Recipe book project

Authors:
Clemens Berger 120040982
Valerii Drobiazgho 10348123

Idea:

There are a lot of different recipes on the Internet, and there is no problem to cook something according to an already written recipe. But sometimes a good idea comes to mind and the dish turns out to be very tasty. How not to lose a fleeting divine intent?

To do this, we have developed a *Recipe book*, where everyone can add their own recipe, step-by-step instructions with taking a photo of the resulting dish.

Recipe book is an offline application, with the additional option for the user to add recipes himself.

Description:

In the main menu of the application, the user gets access to a list of already saved recipes. Three menu buttons at the bottom of the screen: *add new recipe*, *recipes*, *settings*. In the upper right menu of the screen there is an additional menu, where the user can select all recipes, favorites, dishes from the various categories: *vegetarian menu*, *vegan menu* and *omnivore menu*.

By clicking on a recipe, the user gets detailed *description* of the dish, *instructions*, list of ingredients and an overview of the weight and nutritional value with the calories.

While looking at the description of the recipe, the user has an opportunity to add the recipe to favorites, as well as to remove the recipe from the *favorites* category.
The user moves from one recipe to another simply by swiping.

In order to get access to a new recipe form the user need to click at the *add new recipe* button. The filling form is simple and intuitive for the user. Each template is accompanied by an inscription. When typing the first letter of an ingredient the user gets a choice from a list of saved ingredients.

When adding a new recipe, the user has an opportunity to take a photo of the cooked dish and save it along with the recipe. After saving, this recipe will appear in the list of existing recipes.

By pressing the *settings* button, the user can switch to *Dark mode* and exit it at any time.

Pressing the *recipes* button returns the user to the main menu.

Goals:

Master the skills of developing Android mobile applications.

Work with Android Studio.

Improve Java programming skills.

Learn Android design. The final project is Android mobile app *Recipe book*.

Final goal: offline cookbook with calories display and the option to create your own recipes, with the ability to select an ingredient from the existing list and not the ability to add a new ingredient.

Application structure:

1. Database
The project uses 3 tables: *Recipe*, *Ingredient*, *RecipeIngredient*.

The *Ingredient* table includes ingredient *name*, *calories per 100 grams* and *id*.

The *Recipe* table includes the *recipe name*, *instruction*, *description*, *foodtype recipe*.
It also contains the *favorite* label, *urlpath to the image* and *id*.

The *RecipeIngredient* table is the link between the *Ingredient* table and the *Recipe* table. The table consists of *ingredientId*, *recipeId*, *ingredient weight in grams* and *id.* Therefore this table allows us to get a list of ingredients for each recipe, taking into account the weight.

2. Description of classes

Basemodel - a class so that all models inherit from it the available attributes.

Ingredient model inherits from *Basemodel* and describes the data sets that describe an ingredient: *calories*, *name.*

Recipe model bring all data of recipe: the *name*, *instruction,desciption, foodtype, picture.*

RecipeIngredient model is used for connecting a recipe with its ingredients. Describe a data of *RecipeIngredient*: *ingredientId*, *recipeId*, *ingredient weight*, *ingredient.*

RecipeIngredientIngredient merges a *ingredient* with the *recipeIngredient* object.

FileSystemHandle can create new files in system and stores them.

IngredientRepository is used for operations in the *Ingredient* table.

RecipeRepository is used for operations in the *Recipe* and *RecipeIngredient* tables.

In the *ViewModel* there is an interface between a fragment and a database. The fragment Database makes operation via *Viewmodel*.

3. Description of fragments

Basefragment is a fragment, but all fragments inherit from *Basefragment*.

Inputfragment add new recipes. Ingredients can only be selected from the database list. There is no way to add new ingredients. Pictures are taken directly on the camera. Saved as file in system, add as path in database.

In *RecipePagerFragment* is *RecipePagerAdapter.* At the *RecipePagerAdapter* we don't use position from recipe, but id from recipes.

RecipeListAdapter pushes data from the database to show recipes.

RecipeDetailFragment shows the details of the selected recipe with description, list of ingredients and weight of each one.

A list of all recipes is displayed in *RecipeListFragment*, different recipes can be filtered here by category, recipes can also be marked and then favorited or deleted.

In *RecipeListFragment* is also *RecipeSelectionObserver*, which makes possibility to mark.

RecipePagerfragment allows user to swipe between recipes and recipes are loaded via a *DetailFragment*, there is an option to favorite.

SettingFragment allows you to enter and exit *Darkmode*, there is possibile make zoom on recipe pager.

4. Known bugs

When you switch several times from the recipe to the main menu and back to the recipe, sometimes come a bug with the output of recipe details on the screen.

There is also a problem when several recipes are added to favorites and the user decides to forget the first recipe from the list, this recipe does not disappear instantly and if you add the current removed recipe to favorites back, a bug appears.

There isn't that much test data.