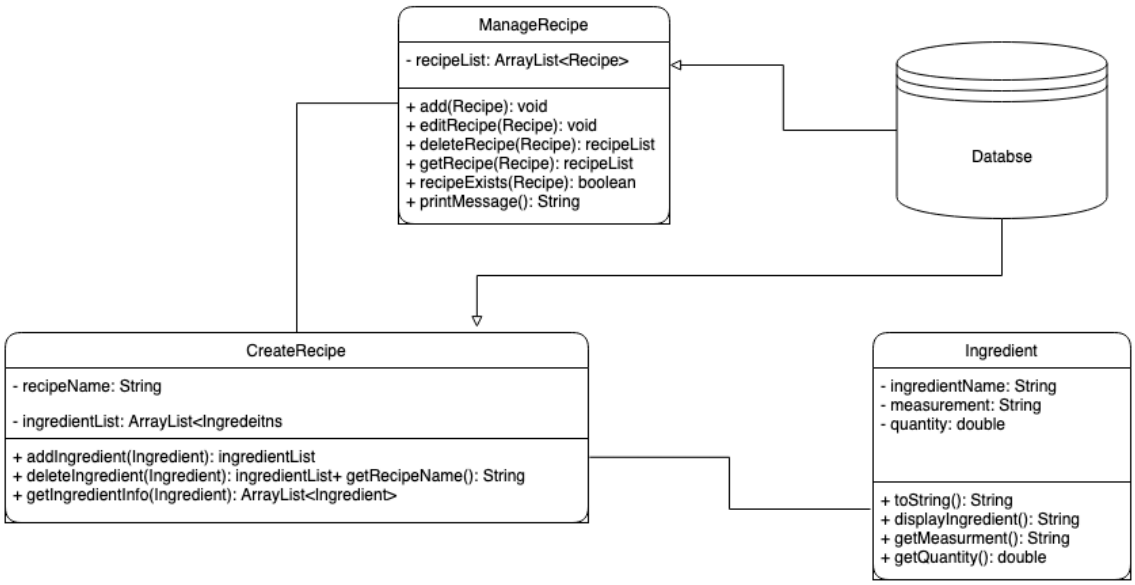
Saurabh Chauhan
Team Name: Projekt Fork
ITSC 4155 Section 5:30pm

