**Applied object-oriented programming (EEN060/EEN065)**

*Tillämpad objektorienterad programmering*

**Final Project**

# Project group

Group number 40

1. Kasper Persson Palmqvist TKDES
2. Alicia Sjödin TKDES

# Project idea

## Project title

Coffee chooser

## Project description

|  |
| --- |
| Create a webb application that allows users to register and create a user profile with email and password. The website has library of different coffee types with different flavor profiles that will be searchable. The user will also have the ability to like different coffee types. These likes and added to the users profile and a their personal list of coffees they like. The aim of the site is not to show other users what coffee you like, the users profile is personal and not searchable for others. |

## Use case 1: List (mandatory)

|  |  |
| --- | --- |
| Title: | Lists of coffee |
| Description: | A search will lead to a list of coffee matches. The user will also have a personal list of their liked coffees. The user will be able to add and remove coffee from this list (like/unlike). |
| Condition: | User needs to be logged in to access their likes  User does not need to be logged in to search for coffee in the library. |
| Screen sketch: (optional) |  |

## Use case 2: Search (mandatory)

|  |  |
| --- | --- |
| Title: | A search function with 4 conditions that determine which coffee you'll be recommended. |
| Description: | First the user chooses a brewing method. The flavor profile is then determined by choosing a value between 1-4 in three different possible parameters (acidity, intensity, body). When the user has chosen the necessary parameters and pressed the search button, a list of coffee appears |
| Condition: | The user has to choose a brewing method, but the flavor parameters are optional. You can choose none or all on a scale 1-4. |
| Screen sketch: (optional) |  |

## Use case 3: Detail (mandatory)

|  |  |
| --- | --- |
| Title: | Detail of coffee |
| Description: | When the user selects a coffee from the list of search matches a page with more details about the coffee appears. The page contains information and a button to add the coffee to your liked list. |
| Condition: | User needs to be logged in to access the liked-list in his or hers profile. And also be logged in to be able to add to the list. |
| Screen sketch: (optional) |  |

## Use case 4: Create

|  |  |
| --- | --- |
| Implemented?[[1]](#footnote-1) | Yes |
| Title: | Add to your liked list |
| Description: | The personal list of liked coffees is updated when a new coffee is liked/added or removed. |
| Condition: | User needs to be logged in to be able to add a new coffee to the list of liked coffees |
| Screen sketch: (optional) |  |

## Use case 5: Update

|  |  |
| --- | --- |
| Implemented?1 | No |
| Title: |  |
| Description: |  |
| Condition: |  |
| Screen sketch: (optional) |  |

## Use case 6: Delete

|  |  |
| --- | --- |
| Implemented?1 | Yes |
| Title: | Delete a liked coffee |
| Description: | The user will be able to remove a liked coffee when in the profile page. |
| Condition: | User needs to be logged in to access the delete function |
| Screen sketch: (optional) |  |

# Project Proposal – Database Description

*Here the students should provide a list containing the tables/classes expected to be included in the project. For each table, should provide a description, the list of relationships (if any). There also should be a list of columns and their respective data type and any specifics of the columns (whether they are required, foreign keys, length limit, etc.) Comments can be added to the tables/columns if needed.*

***Note****: The names of the tables always have capitalized names (e.g., ThisIsASuitableName) and are in singular form.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | User | | |  |
| Description: | Contains the user information and enables the login. Each registered user becomes a row in this table. | | | |
| Relationships: | * One-to-many: a user can like many coffees. A like is associated with a specific user. | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| User id | Integer | primary key, not null | auto increment |  |
| name | String(40) | not null |  |  |
| email | String(120) | unique, not null |  |  |
| password | String(60) | not null |  | Hashed password |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Like list (CoffeeUser) | | |  |
| Description: | Hold the likes associated with the user | | | |
| Relationships: | * One-to-many: One user can be associated to many liked coffees. * Many-to-many: One coffee can be in several categories (brewing methods) and one category can be associated with many coffees * One-to-many: one coffee can only have one specific value associated with a specific flavor parameter; the specific value can be associated with many coffees. | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| User id | int | primary key, not null | Auto increment |  |
| Coffee id | int | primary key, not null | auto increment |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Brewing method | | |  |
| Description: | Holds the four different categories of brewing methods | | | |
| Relationships: | * One to many: One brewing method can be associated with many coffees. One coffee can be associated with one category | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| Brewing id | int | primary key, not null | auto increment |  |
| Name | str | not null |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Coffee | | |  |
| Description: | The list of registered coffees in the whole system. And their properties. | | | |
| Relationships: | * One coffee can be associated with many users. One user can be associated with many coffees. * One coffee can be associated with one category (brewing method) | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| Coffee id | int | primary key, not null | auto increment |  |
| Coffee name | str | Not null, unique |  |  |
| Coffee description | str | Not null |  |  |
| Picture |  | Not null, unique |  |  |
| Intensity id | int | Not null, foreign key |  |  |
| Body id | int | Not null, foreign key |  |  |
| Acidity id | int | Not null, foreign key |  |  |
| Brewing id | int | Not null, foreign key |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Acidity | | |  |
| Description: | The acidity level is described is described on a level 1 to 4. One value places the coffee in a acidity category. | | | |
| Relationships: | * <Describe here any relationship of this table with other tables> | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| Acidity id | Integer | Primary key, not null |  |  |
| Acidity value | Integer | 1-4 (min to max value) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Intensity | | |  |
| Description: | The intensity level is described is described on a level 1 to 4. One value places the coffee in a intensity category. | | | |
| Relationships: | * One category can be associated with many coffees, one coffee can be associated with one category | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| Intensity id | Integer | Primary key, not null |  |  |
| Intensity value | Integer | 1-4 (min to max value) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: | Body | | |  |
| Description: | The body level is described is described on a level 1 to 4. One value places the coffee in a body category. | | | |
| Relationships: | * One category can be associated with many coffees, one coffee can be associated with one category | | | |
| Column name | **Data type** | **Constraints** | **Default** | **Comment** |
| Body id | Integer | Primary key, not null |  |  |
| Body value | Integer | 1-4 (min to max value) |  |  |

# Database execution of the final project

Follow the instructions in the assignment page.

# Final Project

Follow the instructions in the assignment page.

[ ] I allow my final project to be used as example in the next instances of this course.

[ ] While sharing our project with other students, I want the names of the students to not be associated with the project (anonymized project).

1. As an optional use case, please mark whether or not your group is planning to implement this use case. Please update it in the final version of this document to reflect the final submitted version of your project. [↑](#footnote-ref-1)