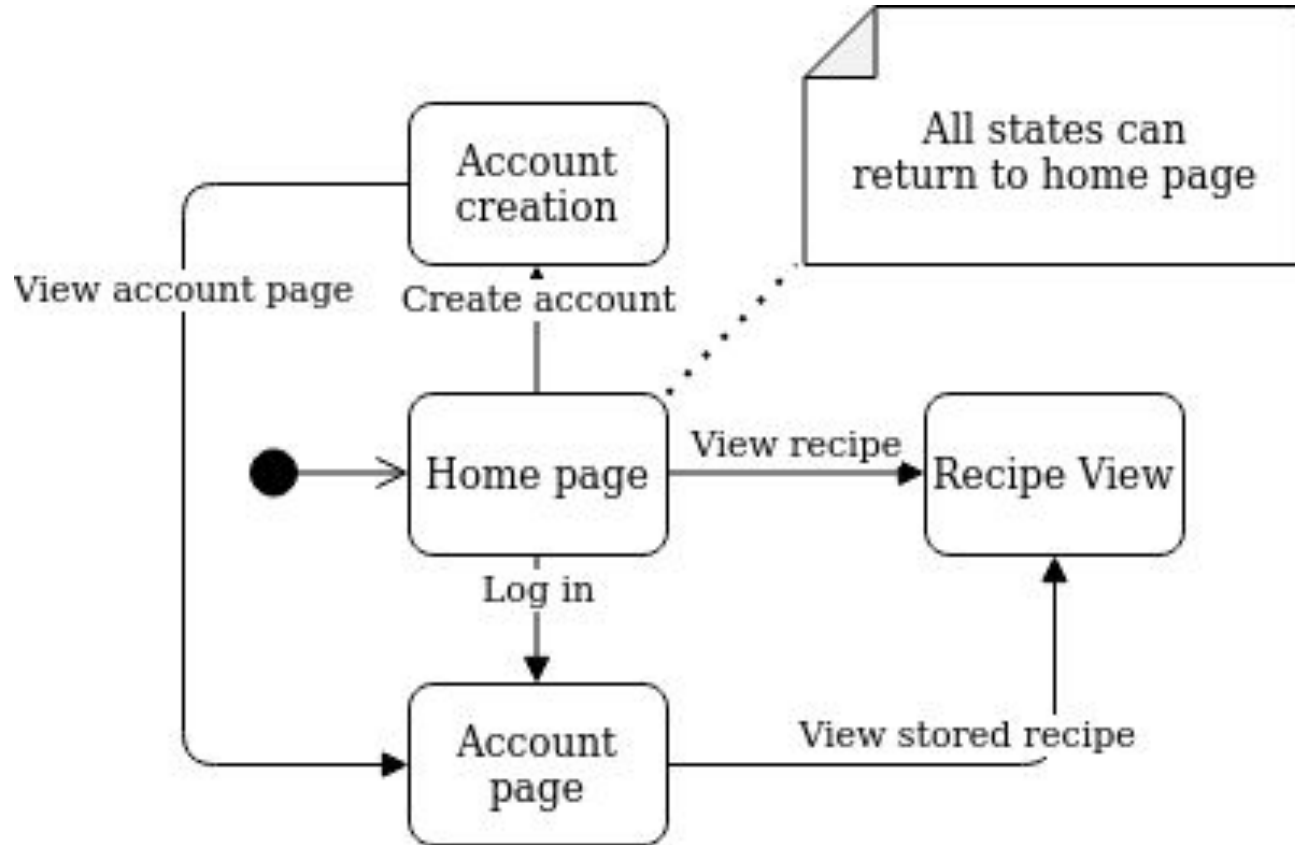
Team 22



Tech stack

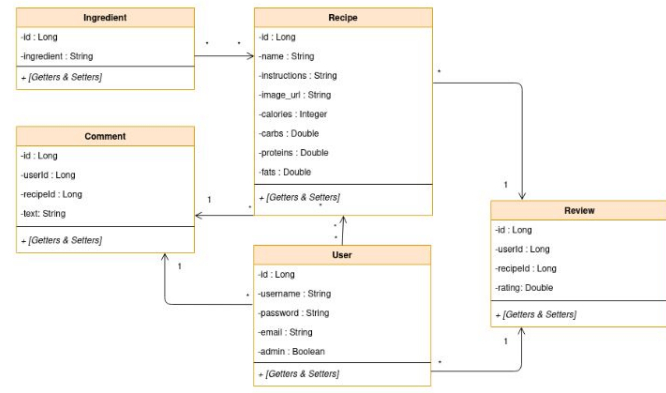
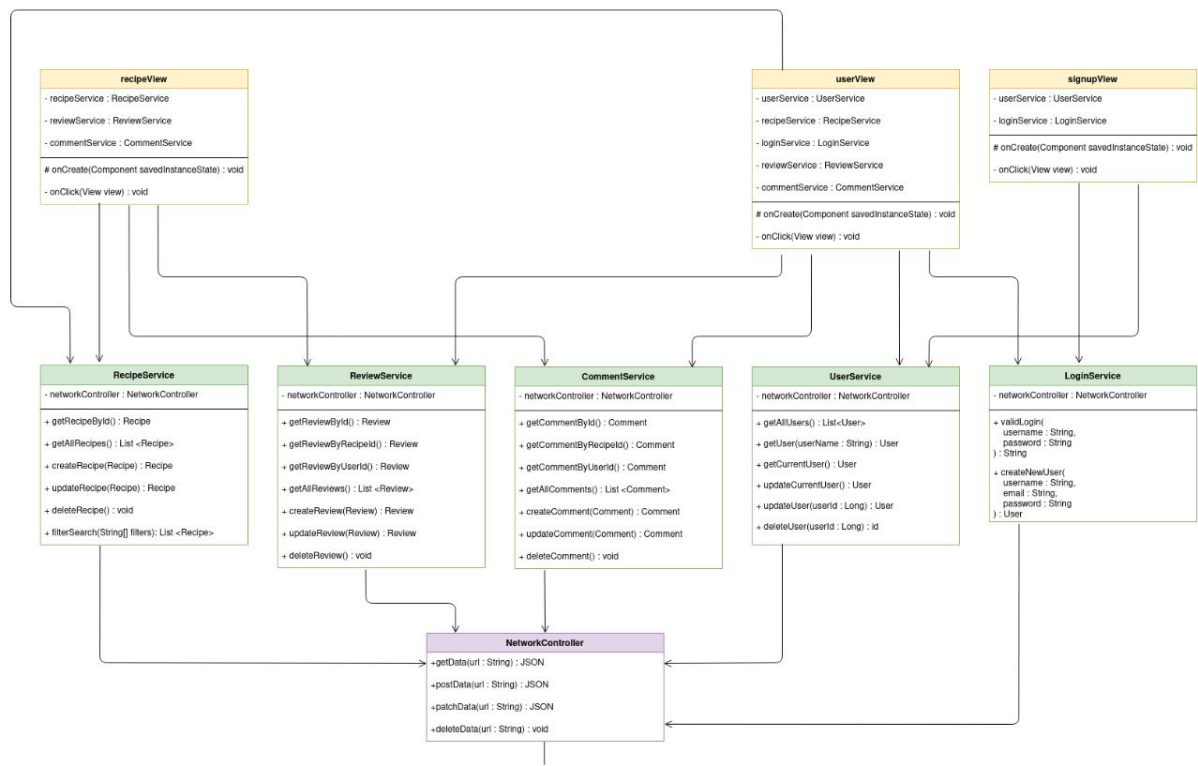
- Java Spring back-end service built on top of PostgreSQL
 - VueJS front-end for client-side rendering
 - Communicate using HTTP requests and responses
 - Structure of communication is “RESTful” and uses JSON
 - Simple GitHub git repository for version control
-

