University of Zurich[UZH]

**Department of Informatics**

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

**Prof. Dr. Michael Böhlen**
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, 24. Juli 2017

**Software Project: Development of a dynamic web application**

## CONTEXT

The Swiss Feed Database[1] contains historical data about 100 feed types and 900 nutrients. A web application is provided allowing users to define a set of selection criteria (e.g., nutrient and feed types, temporal and geographical information about feed samples), retrieve nutrient content data, and overview results in various intuitive forms. This project aims at continuing the work of Lukas Yu [1] on developing Swiss Feed Database v2.0 following the Model-view-controller design pattern that separates data management (model), interface (view), and business logic (controller) of the application. The goals of this re-implementation are:

- to end up with a codebase that complies with the MVC design pattern principles.
- to design a star schema for the database that satisfies referential integrities and controls redundancy
- to optimize how much time is spent on server and client side, respectively, document the results, and implement a web application that asynchronously load data.
- to design a user interface that follows a responsive web design approach and extend the functionality (e.g., new selection criteria).

## FUNCTIONALITY

Swiss Feed Database v2.0 implemented a viewer. So far this second version provides a dynamic data-driven task bar. The options of the task bar depend on the data stored in the database, the access rights of the user and already chosen options. After a query has been formed and posed, data are presented to the user with a sortable table with one nutrient per column and one feed type per row. Two additional viewers must be implemented:

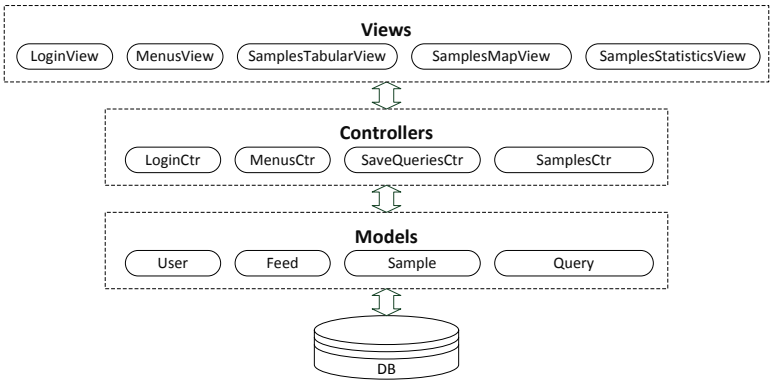
[1] `http://feedbase.ch`

**Statistics:** A drop-down menu must provide various statistical tools. Graphs like a scatter plot or a correlation diagram between nutrients must be provided as options. Another option must be a statistical report, containing sample count, min, max, average, standard deviation. Finally, a tool of finding the most similar or dissimilar samples of a selected sample must be provided.

**Sample origins map** A map must depict the origin location of samples and it can also visualize density distribution of sample origins and spatial concentration of selected nutrients.

## ARCHITECTURE

The architecture of the Swiss Feed Database v2.0 is depicted in the following figure:




Models define objects (e.g., User, Feed) representing data in the database and provide basic operations on them (e.g., retrieve all available feeds). Model objects are available to controllers so that they implement the actual business logic of the application. For example, `LoginCtr` implements an authentication process proving that a user is registered in the database and retrieving the user's access rights. Finally, views design the application interface. For example, `LoginView` provides a simple login form, while `MenuView` provides a dynamic menu for selecting feed types, nutrients, etc.

## MILESTONES

Milestone 1: Overview of the current situation (1 week)

The initial version of the web application must be studied. The current implementation of the second version must also be reviewed to assess how complete it is and correct any existing bugs.

Milestone 2: Study server-client communication. Suggest and implement new ideas about making the application more efficient and more user friendly. Add more search criteria in query menus related to biological and technical sample properties. (2 weeks)

A detailed study must be done about how much time is spent in various operations being executed when a user query is applied. The results must be documented so that space for improvement can be investigated. This study must also consider biological and technical properties of retrieved samples.

Furthermore, the result size must be considered. Some queries generate big results (thousands of tuples) that make the application to be not responsive. Solutions for these queries must also be investigated. For example, a query must always give a limited result (e.g., 1000

samples). Then a user must be able to adjust this limit using a slide bar and add more samples incrementally. The criterion about what samples must be selected to be retrieved can be either arbitrary (random) or temporal (e.g., the most recent ones).

Milestone 3: Advanced Visualization (2 weeks)

The visualization forms of statistic tools and a map must be implemented. For this purpose it is suggested that libraries of Google Maps and Google Visualization are used.

Milestone 4: Documentation (1 week)

The architecture of the application must be presented along with descriptions about implemented models, views, and controllers. The delivered documentation must be oriented to developers. Instructions to install the application and running examples must be provided.

**References**

[1] L. Yu, The swiss feed database documentation (February 2017).

**Supervisor:** Georgios Garmpis (ggarmpis@ifi.uzh.ch)

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor