Poniższy opis jest odwzorowaniem klas i interjesów jakie będę napisane w języku PHP, z wyjątkiem deklaracji typów zmiennych gdyż język PHP do wersji 7 nie wspiera twardego podziału na typy zmiennych. Jednakże jeżeli projekt powstanie w języku Java i w technologii Java Spring oraz Java Spring MVC to nazwy i zawartości interfejsów mogą się różnić od tych opisanych w dokumencie. Wynika to z faktu, że ta technologia jeszcze nie jest nam znana i nie wiemy czy uda się odwzorować opisane w niej interfejsy.

```
struct Ingredient{
      int id;
      string name;
      List<string> units;
};

struct Category{
      int id;
      string name;
};

struct Comment{
      int id;
      int recipeId;
      string comment;
};
```

```
struct Recipe{
      int id;
      string name;
      string description;
      List<Category> categories;
      List<Ingredient, int> ingredients;
      List<string, string> customIngredients;
      List<Comment> comments;
      int prepareTime;
      int costsServing;
      int calories;
      date publishDate;
      date lastCookedDate;
      boolean favorite;
};

exception SQLError {
    attribute string message;
};

exception BDHandleError {
    attribute string message;
};

interface IngredientsList {
      boolean addIngredient(in Ingredient ingredient);
      boolean deleteIngredient(in Ingredient ingredient);
      boolean editIngredient(in Ingredient ingredient);
      List<Ingredient> getIngredients();
};
```

```
interface CategoriesList{

      boolean addCategory(in Category category);

      boolean deleteCategory(in Category category);

      boolean editCategory(in Category category);

      List<Category> getCategories();

};


interface RecipesList{

      boolean addRecipe(in Recipe recipe);

      boolean deleteRecipe(in Recipe recipe);

      boolean editRecipe(in Recipe recipe);

      boolean addToFavorites(in Recipe recipe);

      boolean deleteFromFavorites(in Recipe recipe);

      boolean updateLastCookedDate(in Recipe recipe, in Date date);

      List<Recipe> getRecipes();

};


Interface RecipeCategories{

      void addCategory(in Category category);

      void addCategories(in List<Category> categories);

      void deleteCategory(in Category category);

      void deleteCategories(in List<Category> categories);

};


Interface RecipeIngredients{

      void addIngredient(in Ingredient ingredient);

      void addIngredients(in List<Ingredient> ingredients);

      void deleteIngredient(in Ingredient ingredient);

      void deleteIngredients(in List<Ingredient> ingredients);

};
```

```
interface DataBase{
      readonly attribute string host;
      readonly attribute string port;
      readonly attribute string user;
      readonly attribute string password;
      readonly attribute string base;
      void connect() raises (BDHandleError);
      void disconnect() raises (BDHandleError);
};

interface DataBaseIngredientsList : DataBase {
      boolean addIngredient(in Ingredient ingredient) raises (SQLError);
      boolean deleteIngredient(in Ingredient ingredient) raises (SQLError);
      boolean editIngredient(in Ingredient ingredient) raises (SQLError);
      List<Ingredient> getIngredients() raises (SQLError);
};

interface DataBaseCategoriesList : DataBase{
      boolean addCategory(in Category category) raises (SQLError);
      boolean deleteCategory(in Category category) raises (SQLError);
      boolean editCategory(in Category category) raises (SQLError);
      List<Category> getCategories() raises (SQLError);
};

interface DataBaseRecipesList : DataBase{
      boolean addRecipe(in Recipe recipe) raises (SQLError);
      boolean deleteRecipe(in Recipe recipe) raises (SQLError);
      boolean editRecipe(in Recipe recipe) raises (SQLError);
      boolean addToFavorites(in Recipe recipe) raises (SQLError);
      boolean deleteFromFavorites(in Recipe recipe) raises (SQLError);
      boolean updateLastCookedDate(in Recipe recipe, in Date date) raises
(SQLError);
```

```
	List<Recipe> getRecipes() raises (SQLError);
};
```