Front-end state management

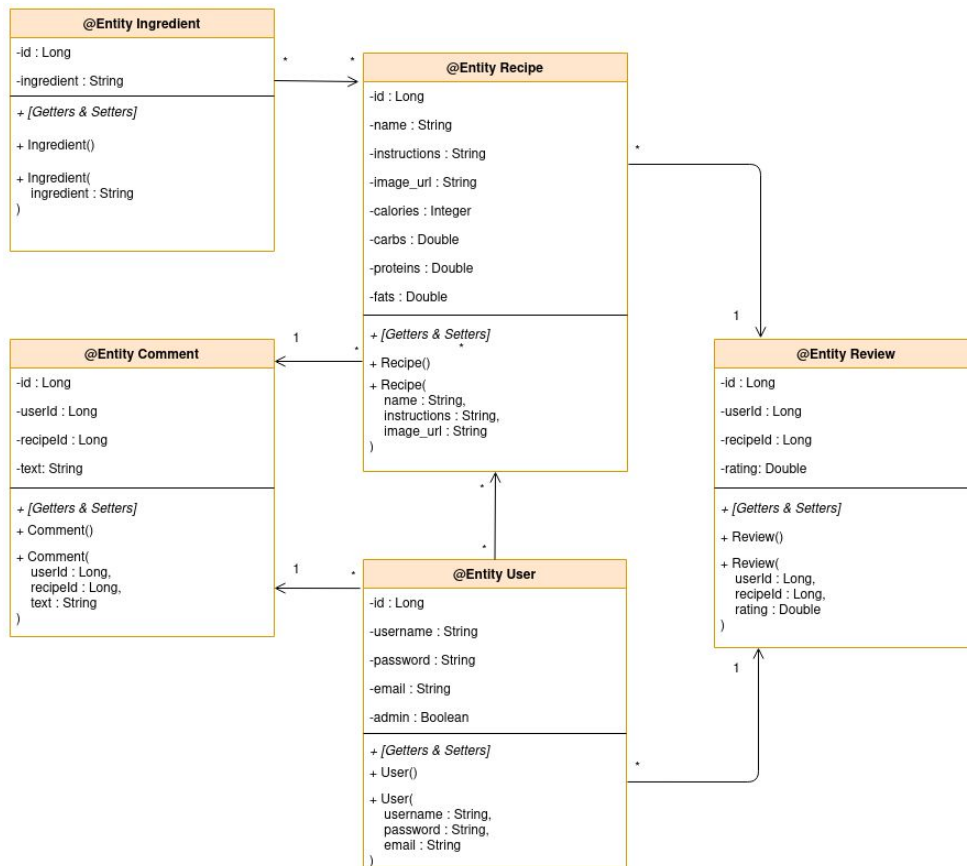


**The design and
changes to it ...**

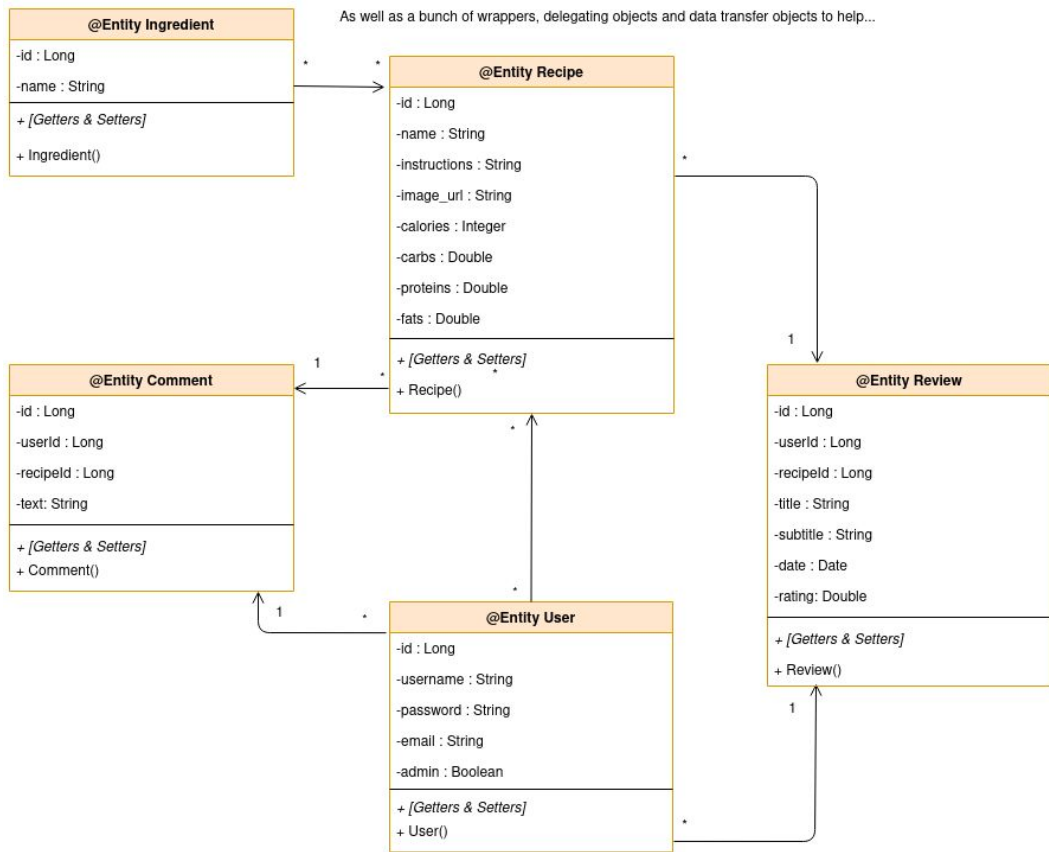
Frontend



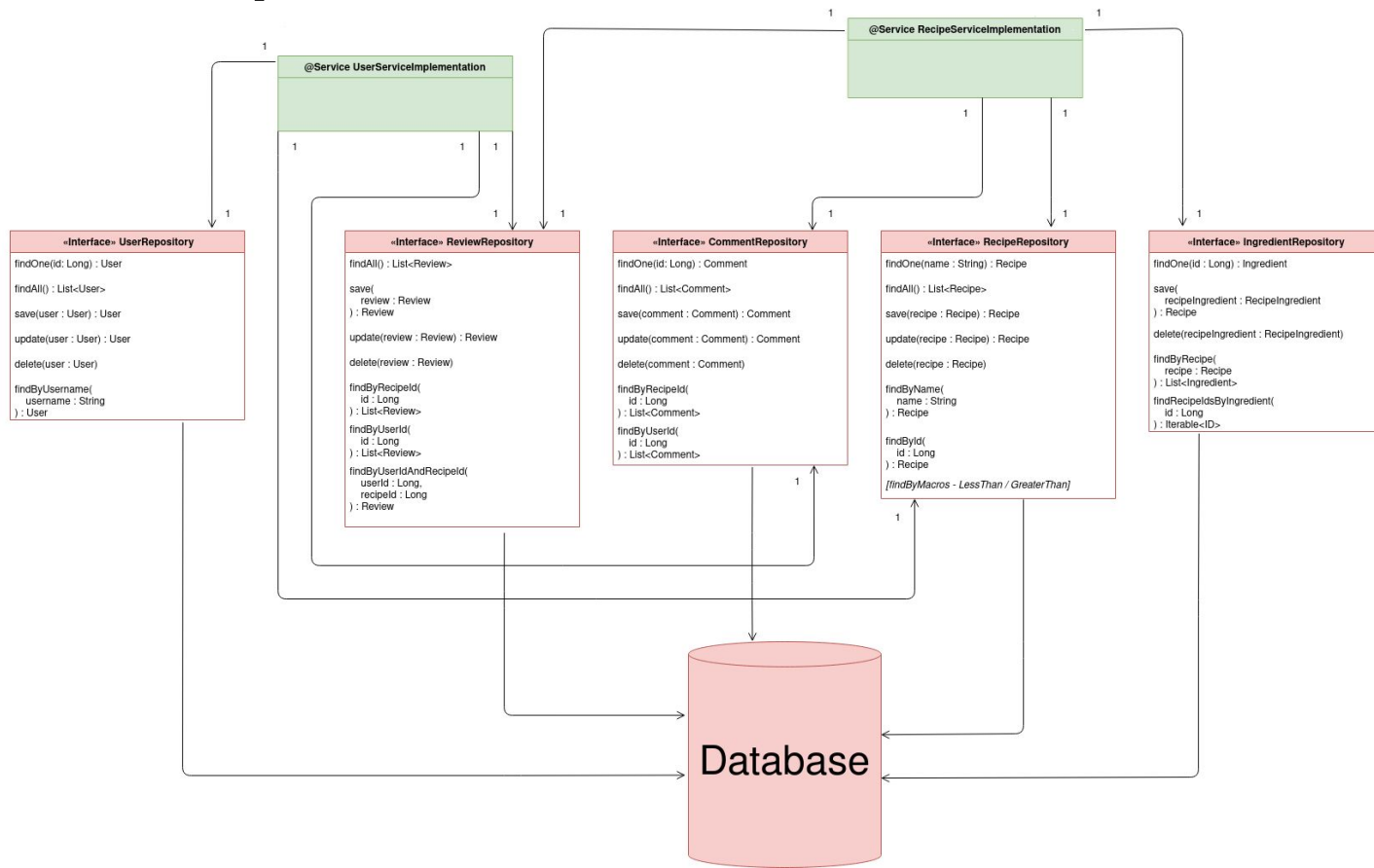
Back-end entities - Planned



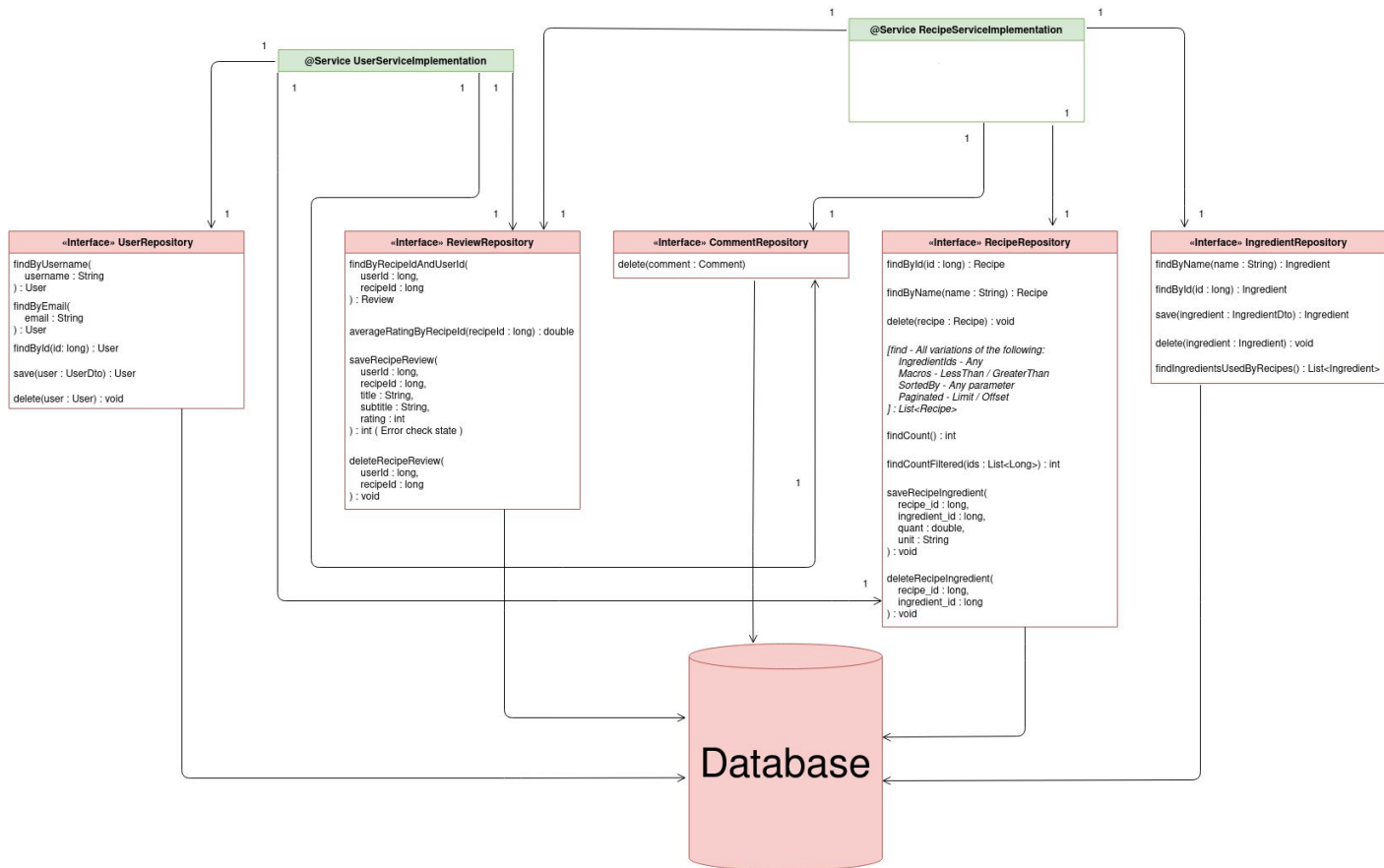
Back-end entities - Actual



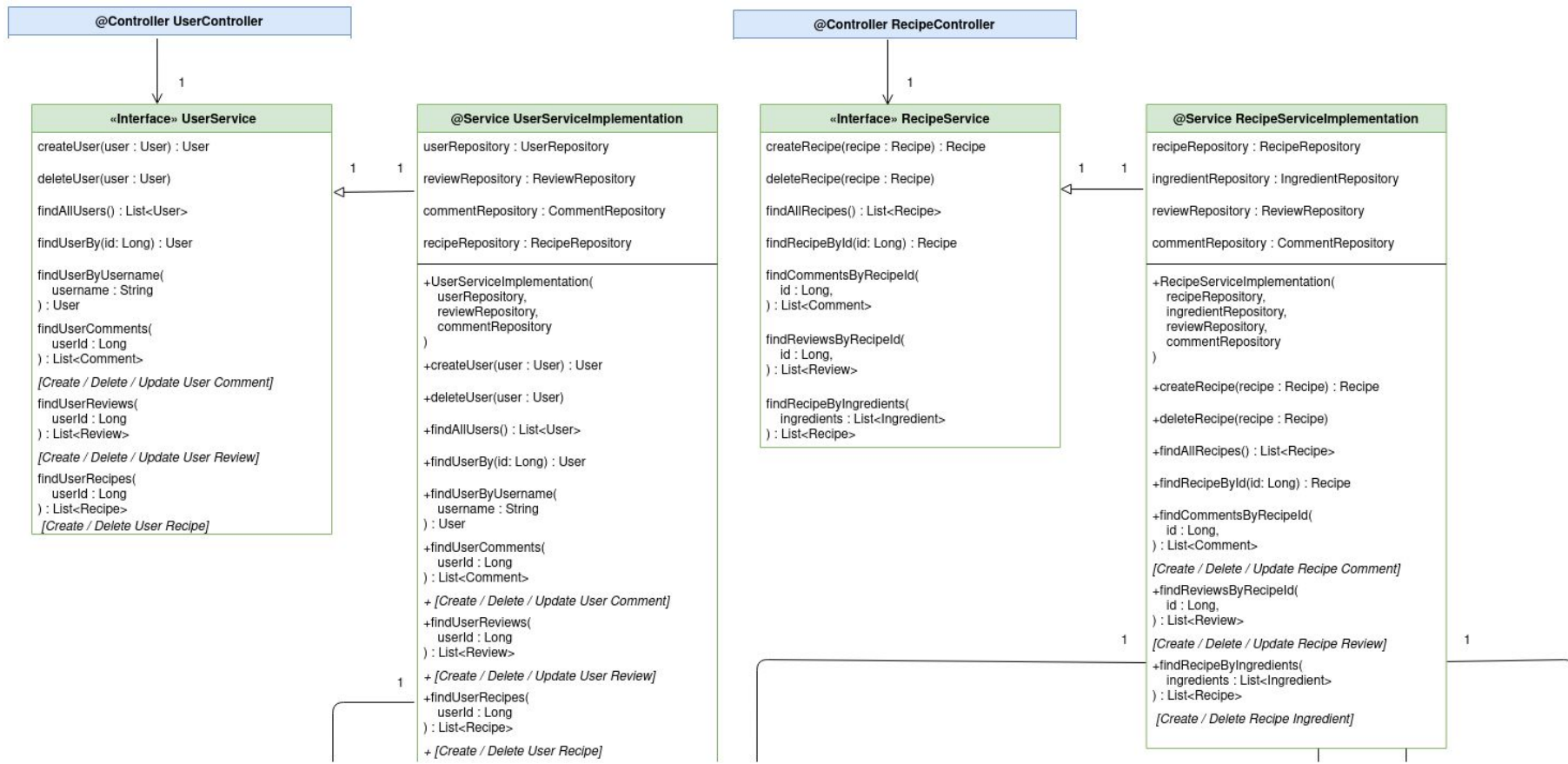
Back-end repositories - Planned



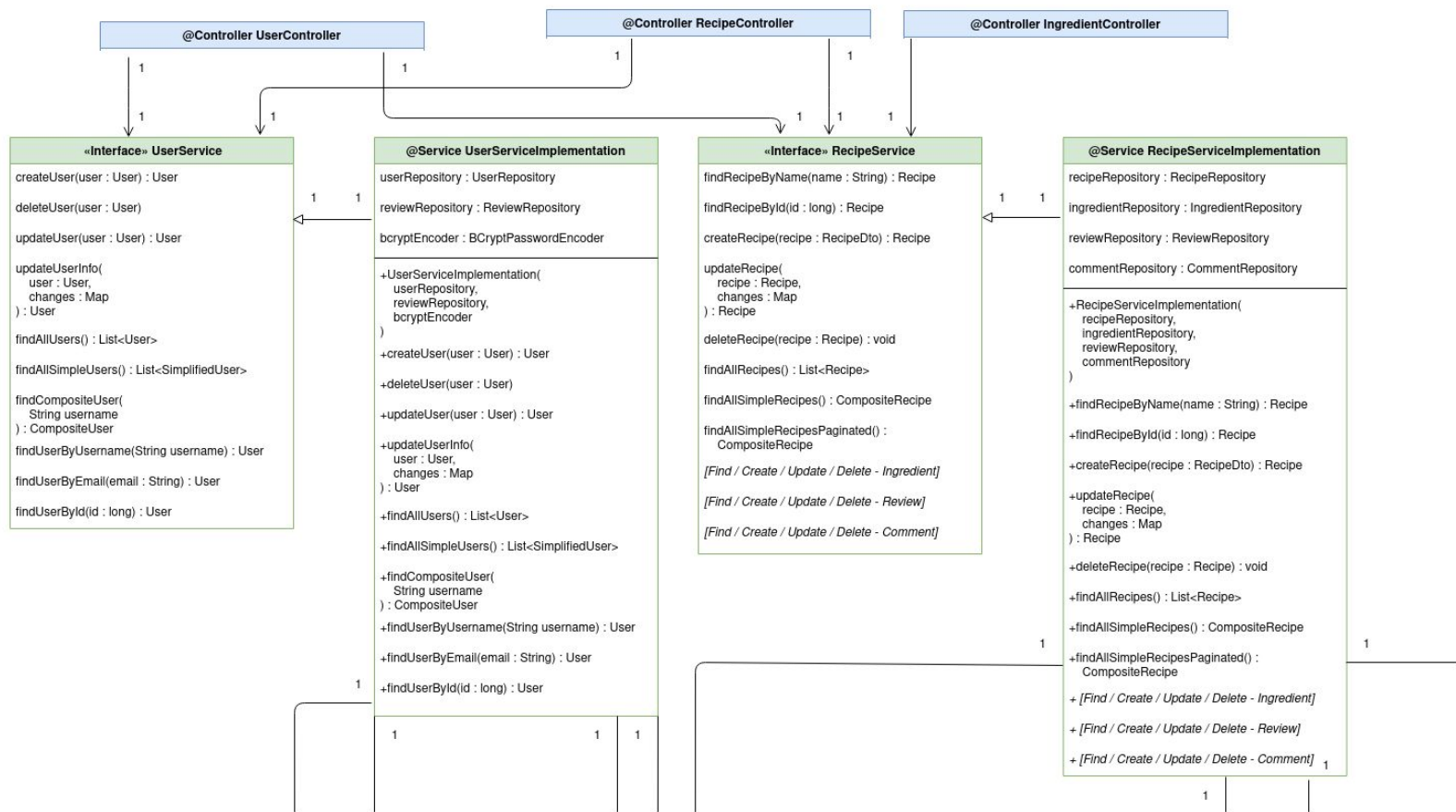
Back-end repositories - Actual



Back-end services - Planned



Back-end services - Actual