# Class Diagram:

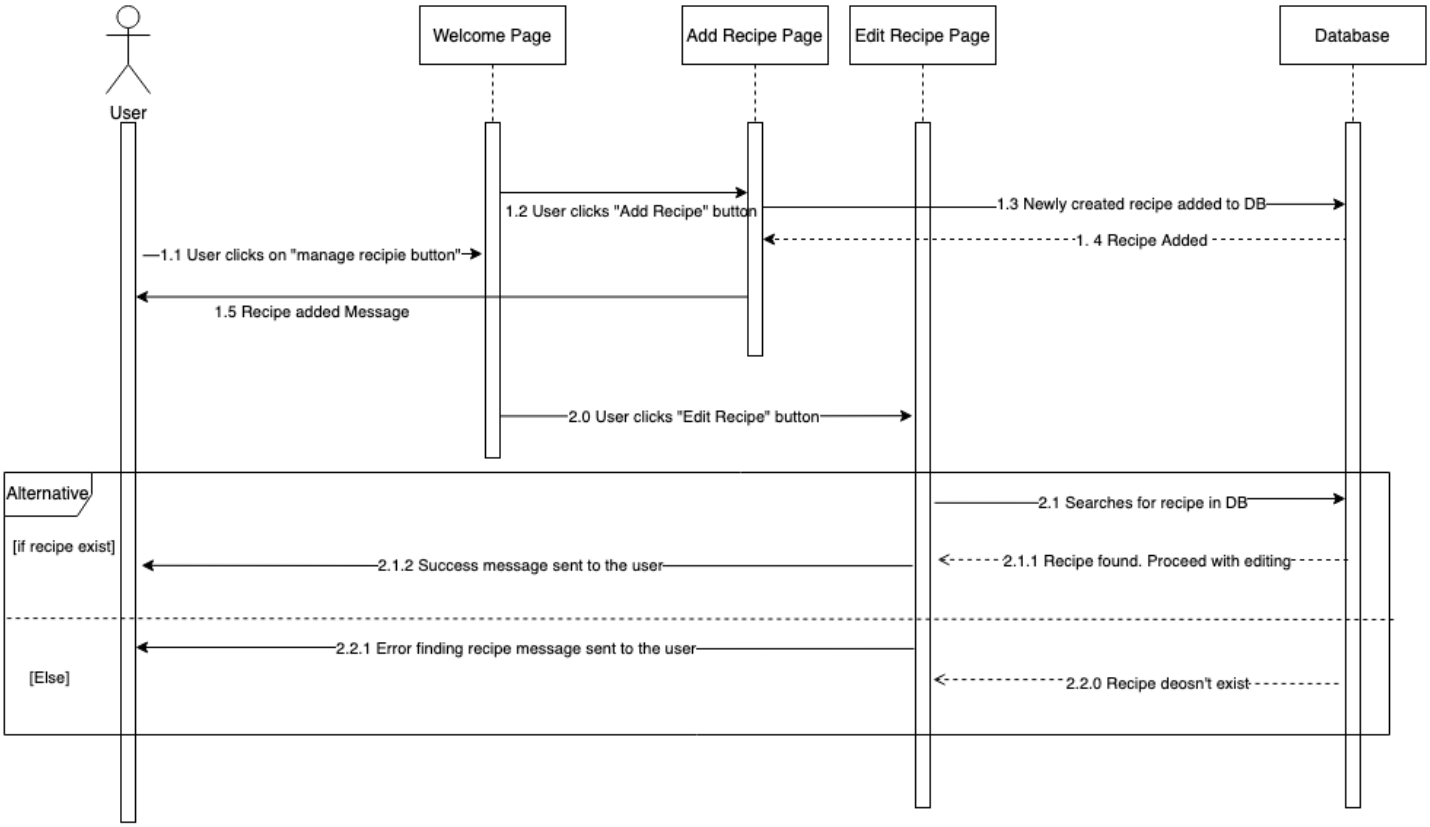



In this class diagram I have created 3 classes and a database system. The description of each class are as follows:

1. Create Recipe Class:
    a. This class is responsible for creating a new recipe from scratch. The pertains to our user story: “As a user, I would like to create new recipes”. It contains two fields: the name of the recipe and an array list of ingredients that will be needed to make the recipe. Following that, I made four functions that I will be implementing to make “creating a recipe” work correctly.
        i. addIngredients: Takes an argument of class Ingredient and stores the values in a list named IngredientList.
        ii. deleteIngredients: Also takes class incredient as an argument and deletes the individual ingredient from the list.
        iii. getIngredientInfo: Takes indredient as an argument and returns that ingredients information from the Ingredient class
        iv. displayMessage(): This method is responsible for display any error message that occur when adding/creating new recipies
2. Manage Recipe Class:
    a. This class is responsible for adding or editing a class and saving the final to a list.

        i. addRecipe: This function is responsible for adding a recipe and saving it in a list named recipeList.
        ii. editRecipe: This function is responsible for editing a recipe and saving the changes in a list named recipeList.
        iii. deleteRecipe: This function is responsible for deleting a recipe from the recipeList array list.
        iv. getRecipe: This function takes the name of the recipe as an argument and gets all the details of the list.
        v. recipeExists: This is a Boolean function that takes the name of the recipe as an argument and looks it up in the database or array list. If the recipe exists, it returns a success message. Else, it returns false.
3. Ingredient Class:
    a. This class is responsible for adding new ingredient and formatting them with the correct specs. This class has three attributes.
        i. Name of the ingredient (a string format)
        ii. The measurement of the ingredient (i.e., pounds, kilos, etc.)
        iii. The quantity of each ingredient (i.e., 1.5 pounds, 2 kilos, etc.)
    b. This class has four functions as follows:
        i. toString: This function is responsible for any conversion of integers/double numbers to a string so it can be printed correctly
        ii. displayIngredient: This function is responsible for printing out the ingredients correctly with all its attributes (i.e., name, measurements, etc.)
        iii. getMeasurments: This function is responsible for getting the measurement and returning the value as a string
        iv. getQuantity: This function is responsible for getting the quantity of each ingredient and returning the value as a double value (decimal).
4. The database system is going to be keeping a track of the data storage.

# Sequence Diagram:




This sequence diagram shows the sequence of what and how the user can react to with our app. This pertains to a couple of user stories:

- "As a user, I want to be able to create a recipe from scratch and save it to the database."
    - The user clicks on a button "Create Recipe". Thereafter, the user is taken to the "Add/Create Recipe" page. After the user is done creating the recipe, the user clicks "Add recipe" button. The recipe is then saved to the database.
- "As a user, I was to be able to edit an existing recipe."
    - The user clicks on a button "Edit recipe". Thereafter, the app checks if the recipe already exists in the database.
        - If the recipe exists in the database, the app returns a message back and the user is automatically taken to the edit recipe page.
        - If the recipe does NOT exist, an error message is returned and then a printed error is returned to the user stating the recipe they are wanting to edit, does not exist in the database.