Information Management System for a Research Project on Water Kefir Fermentation

Case study fall term 2024/2025

Scenario

An information management solution is needed for a new research project. The project focuses on developing an optimal fermentation process and recipe for water kefir. The goal is to determine the best process parameters and recipe components through various experiments. These include process parameters such as ambient temperature, fermentation duration, light conditions, and oxygen exposure. The ingredients include water, water kefir crystals, citric acid, sugar, and dried fruit as a source of nitrogen. The process parameters and the amount of each ingredient is adjusted experimentally to find the best combination.

Each experiment is carefully documented: This includes the start and end times as well as a logbook where observations and comments are recorded by the researchers during the experiment. Each log entry includes the name of the researcher, the date and the time. After the fermentation process is completed, the resulting water kefir is analyzed in the university's chemical lab, focusing on alcohol content, residual glucose, and acetic acid. These results are stored alongside the experiment data. Each fermentation takes place in one of the university's sixteen fermentation tanks, and after each experiment, the used tank is thoroughly cleaned. A total of five researchers are working on this project.

For batches meeting specific outcome criteria, researchers conduct a sensory study where a panel of tasters evaluates taste and aroma on a scale from 1 to 5. The number of tasters varies depending on panelist availability. This feedback is crucial for optimizing the water kefir further. Researchers store only the gender, age, and whether the panelist is vegetarian or vegan.

Task 1: Modeling the database

Our first step is to get a clearer picture about the information requirements and how a database model could look like!

1. Model the information requirements using an **Entity Relationship Diagram**. Take into account the scenario description from above!

2. Transform the ERM into a **relational database model**!

Task 2: Implementing the database

Implement the relational database model in a MySQL database (or any other relational database). Write an SQL script that creates the entire database. Ensure that this script can be executed repeatedly without causing errors. Choose appropriate data types for each column in your database. Don't forget to include primary and foreign key relationships in your design, and follow standard SQL naming conventions.

Task 3: Modeling the business process(es)

Now that the database is set up, plan the required steps and consider how these can be supported by the information management system. Put yourself in the shoes of the researchers. Model the entire process using Business Process Modeling Notation (BPMN) to understand and design the process from all perspectives. Include in your BPMN model the points where access to the database is required, both for reading and writing data.

Task 4: Building the data layer

Identify, in each step of the business process that requires data access, which of the four CRUD operations (Create, Read, Update, Delete) is needed. Write a corresponding Python function to perform this operation. Include all necessary parameters in the function signature and give your functions meaningful names.

Task 5: Adding a user interface

Develop a simple user interface using Streamlit. For each step in the business process that requires human input, create an appropriate page. This page should include all elements necessary to collect information from the users and display important information. Ensure that these pages are created for all user roles, considering their different needs. Use the data access layer to connect to the database as needed.

Task 6: Programming the application logic

The system should automatically flag an experiment as eligible for tasting if the alcohol content is below 1% and the residual glucose is above 2 grams per liter. All other configurations should be flagged as not eligible for tasting. Implement this logic in your application layer.