Back-end controllers - Planned

Backend

HTTP

@Controller UserController
userService : UserService
+UserController(userService : UserService)
+findAllUsers() : String
+createUser(model : Model) : String
+validateUser(model : Model) : String
+findCurrentUserInfo() : String
+editCurrentUser(model : Model) : String
+findUserById() : String
+updateUserById(model : Model) : String
+deleteUserById() : String

@Controller RecipeController
recipeService : RecipeService
+RecipeController(recipeService : RecipeService)
+findAllRecipes() : String
+createRecipe(model : Model) : String
+findRecipeInfoById() : String
+updateRecipeById(model : Model) : String
+deleteRecipeById() : String

1

Back-end controllers - Actual Backend

HTTP

@Controller UserController
recipeService : RecipeService
userService : UserService
jwtTokenUtil : TokenProvider
authenticationManager : AuthenticationManager
+UserController(...)
+findUserById(id : PathVariable) : ResponseEntity
+findAllUsers() : ResponseEntity
+findAllSimpleUsers() : ResponseEntity
+validateUser(loginUser : RequestBody) : ResponseEntity
+createUser(user : UserDto) : ResponseEntity
+findCurrentUserInfo() : ResponseEntity
+updateCurrentUserInfo(changes : RequestBody) : ResponseEntity
+ [Find / Create / Update / Delete - UserRecipe]

@Controller RecipeController
recipeService : RecipeService
userService : UserService
+RecipeController(recipeService : RecipeService, userService : UserService)
+findAllRecipes(params : RequestParam[]) : ResponseEntity
+findRecipeInfoById(id : PathVariable) : ResponseEntity
+createRecipe(recipe : RecipeDto) : ResponseEntity
+updateRecipeById(id : PathVariable, changes : RequestBody (Map)) : ResponseEntity
+deleteRecipeById(id : PathVariable) : ResponseEntity
+ [Find / Create / Update / Delete - RecipeIngredient]
+ [Find / Create / Update / Delete - RecipeIngredient]
+ [Find / Create / Update / Delete - UserComment]

@Controller IngredientController
recipeService : RecipeService
+IngredientController(recipeService : RecipeService)
+getAllUsedIngredients() : ResponseEntity
+getAllIngredients() : ResponseEntity
+createIngredient(ingredient : IngredientDto) : ResponseEntity
+deleteIngredient(id : long) : ResponseEntity

1

1

1

Thoughts, Problems and Challenges

- Spring seems to attempt to provide sensible defaults ...
 - But often fails ...
- Tools to work with JSON are comparatively lacking
- JPQL was more limited than expected ...
 - But native queries immediately lose access to some tools
- Repositories and Responses working directly with Entities can cause issues



GitHub link:

github.com/anjrv/HBV501